NASA/CR—2014-216651

# High Fidelity CFD Analysis and Validation of Rotorcraft Gearbox Aerodynamics Under Operational and Oil-Out Conditions

## Final Report

*Robert F. Kunz*
*The Pennsylvania State University, University Park, Pennsylvania*



July 2014

# NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to *help@sti.nasa.gov*

- Fax your question to the NASA STI Information Desk at 443–757–5803

- Phone the NASA STI Information Desk at 443–757–5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/CR—2014-216651

# High Fidelity CFD Analysis and Validation of Rotorcraft Gearbox Aerodynamics Under Operational and Oil-Out Conditions

## Final Report

*Robert F. Kunz*
*The Pennsylvania State University, University Park, Pennsylvania*

Prepared under Contract NNC10CA35C



National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

July 2014

*Level of Review*: This material has been technically reviewed by NASA technical management OR expert reviewer(s).

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

# High Fidelity CFD Analysis and Validation of Rotorcraft Gearbox Aerodynamics Under Operational and Oil-Out Conditions
## Final Report

Robert F. Kunz
The Pennsylvania State University
University Park, Pennsylvania 16804

**Executive Summary**

This report, submitted by the Pennsylvania State University Applied Research Laboratory, represents a final deliverable for NASA Cooperative Agreement NNX07AB34A, "High Fidelity CFD Analysis and Validation of Rotorcraft Gearbox Aerodynamics under Operational and Oil-Out Conditions."

This report is submitted with the two other final deliverables of the contract: 1) NPHASE-PSU Theory, User's and Reference Manual Version 3.16, and, 2) NPHASE-PSU V3.16 source code, installation scripts, tutorial and instructions transferred electronically (CD contains this report, the manual and **nphase-05-31-2013.tar.gz**)

This report summarizes the many technical accomplishments achieved under the two-part NRA. The technical and other accomplishments of the NRA are summarized here:

1. The CFD code, NPHASE-PSU, was instrumented to accommodate the numerous challenges involved in modeling gearbox systems, including:
   - multiphase flow modeling for oil droplets, oil films and compressible air
   - gear motion in the relative and absolute frames of reference
   - gear tooth contact (in the context of overset meshing)
   - conjugate heat transfer
2. These NPHASE-PSU capabilities were brought to bear in analyzing gearbox aerodynamics in the context of other modern CFD practices including high order discretization, parallel programming/execution (on NASA NAS HPC resources), implicit multi-component numerics, and 2-4 equation/LES turbulence modeling.
3. Many verification and validation cases were carried out
4. Significant new understanding of the physics of gearbox windage was achieved. This has led directly to design guidance (including a patent and several rotorcraft industry sponsored projects)
5. This research has led to five conference publications[i-v], two journal publications[vi,vii], two PhD students/theses[viii,ix], and one (provisional) patent[x].
   i) Hill, M.J., Kunz, R.F., Medvitz, R.B., Noack, R.W., Long, L.N., Morris, P.J., Handschuh, R.F., "Application and Validation of Unstructured Overset CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulations," American Helicopter Society 64th Annual Forum, Montreal, Canada, April, 2008.
   ii) Hill, M.J., Kunz, R.F., Noack, R.W., Handschuh, R.F., Long, L.N., Noack, R.W., "CFD Analysis of Gear Windage Losses: Validation and Parametric Aerodynamic Studies," American Helicopter Society 66th Annual Forum, Phoenix, AZ, April, 2010.
   iii) Kunz, R.F., Hill, M.J., Schmehl, K.J., McIntyre, S.M., "Computational Studies of the Roles of Shrouds and Multiphase Flow in High Speed Gear Windage Loss," American Helicopter Society 68th Annual Forum, FortWorth, TX, April, 2012.

iv) McIntyre, S.M., Yu, Q., Saribay, Z.B., Kunz, R.F., "Fluid and Thermal Dynamics Simulation of Spur Gears Under Steady State Operation," Manuscript accepted for American Helicopter Society 69th Annual Forum, Phoenix, AZ, April, 2013.

v) Noack, R. W., Boger, D. A., Kunz, R. F., and Carrica, P. M., "Suggar++: An Improved General Overset Grid Assembly Capability," AIAA Paper 2009-3992, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.

vi) Hill, M.J., Kunz, R.F., Medvitz, R.B., Handschuh, R.F., Long, L.N., Noack, R.W., Morris, P.J. "CFD Analysis of Gear Windage Losses: Validation and Parametric Aerodynamic Studies," *ASME Journal of Fluids Engineering*, 133:031103-1-10, March 2011.

vii) Hill, M.J., Kunz, R.F., Noack, R.W., Long, L.N., Morris, P.J., Handschuh, R.F., "CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulations," Gear Technology, 48-55, August, 2009.

viii) Hill, M.J., A Computational Investigation of Gear Windage, PhD Thesis, Department of Aerospace Engineering, The Pennsylvania State University, 2010.

ix) McIntyre, S.M., Multiphase Modeling of Droplet-Film Flows, PhD Thesis, Department of Aerospace Engineering, The Pennsylvania State University, anticipated 2015.

x) Kunz, R.F., Medvitz, R.B., Hill, M.J., "High Speed Gear Sized and Configured to Reduce Windage Loss," Provisional Patent Application No. 61/477,666.

6. The NRA project has led to MANY synergistic and follow-on activities, which is a testament to the value of this effort:
   - Three Bell sponsored VLC projects (helicals)
   - Boeing sponsored IRAD effort (spiral bevel)
   - VLRCOE LOL project
   - Pratt and Whitney Gearbox Lubrication Facility
   - Current NASA Glenn instrumentation opportunity
   - Benefit of multiphase modeling to Icing NRA with Bell
7. The PhD student who was funded by the NRA graduated (Matthew Hill) in 2011 and now works at Bell Helicopter

# Final Report

In this project, CFD modeling was evolved, applied and validated for gear windage aerodynamics. This final report presents the accomplishments carried out under this NRA project. The author and his students and colleagues executed research under the NRA from 2007 through 2012, and this led to an AHS Forum paper in 2008, which was solicited for and appeared in Gear Technology, an AHS Forum paper in 2010, which was submitted and published in the ASME Journal of Fluids Engineering, and an AHS Forum paper in 2012. Additionally some of our research developed under the NRA appeared in a paper presented at the 2009 AIAA CFD Conference.

These four papers represent all of the research carried out under the NRA. We proceed chronologically here.

In the first two years of the program, we enabled NPHASE-PSU with the instrumentation required to predict incompressible, single phase (air-only), isolated gear windage simulations, using overset meshes, solving on very large meshes in the absolute frame of reference. We demonstrated a very convergent dual-time moving mesh numerical scheme, and excellent parallel scaling for the code on the large NASA clusters used to perform the simulations. The code was verified and validated for several canonical flows, and first-of-their-kind fully 3D windage validation cases were carried out showing good agreement between CFD and experiment. Viscous/turbulence modeling was identified as a shortcoming since loss power was consistently underpredicted for viscous-drag-only spinning disk cases. Low Reynolds number (sublayer resolved) turbulence modeling exhibited improved performance for these cases. High Reynolds number two equation modeling proved appropriate for modeling the moderate speed Diab spur gear suite, due apparently to the dominance of the pressure torques on spin-down. The budgets of viscous and pressure components of spin-down torque suggested that viscous effects will become much more important, perhaps exceeding 50 percent, for gears with pitchline speeds approaching twice that investigated by Diab and the present authors. Some interesting secondary flow and pressure gradient physics were identified, although we did not interpret the physics in this early work enough to draw any significant conclusions. The 2008 AHS Forum paper which summarizes this early research is included here in its entirety.

# Application and Validation of Unstructured Overset CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulation


Matthew J. Hill *
*The Pennsylvania State University*
*University Park, PA*
*mjh414@psu.edu*

Robert F. Kunz †
*The Pennsylvania State University*
*University Park, PA*
*rfk102@only.arl.psu.edu*

Ralph W. Noack ‡
*The Pennsylvania State University*
*University Park, PA*
*rwn10@only.arl.psu.edu*

Lyle N. Long §
*The Pennsylvania State University*
*University Park, PA*
*lnl@psu.edu*

Philip J. Morris ¶
*The Pennsylvania State University*
*University Park, PA*
*pjm@cac.psu.edu*

Robert F. Handschuh‖
*NASA Glenn Research Center*
*Cleveland, OH*
*robert.f.handschuh@grc.nasa.gov*



**An unstructured overset moving mesh CFD method is adapted, validated and applied to spinning gear systems with emphasis on predicting windage losses. Several spur gears and a disc, spinning in air at rotation rates up to 1200 $s^{-1}$ are studied. It is observed that the CFD simulations return good agreement with measured windage power loss, as determined by the deceleration of the gears due to torques exerted by viscous and pressure forces on the gear surfaces. Turbulence modeling choices, the relative importance of viscous and pressure torques with gear speed, and the physics of the complex 3D unsteady flow field in the vicinity of the gear teeth are studied. The capabilities and challenges associated with the overset mesh approach for enclosed gear train applications are demonstrated.**


## Introduction

Gearbox windage refers to the power losses associated with rotational deceleration torques exerted on spinning gears by aerodynamic forces (pressure and viscous) within the air-oil atmosphere present within a gearbox. Windage losses are a source of significant heating and fuel consumption in rotorcraft and other VTOL systems. The weight and packaging constraints inherent in these systems require the gearing components to be lightweight and heavily loaded. The components are also required to operate at high rotational speeds where windage losses become significant with respect to other gearbox losses (rolling, sliding, and lubrication losses.) Windage losses are relevant to aircraft design for several reasons: 1) They can consume several percent of the transmitted power. This has significant implications for onboard oil cooling requirements and lube system capacity thereby compromising range and standby military readiness for rotorcraft and carrier-based aircraft. 2) Rotorcraft platform survivability under transmission *oil-out* conditions is exacerbated by windage losses which are manifested as added dissipative flow heating to these already critically thermally stressed systems.

Despite this significant relevance, design efforts throughout the gearing industry aimed at reducing windage losses have generally fallen into the trial and error category. Nevertheless, it has been shown by Winfree[1] and others that modest geometric modifications to control the air flow path, such as shrouding and baffle configurations, can very significantly reduce both windage losses and lubricating oil consumption (80% and 40% reductions observed, respectively, in Ref. 1). However, these hardware specific approaches are empirical, expensive, and time consuming, and to be relevant need to be performed late in the design cycle.

A host of experimental studies have appeared in the literature.[1–8] These studies employ either closed loop systems[2–6] or treat isolated gears[1,7,8] where windage (and other) losses are determined by measuring spin-down velocities once the gear+shaft assembly is disconnected from the drive torque. Figure 1 shows a diagram of the high speed helical gear train test facility at the NASA Glenn Research Center. It is a closed-loop system that has been


*Ph.D. Candidate, Department of Aerospace Engineering
†Senior Research Associate, Applied Research Laboratory
‡Research Associate, Applied Research Laboratory
§Professor of Aerospace Engineering
¶Boeing/A.D. Welliver Professor of Aerospace Engineering
‖Aerospace Engineer, Army Research Laboratory Vehicle Technology Directorate



Presented at the American Helicopter Society 64th Annual Forum, Montreal, Canada, April 29 - May 1, 2008.

setup to study the thermal behavior of aerospace-quality gear components under various speeds, loads, and lubricant flow rates.[5] Generally, these studies parametrize gear geometry elements, rotational speed, enclosure geometry, and lubrication system characteristics (flow rate, jet location, lubricant rheology), and use dimensional analysis to develop correlations for the power losses. These correlations, although useful in the design process, are inherently limited by the large number of system variables and the attendant limited range of their applicability. Of particular concern here is the paucity of data/correlations available for high speed gears of interest. Indeed, compressibility effects are mostly not even considered in the literature, although high speed gears can have tip Mach numbers reaching 0.75.

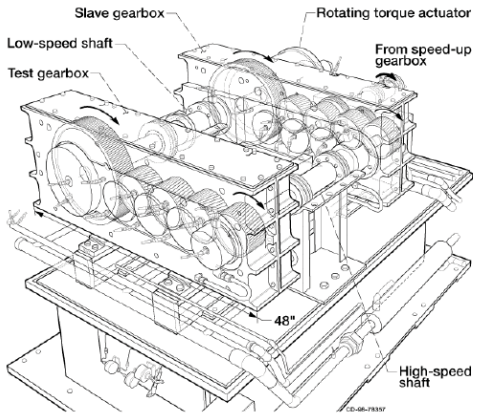



**Fig. 1 NASA High-Speed Helical Gear Train Facility.[5]**

Unfortunately, the physics of these systems are so complex that to date there have been no attempts made to employ many modern elements of 3-D computational fluid dynamics (CFD) in analyzing gearbox windage. Recent 2-D studies[9] were performed using the commercial CFD solver FLUENT, where a side correlation factor was used to to account for 3D effects (although these authors state that work is under way to extend their simulations to 3-D[10]) Specifically, the fluid mechanics involve complex separated air flow, disperse multiphase flow (oil droplets) and continuous multiphase films (lubricating oil on gears), moving boundaries in contact, and all modes of heat transfer. Accordingly, for a CFD tool to resolve all of the relevant physics of this problem it must:

- support moving meshes (either adaptive/deforming or overset) necessary to resolve the gears in relative motion and contact at the gear face
- contain non-equilibrium multiphase flow capability (separate continuity and momentum equations for each phase; slip between phases) to accommodate the disperse mist/droplet and continuous film flows
- support suitable turbulence modeling to accommodate the complex high speed separated flow within the gearbox, and to accurately represent the cascade of energy through turbulence scales into viscous heating
- posses suitable preconditioned time-stepping algorithmics to efficiently accommodate Mach numbers ranging from near zero through high subsonic
- support conduction (gears themselves meshed internally - solve conduction problem in these elements), convection and (for near-failure conditions) radiation modeling
- be parallelized to run efficiently on modern High Performance Computing (HPC) systems.

The overall goal of this research is to adapt, validate and apply a CFD tool that supports these basic gear windage modeling requirements. This paper represents the first publication of the authors' efforts in this area, and focuses on the single phase aspect of the problem, although, the CFD code used supports the multiphase flow requirements that will be needed in the future. The paper is organized as follows: First, the governing equations and numerical procedures, including overset grid technology are reported. Second, several *verification* cases are reported, where results are compared to analytical solutions for rotating Couette flow, flow near an infinite spinning disc, and thermal Couette flow. Next several *validation* cases are reported, including windage loss predictions compared to data obtained by Diab et al.,[8] for a spinning disk and four spur gears. Gridding requirements and turbulence modeling are discussed.

## Theoretical Formulation

*Governing Equations*

The conservation of mass, momentum and energy can be written in integral conservation law form for a compressible flow through a moving mesh as:

$$\frac{\partial}{\partial t}\int_{\forall}\rho d\forall+\int_{\underline{S}}\rho(\underline{V}-\underline{W})\cdot d\underline{S}=0 \tag{1}$$

$$\frac{\partial}{\partial t}\int_{\forall}\rho\underline{V}d\forall+\int_{\underline{S}}\rho\underline{V}(\underline{V}-\underline{W})\cdot d\underline{S}=-\int_{\underline{S}}pd\underline{S}+\int_{\underline{S}}\underline{\underline{\tau}}\cdot d\underline{S} \tag{2}$$

$$\frac{\partial}{\partial t}\int_{\forall}\rho Ed\forall+\int_{\underline{S}}\rho H(\underline{V}-\underline{W})\cdot d\underline{S}=\int_{\underline{S}}\left(\underline{\underline{\tau}}\cdot\underline{V}\right)\cdot d\underline{S}+W_f+q_H \tag{3}$$

In equations 1-3, $\underline{V}$ is the velocity vector, and $\underline{W}$ is the velocity of the surface element, $d\underline{S}$, both in the absolute frame of reference. In the present work, all verification and validation flows studied are either incompressible, or have maximum local absolute Mach numbers of less than 0.35. Accordingly, for all simulations presented in this paper, an incompressible assumption is invoked and the energy equation is not solved (except for the thermal Couette flow simulation where it is solved subject to a constant density constraint).

A high-Reynolds number k-ε turbulence model and a sublayer resolved hybrid k-ε/k-ω turbulence model due to Menter[11] are used in the studies that follow. No explicit transition model was employed as justified, for now, by the small contribution of near-axis viscous torques on windage loss.

*CFD Numerics and Code*

The CFD code used in this work, NPHASE-PSU[12] is a parallel face-based, cell-centered, arbitrary-element unstructured multiphase flow solver which has been instrumented with overset mesh capability. The baseline algorithm follows established segregated pressure based methodology. A collocated variable arrangement is used and a lagged coefficient linearization is applied.[13] Diagonal dominance preserving, finite volume spatial discretization schemes are used for the scalar transport equations. Continuity is introduced through a pressure correction equation, based on the SIMPLE-C algorithm.[14] In constructing cell face fluxes, a momentum interpolation scheme[15] is employed which introduces damping in the continuity equation. Grid motion/deformation terms are implemented in a Geometric-Conservation-Law (GCL) preserving fashion.[16] A dual-time formulation is employed where at each physical timestep, between 5 and 20 pseudo-timesteps of the SIMPLE-C algorithm are applied. Specifically, at each pseudo-timestep, the discrete momentum equations are solved approximately (using a simple point iterative scheme), followed by a more exact solution of the pressure correction equation (using the PETSC[17] parallel LU preconditioning+GMRES utilities). Turbulence scalar and energy equations are then solved in succession. Parallelization is implemented in a standard fashion by invoking domain decomposition based on METIS[18] in the front end, and MPI based message passing in the CFD code. All of the large scale simulations presented in this paper were executed on the Columbia supercomputer at NASA Ames Research Center. The code scales very well on this system as illustrated in Figure 2.

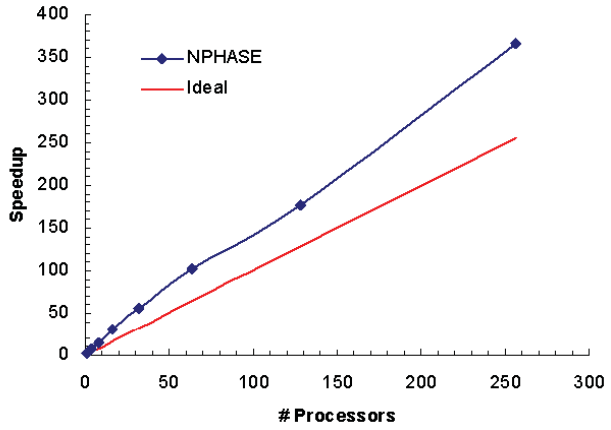



**Fig. 2 Parallel efficiency of NPHASE-PSU on Columbia system for a 1.1x10$^6$ cell spinning cylinder case.**

**DiRTlib and SUGGAR**

The overset grid approach[19] utilizes a composite grid consisting of a set of overlapping component grids to discretize the domain. No point-to-point or face-to-face matching is required between component grids. The solution on the component grids are linked by identifying appropriate intergrid boundary points (IGBPs) where the solution is given by a specified boundary value obtained by interpolation from another overlapping donor component grid. The overset domain connectivity information (DCI), which consists of the identification of the intergrid boundary points and corresponding interpolation sources, is obtained by an overset grid assembly step. The current effort utilizes two overset software libraries to add the overset capability to NPHASE-PSU.

DiRTlib,[20] which stands for Donor interpolation Receptor Transaction library, is a solver neutral library that encapsulates the functionality required by the solver to utilize the overset domain connectivity information. It is independent of the solver grid storage and topology, dependent variables, etc. and can be used with any solver. The solver calls a few functions to initialize the library, load the DCI interpolation, and transfer the data to appropriate processors in a parallel execution environment, and apply the interpolated data as boundary values at IGBPs. Solver functions must be provided and are called by DiRTlib to get and put data in the correct solver dependent variable storage locations. When the solver executes in a distributed memory parallel computational environment the solver must also inform DiRTlib of the parallel decomposition enabling DiRTlib to get/put data from the appropriate parallel process.

The current overset grid assembly process is performed using the SUGGAR code,[21] which stands for Structured, Unstructured, Generalized overset Grid AssembleR. SUGGAR is a general overset grid assembly code with the capability to create the domain connectivity information at node and/or element centers for most current grid topologies including any combination of structured Cartesian and curvilinear, unstructured tetrahedral and mixed element, general polyhedral, and octree based Cartesian grids. For static grid assemblies with no motion between component grids the grid assembly is a pre-processing step. The case of solution and time dependent motion, requires the solver to communicate the new body and grid positions to the grid assembly process, wait for it to complete, and then load the new DCI. For the case of prescribed motion, such as used in the present study, the DCI is a-priori computed and saved in a file for each time step in the simulation and the solver simply loads the file appropriate for each time step.

The donor interpolations produced by SUGGAR are a set of linear weights that multiply the values at the donor members. For a cell centered flow solver, such as NPHASE-PSU, the interpolation stencil will use as members the cell in the donor grid that was found to contain an IGBP and the neighboring cells that share a face with the donor cell. The interpolation weights are computed using

an unweighted least square procedure.

## Results

### Verification Studies

In the context of the overset meshing strategy employed for gearbox windage simulations, the meshes will be in motion relative to one another. As indicated above, the approach taken here is to solve the flow in the absolute frame of reference for the entire computational domain, i.e. on all meshes, those that are rotating and those that are stationary. In order to verify that NPHASE-PSU correctly handles these gear relevant rotating mesh systems, two verification studies were performed, rotating Couette flow, and flow near an infinite rotating disc, both of which have available analytical solutions.

*Rotating Coutte Flow*

Figure 3 is an illustration of the 33x81 (radial x azimuthal) computational domain for the incompressible rotating Couette flow case. The inner and outer boundaries are walls. In this case, $r_{inner} = 0.5$ and $r_{outer} = 1.0$. The outer cylinder is held stationary and a rotation rate of $\omega = 1s^{-1}$ is specified for the inner cylinder. The Reynolds number independent analytical solutions for the tangential velocity and pressure are:

$$V_\theta = Cr\left(1-\frac{1}{r^2}\right) \quad (4)$$

$$p = \rho\left(\frac{C^2r^2}{2} - 2C^2 ln(r) - \frac{C^2}{2r^2}\right) \quad (5)$$

$$C = -\frac{\omega}{3} \quad (6)$$

Figure 4 shows a comparison of the analytical solution with three NPHASE-PSU runs, designated Approaches 1, 2, and 3. Approach 1 solves the absolute velocities in the absolute frame on a stationary grid (adapting inner cylinder boundary conditions accordingly). Approach 2 solves for the relative velocities in the relative frame on a stationary grid (i.e. frame of reference rotating with angular velocity, ω, adapting the momentum equation source terms and outer cylinder boundary conditions accordingly). Approach 3 solves for absolute velocities in the absolute frame using a time accurate analysis on a rotating mesh. Approach 3 is the most relevant for gear analysis. Figure 4 illustrates that the code returns the analytical solution for all three simulation approaches to within the accuracy of the second order accurate discretization numerics and grid used.

*Flow Near an Infinite Rotating Disc*

The second verification case is a classic 3-D exact solution to the incompressible Navier-Stokes (N-S) equations. An infinite radial span disk rotates with an angular velocity, ω. This induces tangential flow in the direction of rotation, radial outflow, and an axial flow towards the center of the disc. In this case, the N-S equations reduce to a system of

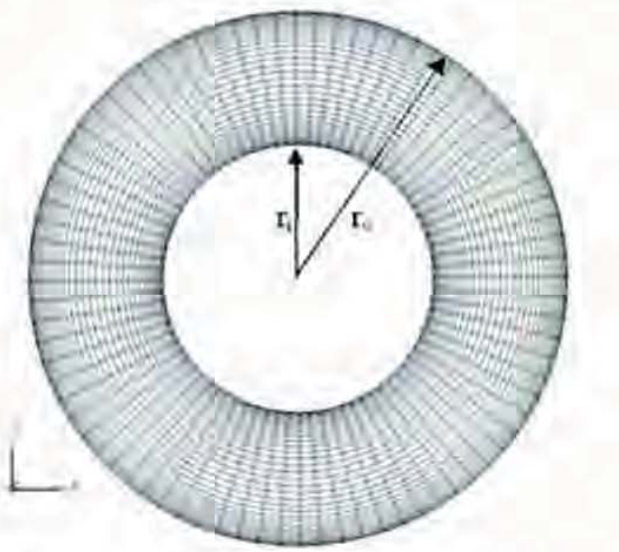


**Fig. 3 Computational domain for the rotating Coutte flow case.**

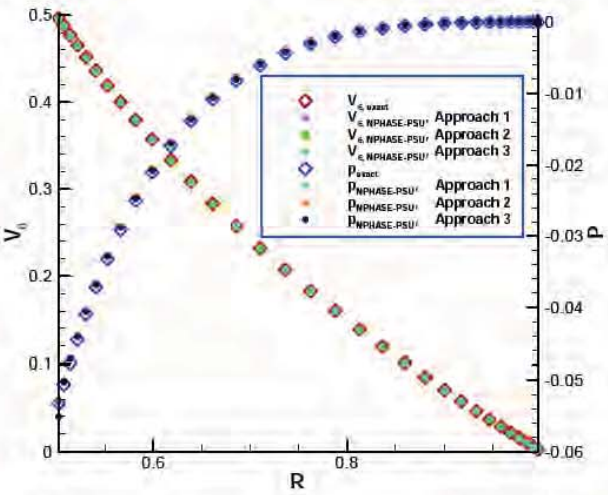



**Fig. 4 Comparison of NPHASE-PSU and analytical solutions for the rotating Couette flow case.**

non-linear ODEs which can essentially be exactly solved numerically. Figure 5 is notional sketch of the flow field from Schlichting.[22]

Figure 6 shows the 232662 element unstructured mesh employed for the analysis. The extent of the computational domain was selected to be $r_{max} = 1.0$, $z_{max} = 1.0$. For a choice of $\omega = 1.0$ and $\mu = 1.0\text{x}10^{-2}$, this domain provided that the solution sampled within the region $r \leq 0.2, z \leq 0.4$ compares very closely with the exact solution despite the necessarily finite extent of the domain.

NPHASE-PSU was applied using two approaches. Approach 1 solves the absolute velocities in the absolute frame on a stationary grid (adapting disk boundary conditions accordingly). Approach 2 solves for absolute velocities in the absolute frame using a time accurate analysis on a rotating mesh. Figure 7 illustrates that the code returns the exact solution for both simulations to within the accuracy

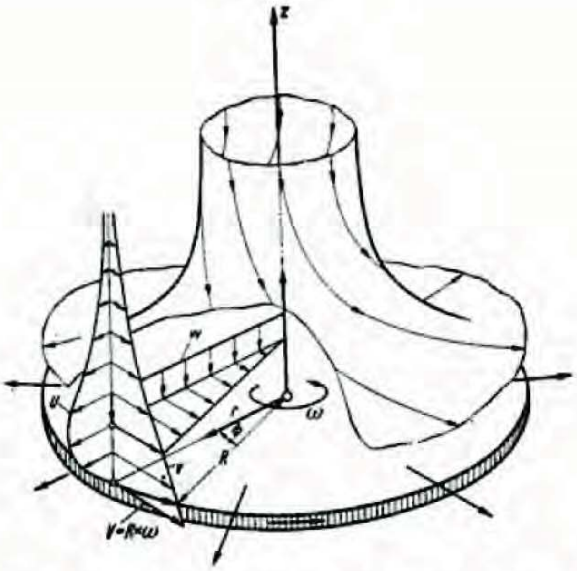



**Fig. 5 Sketch of the flow field in the vicinity of an infinite span rotating disc.**

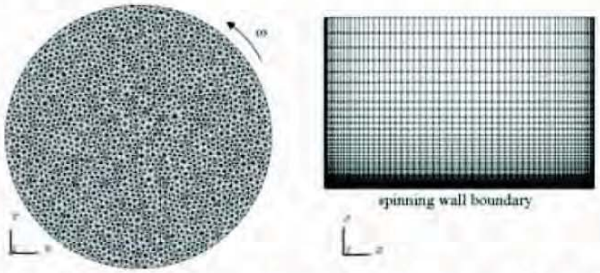



**Fig. 6 232662 element unstructured mesh employed for the infinite spanning rotating disc.**

of the second order accurate discretization numerics and grid used.

*Couette Flow with Wall Heating*

The third validation performed sought to verify the viscous dissipation term implementation in NPHASE-PSU. The relevance of viscous dissipation physics to gear windage is significant as discussed above. The analytical solution for the temperature distribution in a laminar linear Couette flow was chosen. Figure 8 shows a diagram of the configuration. The product of the Prandtl number ($Pr \equiv \mu C_p/k$) and the Eckert number ($Ec \equiv U_1^2/C_p(T_1 - T_0)$), $PrEc$, is a measure of the role of viscous dissipation in a flow. The nearly exact comparisons between CFD result and the analytical solution, shown in Figure 9, across a range of $PrEc$, illustrates that the viscous dissipation terms in NPHASE-PSU are implemented correctly.

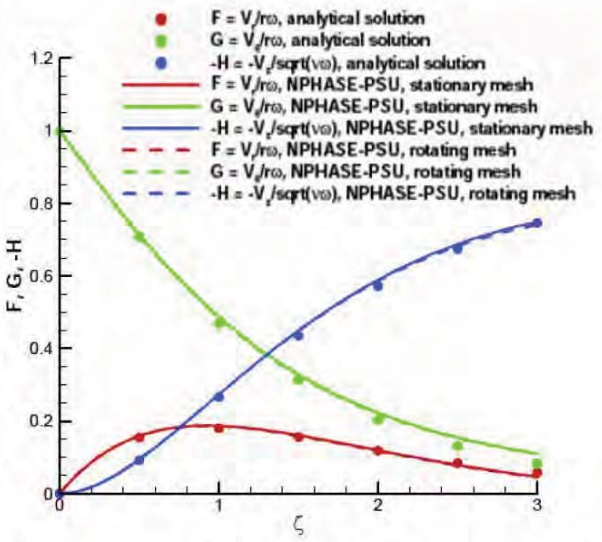



**Fig. 7 Comparison of NPHASE-PSU and exact solutions for the infinite span rotating disc.**

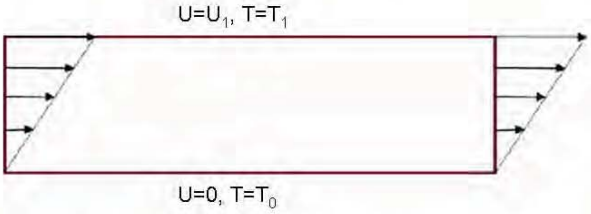



**Fig. 8 Illustration of the linear heated Couette flow case.**

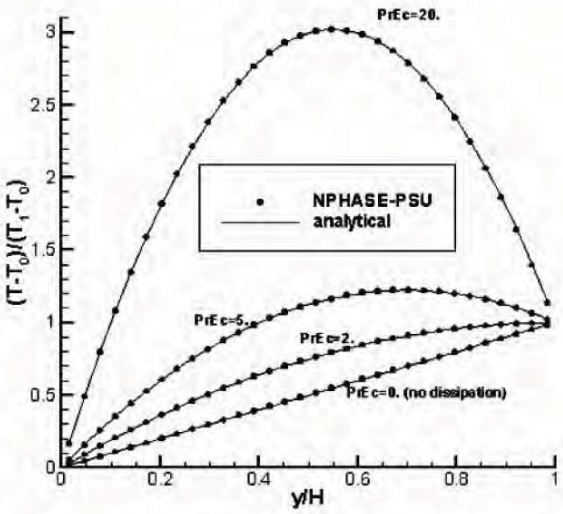



**Fig. 9 Comparison of NPHASE-PSU and exact solutions for the linear heated Couette flow case.**

**Validation Studies**

The experimental data of Diab et al.[8] was used to validate NPHASE-PSU for the case of unshrouded, isolated, rotating spur gears. Diab et al.[8] tested four different spur gears and a disk in free air on a spin-down test rig. The gears varied in diameter, width, and tooth count. The properties of the gears and disk used by Diab et al.[8] are provided in Table 1. Diab et al.[8] did not study the effects of gear enclosure or lubrication. A sequence of prescribed constant rotation rate simulations was used to replicate the experiment.

Table 1 Diab et al.[8] Gear Properties.

| | Pitch Diameter (mm) | Tooth Width (mm) | Module (mm) |
|---|---|---|---|
| Gear 1 | 288 | 30 | 4 |
| Gear 2 | 144 | 30 | 4 |
| Gear 3 | 144 | 60 | 4 |
| Gear 4 | 144 | 60 | 6 |
| Disk | 300 | 30 | |

*Single Grid Simulations*

Grids were generated for all four spur gears and the disk. For the gear studies where the high Reynolds number k-ε turbulence model was used, near-wall grid spacing was defined to accommodate wall-functions (e.g., $y^+ \simeq 70$ for gear 1, $\omega = 1000\ s^{-1}$). The single plane of symmetry in the problem was exploited to reduce total cell count by a factor of 2. Grid cell counts for the different cases varied between $2x10^6$ (Gear 4) and $8x10^6$ (Gear 1). Grid generation was further simplified by employing a hybrid mesh topology, as illustrated in Figure 10. Specifically, for the regions above the surface of the gear teeth to the outer boundary, structured hexahedral cells were used. For the region above the gear face surface, unstructured prism cells were employed. The meshes were generated using the commercial grid generation software package Gridgen.[23] The computational domain of the isolated gear grids was extended to approximately five times the gear radius from the gear surface in all directions. It was found that this distance was adequate for defining a symmetry boundary condition since the flow is nearly stagnant there.

An azimuthal step size of 1/40th of one tooth passage duration (the time it takes one tooth to rotate to the position of the tooth adjacent to it) was used in all CFD calculations. This corresponds to 2880 timesteps per gear revolution for Gear 1, 1440 timesteps for Gears 2 & 3, and 960 timesteps for Gear 4. All cases used 10 pseudo-time iterations per physical timestep.

CFD runs were made for four gears and the disk at a number rotation speeds. All cases were run for at least 2 complete revolutions to remove simulation startup transient behavior. Convergence histories show that transients leave the solution after about one revolution as illustrated in Figure 11, where it is also observed that pseudo-time residual drop approximately 2 orders of magnitude in each physical timestep when 10 pseudo-timesteps are used per physical timestep.

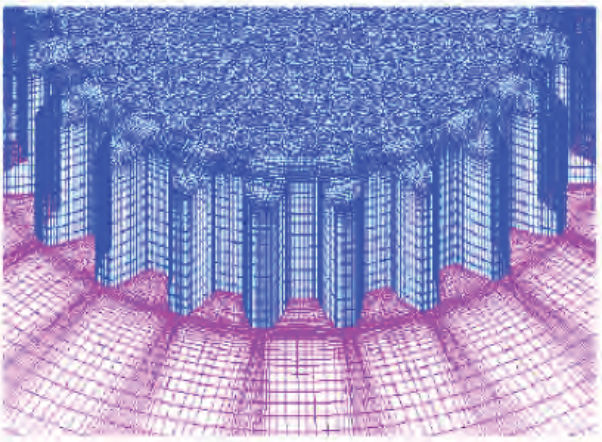


Fig. 10 Grid topology of Gear 4.

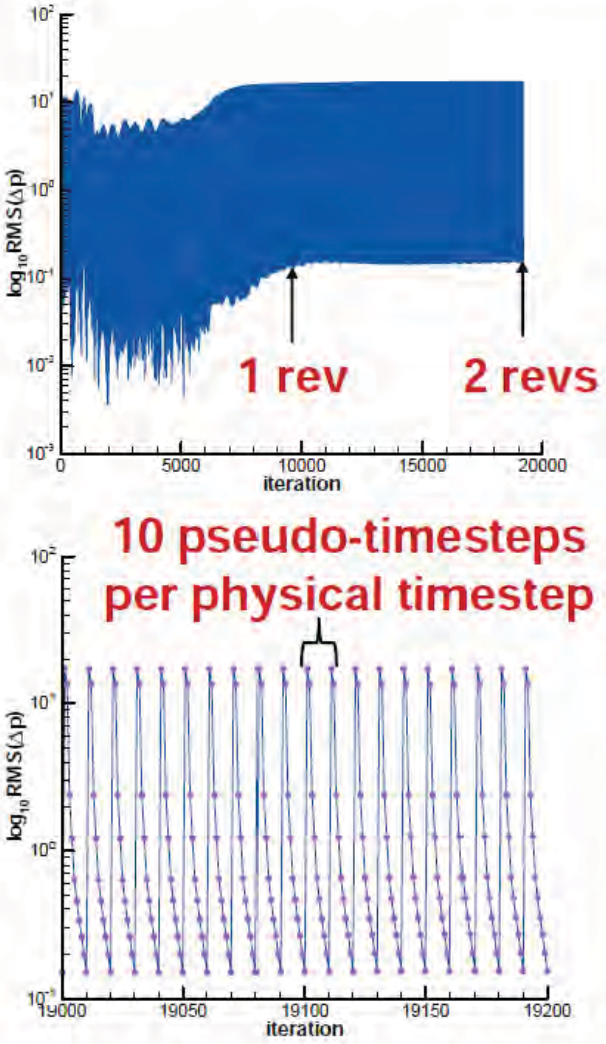



Fig. 11 Example convergence history for Diab Gear 4.

Comparisons between the power loss results of Diab et al.[8] and the NPHASE-PSU analysis are presented in Fig. 12 for all four gears. The CFD analysis for all 4 gears exhibited very good agreement with experiment. The disk case, however, did not share this same level of agreement, as illustrated in Figure 13, where NPHASE-PSU results are seen to underpredict the measured power loss. In order to elucidate the reasons for the deterioration in solution accuracy observed for the disk case, a number of observations and studies were made. First, it is observed that the measured (and computed) windage loss power levels for the disk are much smaller than the comparably sized spur

gear (Gear 1, D $\simeq$ 300 mm). This arises due to the absence of any azimuthal pressure variation in the disc flow - torque losses are due entirely to viscous effects, and these are clearly underpredicted. Indeed the absolute magnitudes of loss underprediction between the disk and Gear 1 are comparable (e.g. $\simeq$ 50W @ $600s^{-1}$), so presumably this underprediction of shear is present in all of the gear simulations, however its relative magnitude is small.

To explore this further, the low-Reynolds number Menter model was applied to the spinning disc case (using an appropriate sublayer resolved mesh). Figure 13 illustrates that improved turbulence modeling does benefit solution accuracy especially at higher rotation rates. This observation is not as important for the gear cases since the spin-down torques associated with azimutally varying pressure forces in the vicinity of the gear teeth dominate the viscous forces as shown in Figures 14 and 15. It can be seen that, especially at lower rotation rates, the contribution of viscous loss to total windage loss, is small. However, we do observe that at higher pitchline velocities the relative magnitude of the viscous torque increases with respect to the pressure torque. This can be see for higher rotation rates on Gear 1 (Figure 14) as well as by considering the smaller size (and hence lower pitchline velocities) of Gear 2 (Figure 15).

In summary our results suggest that for the very high speed gears to be encountered in rotorcraft (and other high performance aircraft) transmissions, viscous effects will become more important than encountered here, and will therefore require attendant research attention.

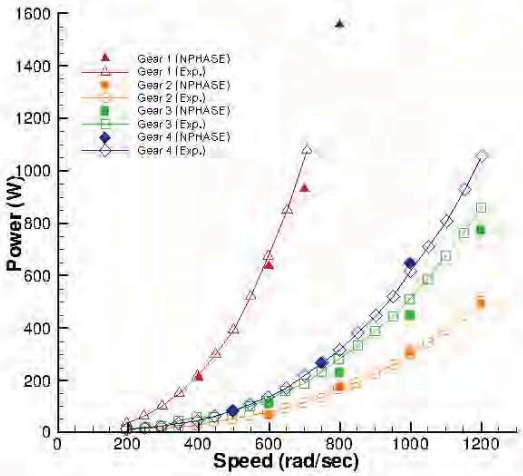



**Fig. 12 Comparisons between the experimental results and the NPHASE-PSU analysis.**

*Details of 3-D Flow Field*

A number of important physical features of the predicted flow field are available upon interrogation of the CFD simulations. Figure 16 shows a view of the predicted secondary velocity vectors on the symmetry plane in the gear relative frame of reference for one of the Diab cases. One can see

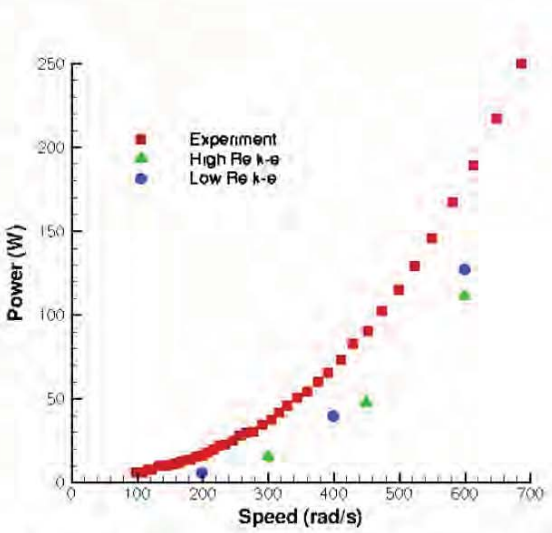



**Fig. 13 Effect of turbulence model selection on viscous work prediction.**

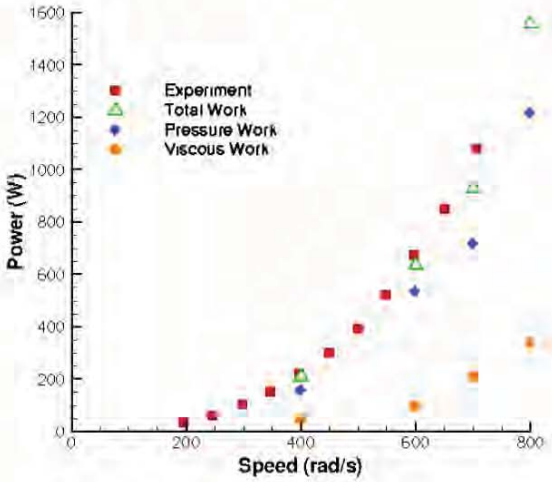



**Fig. 14 Breakdown of windage power losses for Diab 1.**

a significant vortical structure within the gear tooth region, and the tooth-to-tooth periodicity that has been achieved in the transient simulation. Figure 17 shows a view of the predicted surface pressure distributions for the four Diab gears. There, comparatively large vs. small pressures are observed on the leading and trailing tooth faces - this difference being the source of the pressure component of the spin-down torque. The highly three-dimensional nature of the flow in these relatively low aspect ratio spurs gears are also clearly seen (figure shows only 1/2 of each gear). Significant 3-D effects are also clearly visualized in Figure 18 where gear relative streamlines are displayed in the near tooth region along with an isosurface of predicted static pressure in a region of high pressure.

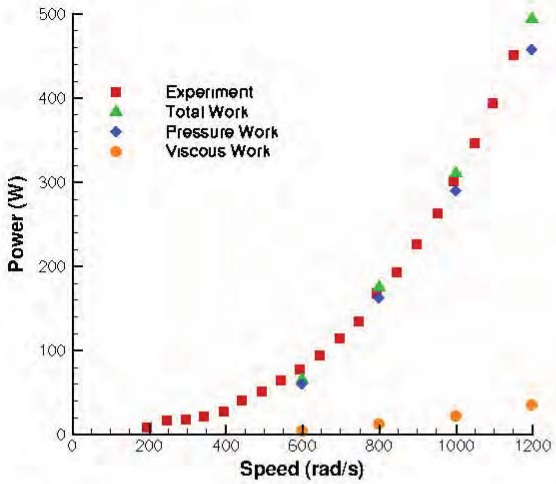



**Fig. 15 Breakdown of windage power losses for Diab 2.**

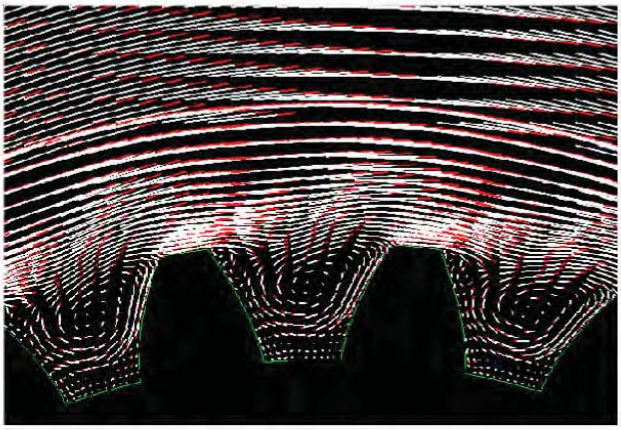


**Fig. 16 Predicted velocity vectors in the symmetry plane of a Diab gear.**

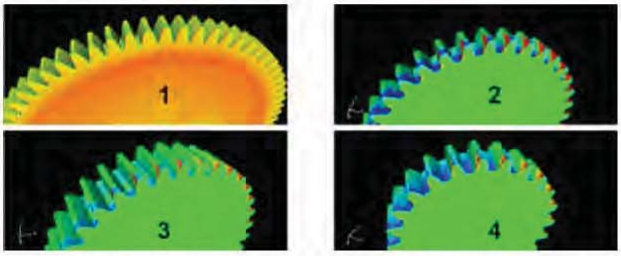



**Fig. 17 Predicted surface pressure distributions for the four Diab gears.**

*Overset Grid Simulations*

As of the writing of this paper, the overset capability for spinning gear simulations has been established in NPHASE-PSU. In this context, we are pursuing an overset verification effort and a relevant validation effort. The verification effort involves solving isolated spinning gear cases studied above using a rotating near-gear mesh and a stationary farfield mesh, the necessary approach to be used in ongoing gearbox windage activities. Figure 19 shows a view of an overset Diab Gear 4 simulation. Surface pressure

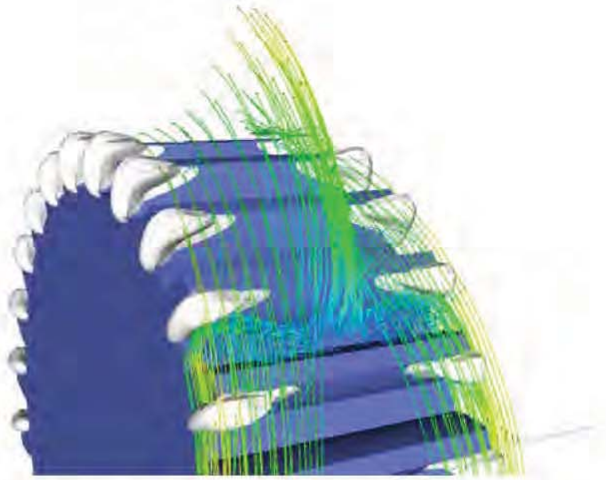


**Fig. 18 Predicted relative frame streamlines and isosurface of a high pressure region for Diab Gear 4.**

contours are displayed along with overset gear and background meshes on the symmetry plane. As expected, the CFD code returns nearly identical results for this moving grid overset simulation to the non-overset results reported above. We continue to parameterize gridding requirements in the overlap region, including the proximity of the overlap region to the gear in assessing the retained accuracy of the overset approach.

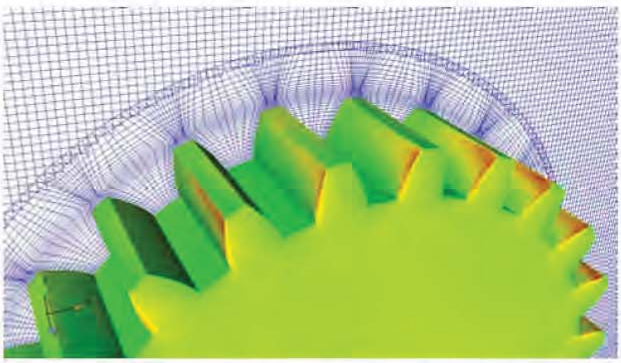


**Fig. 19 Verification study: 3-D overset grid solution and topology for Diab Gear 4.**

The validation effort under way involves simulating a series of shrouded low speed gears for which windage loss measurements were made by Dawson.[7] To date we have developed valid overset assembly grid topologies for this, as shown in Figure 20, and anticipate reporting CFD analysis for these configurations in the near future.

## Conclusion

This paper has summarized the adaptation and validation status of a CFD method for gear windage aerodynamics. Validation studies of 3-D spur gears in free space demonstrate very good agreement with published data. The following conclusions apply: 1) Viscous/turbulence modeling was identified as a shortcoming since loss power was consistently underpredicted for the viscous-drag-only spinning disk cases. 2) Low Reynolds number (sublayer resolved)

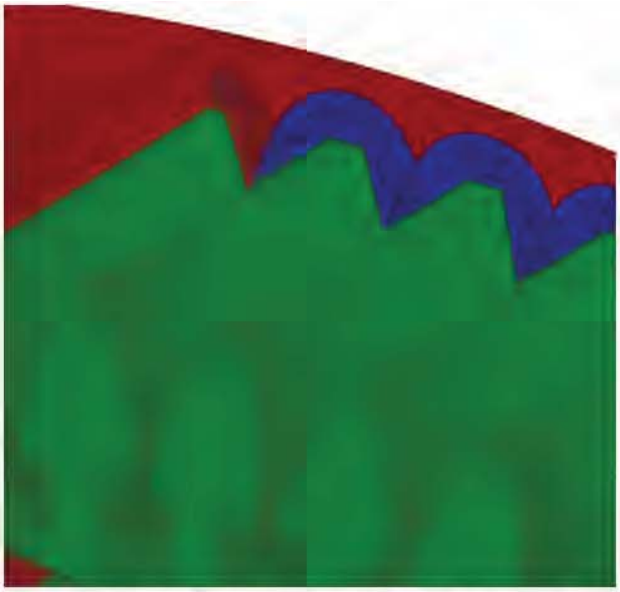


**Fig. 20 Validation study: 3-D overset grid topology for Dawson shrouded gear.**

turbulence modeling exhibited improved performance for this case. 3) High Reynolds number two equation modeling proved appropriate for modeling the moderate speed Diab spur gear suite, due apparently to the dominance of the pressure torques on spin-down. 4) The budgets of viscous and pressure components of spin-down torque suggest that viscous effects will become much more important, perhaps exceeding 50 percent, for gears with pitchline speeds approaching twice that investigated here. 5) Overset meshing is and will continue to be a critical enabler in this effort. The capability has been established and demonstrated here and will become integral in further studies where the isolated gear assumption will not be relevant.

Our continuing work in the gear windage aerodynamics area will focus on: 1) Further validation for shrouded gears, 2) Turbulence modeling including sublayer modeling, $v^2$-f modeling and/or Reynolds Stress Modeling (RSM) and/or Detached Eddy Simulation, 3) The role of laminar-turbulent transition, 4) Multi-phase flows, 5) Overset gridding in the context of gear tooth contact, 6) Industrial and NASA gearbox configuration applications, and 7) Evolution of design guidance for minimizing windage losses

## Acknowledgments

This work was supported by NASA under NASA Cooperative agreement NNX07AB34A.

## References

[1]Don D. Winfree, "Reducing Gear Windage Losses from High Speed Gears," *AME Power Transmission and Gearing Conference*, ASME, Baltimore, MD, Sept. 2000.

[2]Neil E. Anderson and Stuart H. Loewenthal, "Spur-Gear-System Efficiency at Part and Full Load," Tech. rep., NASA, 10 Dec. 1979, TP 1622 AVRADCOM TR 79-46.

[3]Neil E. Anderson and Stuart H. Loewenthal, "Effect of Geometry and Operating Conditions of Spur Gear System Power Loss," *Journal of Mechanical Design*, Vol. 103, Jan. 1981, pp. 151–159.

[4]Neil E. Anderson and Stuart H. Loewenthal, "Efficiency of Nonstandard and High Contact Ratio Involute Spur Gears," *Fourth International Power Transmission and Gearing Conference*, ASME, Cambridge, MA, Oct. 1984, NASA TM 83725 USAAVSCOM TR 84-C-9.

[5]Robert F. Handschuh and Charles J. Kilmain, "Preliminary Investigation of the Thermal Behavior of High-Speed Helical Gear Trains," *International Conference on Gears*, International Federation for the Theory of Machines and Mechanisms, Munich, Germany, March 2002, NASA/TM-2002-211336 ARL-TR-2661.

[6]Petra Heingartner and David Mba, "Determing Power Losses in the Helical Gear Mesh," *Gear Technology*, Sept. 2005.

[7]P. H. Dawson, "Windage loss in larger high-speed gears," *Proc Instn Mech Engrs*, Vol. 198A, No. 1, 1984, pp. 51–59.

[8]Y. Diab, F. Ville, P. Velex, and C. Changent, "Windage Losses in High Speed Gears - Preliminary Experimental and Theoretical Results," *Journal of Mechanical Design*, Vol. 126, Sept. 2004, pp. 903–908.

[9]K. Al-Shibl, K. Simmons, and C. N. Eastwick, "Modelling windage power loss from an enclosed spur gear," *Journal of Power and Energy*, Vol. 221, 2007, pp. 331–341, Proc. IMechE Part A.

[10]Carol N. Eastwick and Graham Johnson, "Gear Windage: A Review," *Journal of Mechanical Energy*, Vol. 130, March 2008.

[11]Florian R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, 1994, pp. 269–289.

[12]Robert F. Kunz, Wen-Sheng Yu, Steven P. Antal, and Stephen M. Ettorre, "An Unstructured Two-Fluid Model Based on the Coupled Phasic Exchange Algorithm," *15th AIAA Computational Fluid Dynamics Conference*, Norfolk, VA, June 2001, AIAA 2001-2672.

[13]S. S. Clift and P. A. Forsyth, "Linear and Non-Linear Methods for the Incompressible Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 18, 1994, pp. 229–256.

[14]J. P. Van Doormal and G. D. Raithby, "Enhancements of the SIMPLE Method for Predicting Incompressible Flows," *Numerical Heat Transfer*, Vol. 7, 1984, pp. 147–163.

[15]C. M. Rhie and W. L. Chow, "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation," *AIAA Journal*, Vol. 21, 1983, pp. 1527.

[16]P. D. Thomas and C. K. Lombard, "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, 1979, pp. 1030–1037.

[17]*PETSC*, http://acts.nersc.gov/petsc.

[18]*METIS*, http://glaros.dtc.umn.edu/gkhome/views/metis.

[19]J. A. Benek, J. Steger, and F. Dougherty, "A Flexible Grid Embedding Technique with Applications to the Euler Equations," 1983, AIAA 83-1944.

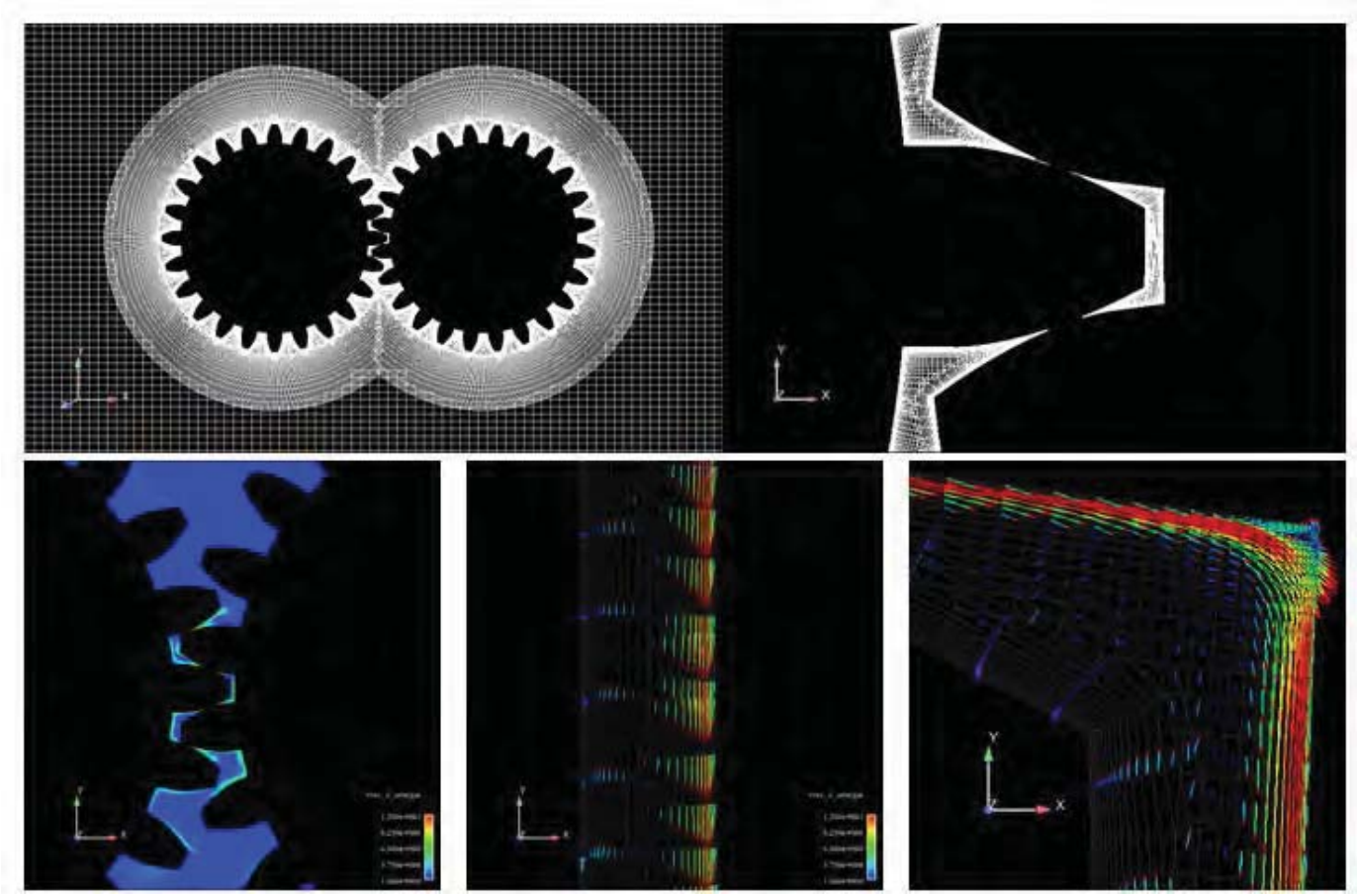
[20]Ralph W. Noack, "DiRTlib: A Library to Add an Overst Capability to Your Flow Solver," *17th AIAA Computational Fluid Dynamics Conference*, AIAA, Toronto, Ontario, Canada, June 2006, AIAA 2005-5116.

[21]Ralph W. Noack, "SUGGAR: a General Capability for Moving Body Overset Grid Assembly," *17th AIAA Computational Fluid Dynamics Conference*, AIAA, Toronto, Ontario, Canada, June 2005, AIAA 2005-5117.

[22]H. Schlichting, *Boundary Layer Theory*, McGraw-Hill, New York, 6th ed., 1968.

[23]Pointwise, *Gridgen User Manual*, version 15.10 ed., 2007.

An important and challenging element of CFD modeling for gear systems is the relative motion between, and the contact between individual gears. This challenge led us to invent/develop what we have now come to term the Hybrid Overset Immersed Boundary Method (HOIBM). The HOIBM was developed under this NRA program and was first published by us at the AIAA CFD Conference in 2009. Although that paper, AIAA-2009-3992, included many other recent overset grid technology advancements, we include below only the content of the paper that treated the HOIBM. In addition to the technical explanation and canonical test case presented in that paper, we also presented to the NASA sponsor, and VLC and VLRCOE audiences, the application of the method to spinning spur gears in contact. Some of these results are included here (although they can be best appreciated only by viewing an animation of the time accurate simulation of the gear meshing).



**Elements of HOIBM application to meshing spur gears. Grids, overset assembly and solution time snapshots.**

**(Extracted from AIAA-2009-3992):**

## VIII. Demonstration of Immersed Boundary Approach to Eliminating Orphans

The use of the Suggar++ identification of immersed boundary locations to eliminate orphans in an overset grid system will be demonstrated for two different problems using two different flow solvers. The results presented here are intended as demonstrations only and additional development and validation of the results will be presented in the future.

### A. Two-Dimensional Glottal

The first demonstration problem is a simple two-dimensional geometry. This configuration, which is illustrated in Figure 47, is a canonical model of the human glottis, for which experimental measurements are available.[19] In the experiment, the nominal "vocal folds" are closed at t=0, and a pressure difference is applied between the inflow and outflow boundaries. The "vocal folds" are then opened and closed over a period of 5.67 seconds, following a non-sinusoidal path controlled by a stepper motor. As the folds open, the $\Delta p$ across the test section drives a jet flow through the space between the folds. This test case challenges the traditional overset approach due to the point contact between the cylindrical folds, which would normally result in orphans. This motivated the mesh refinement near the contact point illustrated in the figure, so as to suitably approximate point closure.

The equations for the fluid motion was solved using the NPHASE-PSU code, an unstructured, parallel finite volume solver. Algorithmically, NPHASE-PSU follows the well-established segregated pressure based method. A colocated variable arrangement is used, and a lagged coefficient linearization is applied. In this laminar flow example, $2^{nd}$ order upwind and central discretization schemes were selected for the convection and viscous terms in the momentum equations. Continuity is introduced through a pressure correction equation based on the SIMPLE-C algorithm. In constructing element face fluxes, a Rhie-Chow momentum interpolation scheme is employed. Further details of the scheme are available in Reference 20. DirtLib and the necessary "iblanking" instrumentation for the hybrid overset-immersed method were recently installed in the code. Specifically, all out-immersed elements are nullified through appropriate source term treatments, and wall faces are inserted in the data structure at all internal faces that separate out-immersed from active elements.

Since the glottal motion was specified, all of the overset assembly files were pre-generated using SUGGAR driven by a Python motion controller. In order to accommodate the fully closed conditions (no flow, despite a driving $\Delta p$ between inlet and outlet), a constant stagnation pressure condition was specified at the inlet to match the experimentally-obtained peak jet centerline velocity. A constant reference pressure was specified at the domain outlet for this incompressible flow. A dual-time approach was used wherein 50 sub-iterations were employed for each physical time step. At each sub-iteration, the discrete momentum equations are solved approximately, followed by a more exact solution of the pressure correction equation.

Figure 47 shows a portion of the overset grid system for this case when the Glottal is partially open. The grid for the upper glottal is displayed in red, the channel grid is shown in gray, and the two lower collar grids are shown in cyan. The boundaries of all overset grids are also displayed with thick lines. Figure 48 shows the complete overset grid system when the gap is closed and the surfaces are in contact.

Figure 49 shows a closeup of the grid system near the contact point. The upper and lower Glottal grids are shown in red and blue respectively. The cyan and magenta dots are located at the element centers of fringe elements. The red and blue dots are located at the center of elements marked as OUT-IMMERSED in the upper and lower Glottal grids respectively. The faces between active and OUT-IMMERSED elements are shown as thick lines and indicate the approximate geometry being used in the solution at this time step.

Figure 50 shows a sequence of the predicted instantaneous axial velocity through one open-close cycle. The experimental configuration has a square cross section at the inlet, rather than an infinite spanwise extent assumed with the present 2D simulation. The authors are presently extending the model to capture the three-dimensionality of the geometry and the flow field.

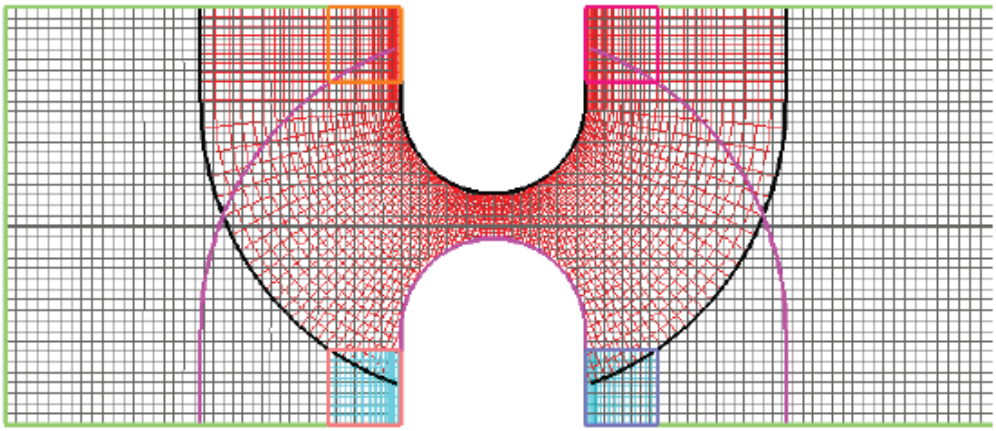


Figure 47. Portion of overset grid system for the 2D Glottal case.

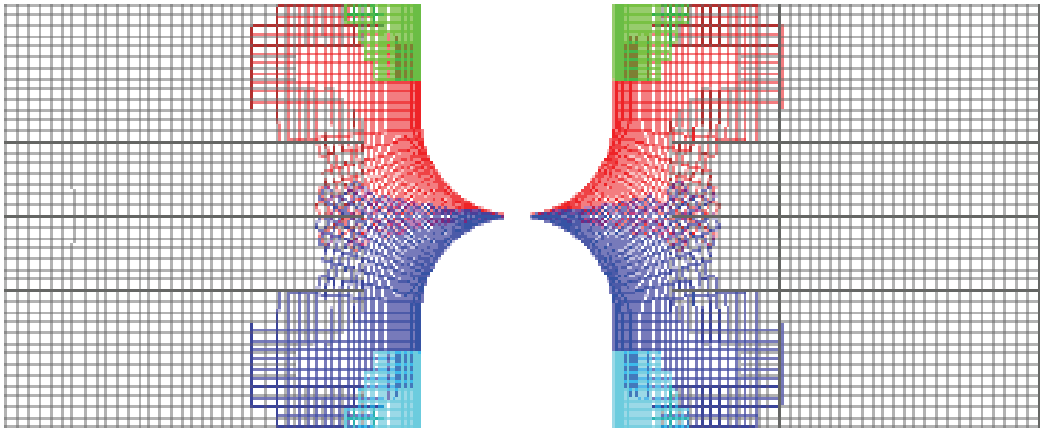


Figure 48. Complete overset grid system for the 2D Glottal case when the gap is closed.

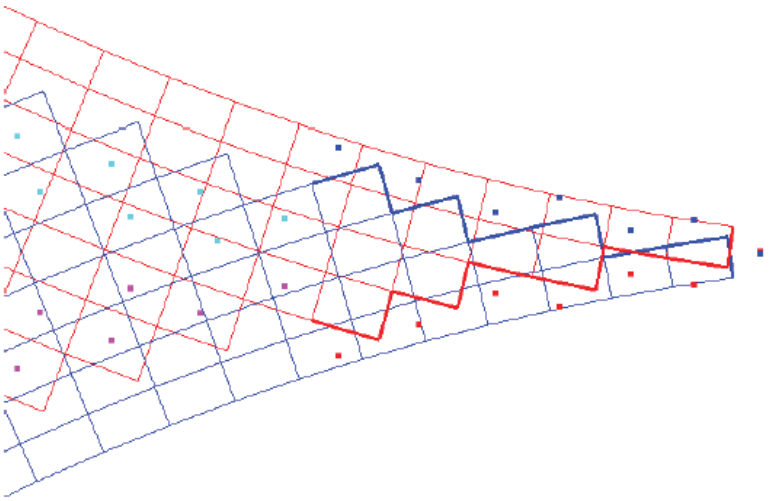


Figure 49. Closeup grid system near the contact point for the 2D Glottal case when the gap is closed.

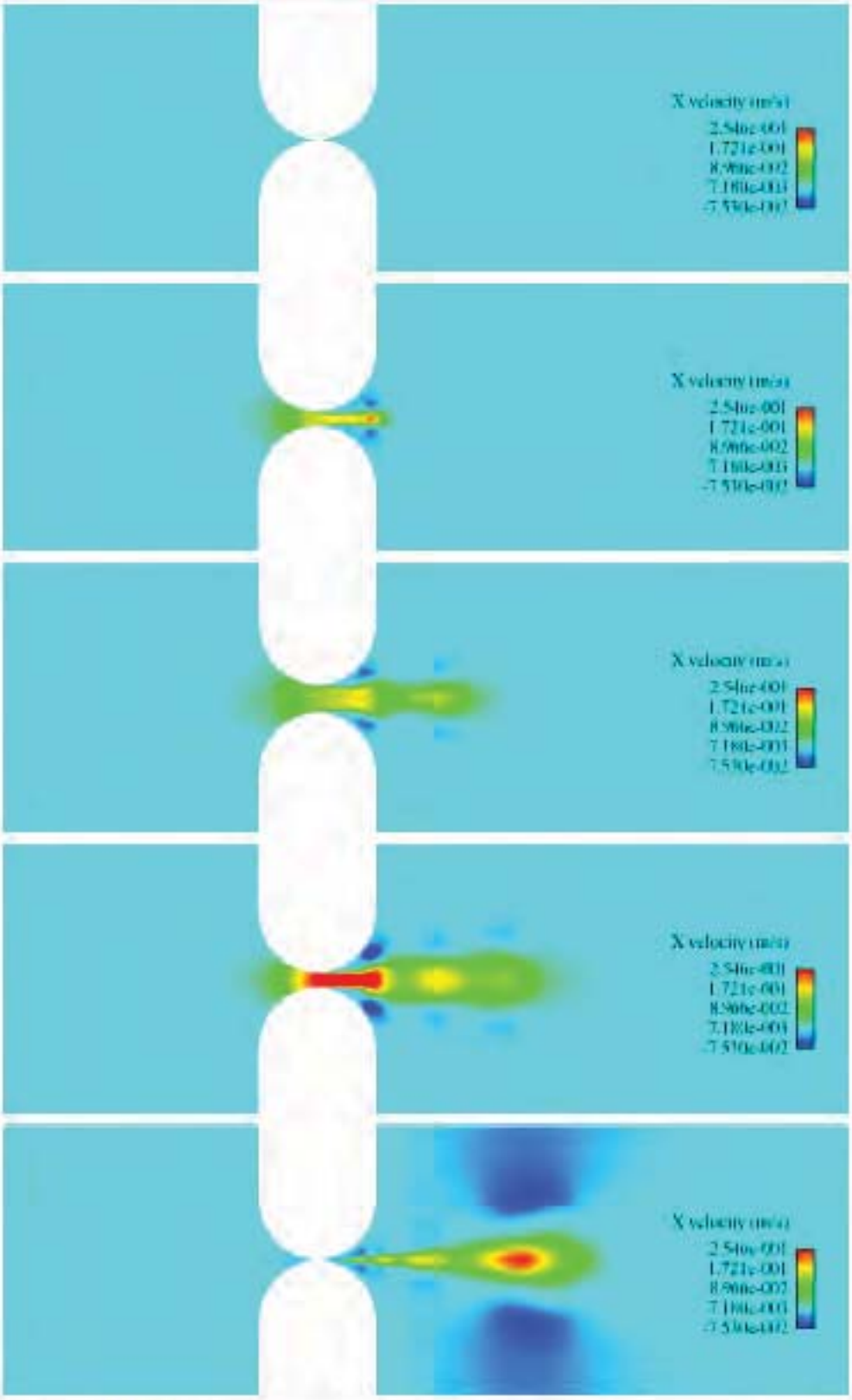


Figure 50. Sequence of the predicted instantaneous axial velocity through one open-close cycle for 2D Glottal case.

As the first phase of the NRA program wound down and we were funded by NASA to pursue a Phase 2, we turned our attention to further validation, physics interpretation/understanding and design guidance. We had for the first time available to us the recently obtained NASA Glenn Windage Facility data for validative comparison. We moved to another CFD code, OVER-REL, for this work due to its well established efficiency for single tooth periodic boundary condition capability. Parametric shroud configuration studies carried out in the Glenn experiments and the CFD analyses elucidated important physical mechanisms of windage losses as well as suggested mitigation strategies due to shrouding and newly proposed tooth contour modifications. The NASA Glenn data and CFD analyses showed that axial and radial gear shrouding are effective in significantly reducing gear windage losses both independently and when employed together. For all configurations studied, the dominant physical mechanism contributing to windage losses is the pressure field associated with diversion and impingement of the high speed relative flow on the leading tooth surface. This was a discovery not understood by the community prior to our research. It was found that shrouding mitigates the magnitude of this mechanism but not its dominance in the loss budget. Although viscous effects were identified as a secondary source of windage losses, they are not insignificant. Also, the sublayer resolved turbulence model used, although more accurate than a high Reynolds number form with wall-functions, apparently still underpredicts viscous losses for the rotational boundary layers studied. Viscous losses were found to be reduced by shrouding as well. However, small axial clearances were observed computationally to increase viscous losses compared with larger shroud clearances, suggesting that this effect could become important for even smaller clearances and at higher speeds. The CFD results showed good agreement with the Glenn experiments which exhibited quite similar loss trends to the idealized shrouded gear configurations studied computationally. T

The physics understanding that our CFD studies led to, suggested a set of possible geometric tooth modifications to further reduce windage loss. We studied these notional concepts and one of them, a spoiler or ramp protruberance appeared quite promising. Further study of this concept led to our patenting the concept. The 2010 AHS Forum paper summarized this early research. We submitted the paper to the ASME Journal of Fluids Engineering and it was accepted and appeared in March 2011. This JFE version is included here in its entirety.

**Matthew J. Hill**
Graduate Research Assistant
Department of Aerospace Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: mjh414@psu.edu

**Robert F. Kunz**
Senior Scientist
e-mail: rfk102@arl.psu.edu

**Richard B. Medvitz**
Research Associate
e-mail: rbm120@arl.psu.edu

Applied Research Laboratory,
Pennsylvania State University,
University Park, PA 16802

**Robert F. Handschuh**
Aerospace Engineer
NASA Glenn Research Center,
Cleveland, OH 44135
e-mail: robert.f.handschuh@nasa.gov

**Lyle N. Long**
Professor
Department of Aerospace Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: lnl@psu.edu

**Ralph W. Noack**
Research Associate
Applied Research Laboratory,
Pennsylvania State University,
University Park, PA 16802
e-mail: rwn10@arl.psu.edu

**Philip J. Morris**
Boeing/A.D. Welliver Professor
Department of Aerospace Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: pjmaer@engr.psu.edu

# CFD Analysis of Gear Windage Losses: Validation and Parametric Aerodynamic Studies

*A computational fluid dynamics (CFD) method has been applied to gear configurations with and without shrouding. The goals of this work have been to validate the numerical and modeling approaches used for these applications and to develop physical understanding of the aerodynamics of gear windage loss. Several spur gear geometries are considered, for which experimental data are available. Various canonical shrouding configurations and free spinning (no shroud) cases are studied. Comparisons are made with experimental data from open literature, and data recently obtained in the NASA Glenn Research Center Gear Windage Test Facility, Cleveland, OH. The results show good agreement with the experiment. The parametric shroud configuration studies carried out in the Glenn experiments and the CFD analyses elucidate the physical mechanisms of windage losses as well as mitigation strategies due to shrouding and newly proposed tooth contour modifications.* [DOI: 10.1115/1.4003681]

## 1 Introduction

Gear windage losses refer to the power losses that are a result of aerodynamic forces (pressure and viscous) acting against the rotation of high speed gears. Windage losses are a source of significant heating and fuel consumption in rotorcraft and other vertical take-off and landing (VTOL) systems. The weight and packaging constraints inherent in these systems require the gearing components to be both lightweight and heavily loaded. Attendant to this, the gears are required to operate at high rotational speeds, where windage losses become significant with respect to other gearbox losses (rolling, sliding, and lubrication losses).

A host of experimental studies of gear windage have appeared in literature [1–10]. These studies employ either isolated gears [1–5] or closed-loop gear-train systems [6–10]. Generally, these studies parametrize gear geometry elements, rotational speed, enclosure geometry, and lubrication system characteristics (flow rate, jet location, and lubricant rheology), and use dimensional analysis to develop correlations for the power losses that have proven useful in the design process.

It was shown experimentally by Dawson [1,2] and Winfree [3] that geometric modifications of the near-gear flow path, such as with shrouding, can significantly reduce both windage losses and lubricating oil consumption (80% and 40% reductions observed, respectively, in Ref. [3]). Such modifications are of significant interest in rotorcraft transmission design. Unfortunately, only a few empirical studies have been made available to date. Accordingly, there is a need for more systematic and prototypical experimental data and a need for improved understanding of the physical mechanisms involved in the schemes used to reduce these

Contributed by the Fluids Engineering Division of ASME for publication in the JOURNAL OF FLUIDS ENGINEERING. Manuscript received August 23, 2010; final manuscript received January 30, 2011; published online March 16, 2011. Assoc. Editor: Edward M. Bennett.

aerodynamic losses.

Recently, the NASA Glenn Research Center has installed a gear windage test facility to address these issues [5]. This facility is expected to produce the most comprehensive set of gearbox windage data to date, and is providing CFD validation data required in the present NASA sponsored research effort. The test facility is designed to parametrize the effects of spur gear geometry, shroud geometries, different lubrication system configurations, system pressures and temperatures, and gear meshing on windage loss. Data from both open literature data and new NASA Glenn facility are used in this paper.

CFD practitioners have also recently begun to analyze gear windage aerodynamics toward the goal of developing a more general design approach. Recent 2D studies [11] were performed using the commercial flow solver FLUENT, where a side correlation factor was used to account for 3D effects (although these authors stated that work is under way to extend their simulations to 3D [12]). Marchesse et al. [13] used a similar approach as Al-Shibl et al. [11] but used a structured grid with the flow solver ANSYS CFX. They also extended their 2D model into three dimensions, using one tooth passage with periodic boundary conditions as their computational domain. Imai et al. [14] investigated 3D bevel gears in mesh in an air-oil atmosphere, modeling the gears as porous bodies. Hill et al. [15] implemented a 3D, unstructured, overset moving mesh method and applied it to isolated spur gears in air and validated the method against data of Diab et al. [4]. Most recently, Hill [16] applied a 3D, structured, overset, steady state CFD method to elucidate the physical mechanics of how shrouding reduces windage effects on spur gears.

The validated capability developed in our earlier work [15] has motivated, among other things, the focus of the present study: to use a CFD model to interrogate the physics associated with gear windage aerodynamics. The attendant goal of this is to develop design guidance principles for methods to minimize these power losses in rotorcraft, and, potentially, other high speed gear systems. This has been and will continue to be enabled by the availability of the NASA Glenn Research Center data sets.

This paper is organized as follows. First, the governing equations and numerical procedures are reported. A series of numerical studies are then performed, which provide insight into several of the physical mechanisms of windage losses, including the role of gear shrouding. This is followed by an overview of the NASA Glenn Gear Windage Test Facility with a selection of data obtained to date. CFD simulations of the Glenn windage loss tests are subsequently presented and discussion of these comparisons is provided. Finally, the results of numerical experiments, guided by the findings described in this paper, are presented, which demonstrate a benefit in windage performance beyond that observed with shrouds alone.

## 2 Theoretical Formulation

**2.1 Governing Equations.** This paper is focused on the physical mechanisms of gear windage loss, and mitigation schemes for reducing them. The available data [1–5] show that windage losses already become significant at low subsonic tip Mach numbers. Accordingly, in this work, we restrict ourselves to incompressible analysis. Also, as will be shown, much can be learned from systems that are either shroud-free or have completely axisymmetric shrouds (i.e., fully enclosed radial shrouds), so we consider single-tooth domains that are periodic and steady in the frame of reference of the spinning gear. The conservation of mass and momentum can be written in integral conservation law form for a system rotating with constant angular velocity $\underline{\omega}$ as

$$\int_{\underline{S}} \rho \underline{V} \cdot d\underline{S} = 0 \tag{1}$$

$$\int_{\underline{S}} \rho \underline{V} \underline{V} \cdot d\underline{S} = -\int_{\underline{S}} p d\underline{S} + \int_{\underline{S}} \underline{\underline{\tau}} \cdot d\underline{S} - \int_{\forall} \rho \underline{\omega} \times (\underline{\omega} \times \underline{r}) d\forall - \int_{\forall} 2\rho(\underline{\omega} \times \underline{V}) d\forall \tag{2}$$

where $\underline{V}$ is the velocity vector in the relative frame of reference. It was shown in Ref. [15] that a sublayer resolved two-equation turbulence model performed better than a high Reynolds number form with wall-functions in predicting viscous losses on a spinning disk. Accordingly, in this work, we adopt $\underline{\underline{\tau}} = (\mu + \mu_t)(\nabla \underline{V} + (\nabla \underline{V})^T)$ and a sublayer resolved $q$-$\omega$ turbulence model due to Coakley [17] is used. The dependent variables in this model are related to the turbulence kinetic energy $k$ and the turbulence dissipation rate $\varepsilon$ through $q = \sqrt{k}$ and $\omega = \varepsilon / k$. In this model, the eddy viscosity is obtained from

$$\mu_t = \rho C_\mu D q^2 / \omega \tag{3}$$

where $C_\mu = 0.09$ and $D$ is the near-wall damping function

$$D = 1 - e^{-\alpha \rho q d_n / \mu} \tag{4}$$

where $\alpha = 0.02$ and $d_n$ is a measure of the normal distance to the nearest wall. The modeled transport equations for $q$ and $\omega$ are

$$\int_{\underline{S}} \rho q \underline{V} \cdot d\underline{S} = \int_{\underline{S}} \left( \mu + \frac{\mu_t}{\mathrm{Pr}_q} \right) \nabla q \cdot d\underline{S} + \int_{\forall} \frac{\rho q}{2} \left( C_\mu D \frac{S}{\omega} - \omega \right) d\forall \tag{5}$$

$$\int_{\underline{S}} \rho \omega \underline{V} \cdot d\underline{S} = \int_{\underline{S}} \left( \mu + \frac{\mu_t}{\mathrm{Pr}_\omega} \right) \nabla \omega \cdot d\underline{S} + \int_{\forall} \rho (C_1 C_\mu S - C_2 \omega^2) d\forall \tag{6}$$

### 2.2 CFD Code and Numerics

*2.2.1 OVER-REL.* OVER-REL is a parallel, cell-centered, incompressible, finite-volume code based on structured overset multiblock grids and the time-marching, pseudocompressibility formulation of Chorin [18]. Inviscid fluxes are formulated from the Roe-approximate Riemann solver [19] and extended to third-order accuracy through the monotone upstream-centered scheme for conservation laws (MUSCL) scheme [20]. Second-order accurate central differences are utilized for the viscous fluxes. Numerical derivatives are used to calculate the flux Jacobians. A symmetric Gauss–Seidel method is applied to solve the resulting linear system of equations. In the present application, the code's turbomachinery analysis instrumentation is employed; all simulations are carried out for a single-gear tooth, with periodic boundary conditions, in a noninertial frame of reference rotating with the gear.

*2.2.2 Overset Meshing: SUGGAR and DIRTLIB.* The tight shroud clearances studied lead to mesh topology constraints that, in turn, lead to poor quality block-structured meshes. OVER-REL has overset meshing capability, which enables high quality meshes for these configurations. The overset assembly process is performed using the SUGGAR code [21], a general overset grid assembly code with the capability to create the domain connectivity information (DCI) at node and/or element centers for general grid topologies. For static grid assemblies with no motion between component grids (as here), the grid assembly is a preprocessing step. The intergrid interpolations produced by SUGGAR use an unweighted least square procedure.

OVER-REL is instrumented with DIRTLIB [22], a solver neutral library that encapsulates the functionality required by the solver to utilize the overset DCI produced by SUGGAR. DIRTLIB is independent of the CFD solver's data structures, and can therefore be used with any solver. The CFD code calls a few functions to initialize the library, load the DCI interpolation, transfer the data to appro-

Table 1 Ref. [4], gear properties

| | Pitch diameter (mm) | Tooth width (mm) | Module (mm) | Teeth |
|---|---|---|---|---|
| Gear 1 | 288 | 30 | 4 | 72 |
| Gear 2 | 144 | 30 | 4 | 36 |
| Gear 3 | 144 | 60 | 4 | 36 |
| Gear 4 | 144 | 60 | 6 | 24 |
| Disk | 300 | 30 | | |

priate processors in a parallel execution environment, and apply the interpolated data as boundary values at intergrid boundary points.

## 3 Results

**3.1 Loss Mechanisms and Budgets in Gear Windage.** In their 2004 paper, Diab et al. [4] studied four spur gears and a disk spinning freely in air (i.e., no shrouding). The gears varied in diameter, width, and tooth module. Table 1 summarizes the gear properties. In the present work, free spinning simulations were first carried out for Diab gear 1 and the disk using OVER-REL. Nonoverset multiblock-structured meshes were employed for both, as shown in Fig. 1. Near-wall grids were constructed to return wall cell $y^+$ values $<1$ and wall normal stretching ratios $<1.2$ everywhere in order to adequately resolve the high Reynolds number boundary layers that arise. The grid topologies, near-wall grid spacing, and grid stretching ratios were maintained as closely as possible between the gear and disk meshes. Per the periodic boundary conditions employed, one tooth passage ($2\pi/72$) is modeled for both configurations. In order to stably time-march the OVER-REL solution, a very small inflow velocity and a pressure outflow boundary were included adjacent to the maximum and minimum axial boundaries upstream and downstream of the rotating elements. This artificial through-flow velocity was successively reduced to where no perceptible change in loss values was returned.

Figure 2 shows OVER-REL results for gear 1 with experimental data and previous CFD results by the authors [15]. The OVER-REL results show very good agreement with both experiment and previous simulations. Figure 3 shows that for gear 1, the pressure torque associated with the integrated pressure difference between leading and trailing tooth surfaces dominates the loss budget. As the rotation rate increases, viscous losses remain a nearly constant fraction of total loss (10%). Figure 4 shows a comparison of loss results between gear 1 and the disk. The total losses are much smaller for the disk and are due to viscous shear alone.

In Fig. 5, the torque per unit span contributed by viscous shear is compared for the disk (up to its outer radius) and gear 1 (up to its base radius). The geometry of these systems requires that the all pressure spin-down torques are due to the pressure differences between the leading and trailing tooth surfaces. Figure 5 illustrates

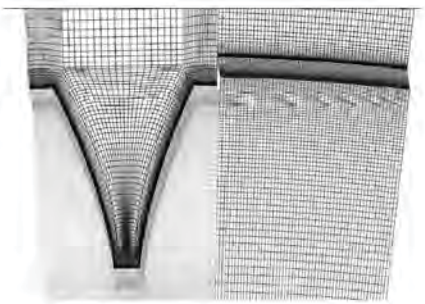


Fig. 1 Comparison of Diab gear 1 and disk grids

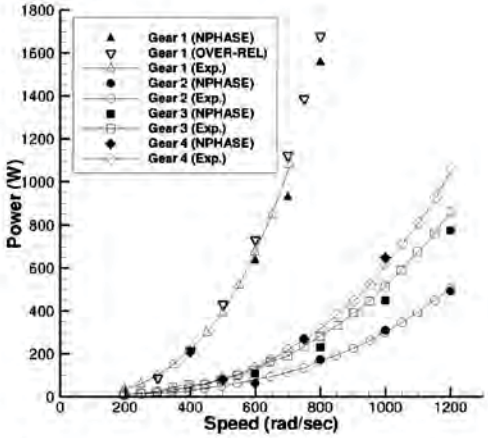



Fig. 2 Comparison of results from experiment [4], NPHASE-PSU [15], and OVER-REL for Diab gears 1–4

that the viscous losses are very similar between these configurations (indeed, the small differences in predicted viscous power values are due to very small grid differences) showing, conclusively, that 3D effects (i.e., nonaxisymmetric) associated with

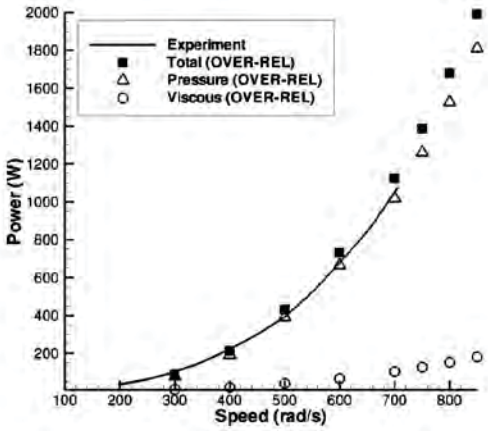



Fig. 3 Comparison of results from experiment [4] and OVER-REL for Diab gear 1. Viscous and pressure loss budgets are included.

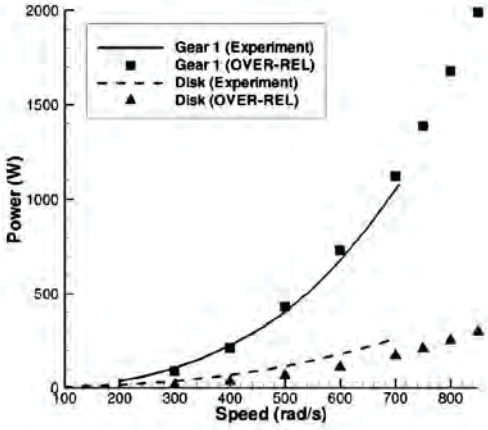



Fig. 4 Comparison of Diab gear 1 and disk measurements and OVER-REL solutions

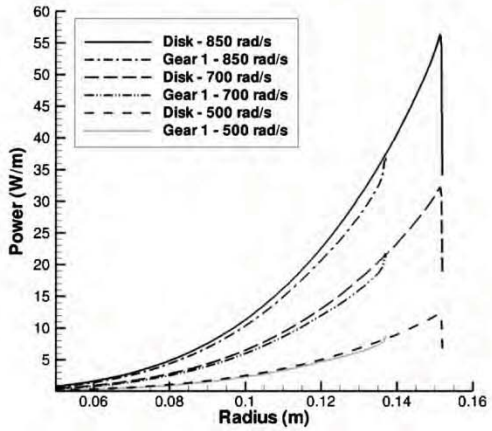



**Fig. 5 Comparison of predicted torque per unit span contributed by viscous shear for the Diab disk (up to its outer radius) and gear 1 (up to its base radius)**

pressure forces are almost completely responsible for the significantly greater losses for the gear.

The physical mechanisms associated with the dominant pressure torque can be studied by interrogating the CFD results for gear 1. In Figs. 6–8, several 3D visualizations are presented for the 850 rad/s simulation. In Fig. 6, a number of relative frame of reference streamlines, colored by local static pressure, are displayed. These streamlines are seeded close to the gear face and teeth and integrated in both directions. Some of the high speed (in the relative frame) tangential flow near the gear face plane is diverted into the tooth passage, where strong secondary flows are evident. A stagnation region is observed on the leading tooth surface, where the diverted relative flow impinges near the gear face.

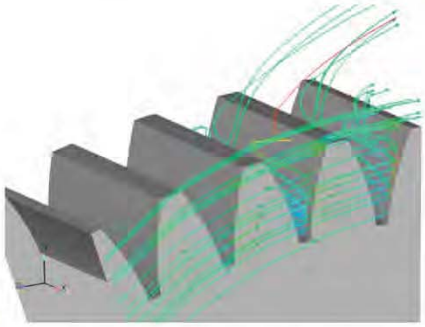


**Fig. 6 Three-dimensional relative frame streamlines colored by static pressure for Diab gear 1 at 850 rad/s**

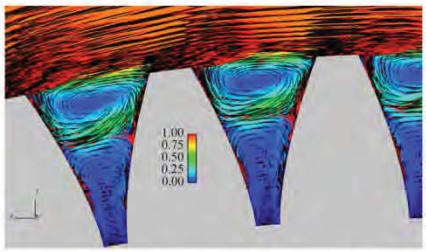



**Fig. 7 Axial projection of velocity vectors halfway between gear face and gear centerline for Diab gear 1 at 850 rad/s. Vector density of 0.5. Background contours of local normalized projected relative velocity magnitude.**

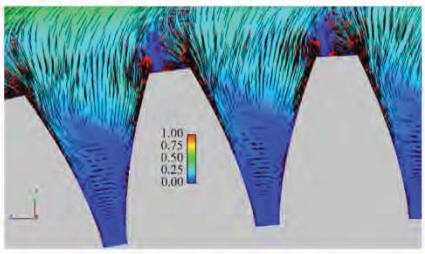



**Fig. 8 Axial projection of velocity vectors near-gear centerline for Diab gear 1 at 850 rad/s. Vector density of 0.5. Background contours of local normalized projected relative velocity magnitude.**

By symmetry, this axial transport arises on both sides of the gear and therefore leads to impingement of oppositely directed flow and radial ejection of momentum near the gear centerline. An axial projection of relative velocity vectors is shown in Fig. 7 at a plane halfway between the gear face and the gear centerline. A vector density of 0.5 (vector plotted at every other grid point) is applied for clarity. Contours of local normalized projected relative velocity magnitude are included ($V^* = \sqrt{V_y^2 + V_z^2}/\omega r$). Two counter-rotating passage vortices are present. Peak normalized secondary velocity magnitudes near $1/2\omega r$ are observed, indicating the strength of these secondary motions. The flow at this axial location is reminiscent of a rearward facing step and/or cavity flow with attendant vortical recirculation regions. Figure 8 shows the same plot but near the gear centerline. Here, one sees the very significant radial ejection quite clearly. The flow has a component directed upstream (against the relative flow) near the leading surface at the tip radius. The magnitude of the ejection flow induces significant blockage and we see values of $V^*$ much less than 1 well beyond the tip radius.

As seen above, some of the high speed tangential flow near the gear face plane is diverted into the tooth passage and a stagnation region appears, where this flow impinges on the leading surface, near the gear face. The net torque due to pressure effects can be represented by the difference between the leading and trailing surface pressure coefficients ($C_P = (p - p_\infty)/0.5\rho V_{tip}^2$), which is shown in Fig. 9. Clearly, the net torque is dominated by the near-

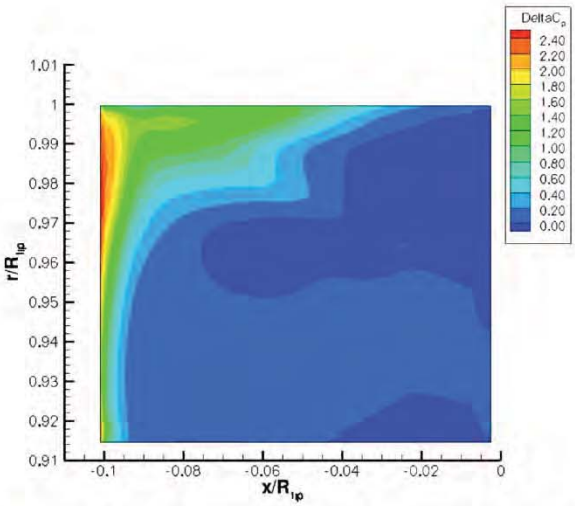



**Fig. 9 $\Delta C_P$ between gear leading and trailing tooth surfaces (one-half of symmetrical gear shown) for Diab gear 1 at 850 rad/s**

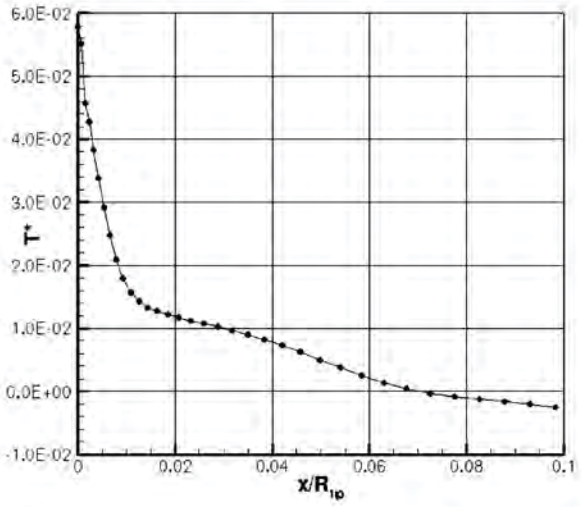



**Fig. 10 Pressure torque per unit width versus axial coordinate (one-half of symmetrical gear shown) for Diab gear 1 at 850 rad/s**

face impingement observed in Fig. 6. In Fig. 10, the torque per unit width is plotted versus distance from the gear face. Here, torque is nondimensionalized as

$$T^*(x) = \frac{\int_{r_{\text{inner}}}^{r_{\text{tip}}} \Delta p \, dA_\theta \, dr}{dx} \left[ \frac{1}{\frac{1}{2}\rho V_{\text{ref}}^2 L_{\text{ref}}^2} \right] \tag{7}$$

where $\Delta p$ is the pressure difference between the grid faces on the leading and trailing surfaces (which have identical $x$-$r$ vertex coordinates), $dA_\theta$ is the tangential projection of the area of the grid face, $r$ is the radial coordinate of the grid face centroid, and reference length $L_{\text{ref}}$ is the gear tip radius. We see that, indeed, it is the near-face region that dominates the pressure windage loss torque.

**3.2 Gear Shrouding.** As mentioned in the introduction, several researchers measured improvements in gear windage loss performance when shrouds of various configurations were employed [1–3,5]. In this section, the aerodynamics of shrouding in the context of geometrically simple spur gear is studied and several conclusions are drawn, which may impact design considerations. The Diab gear 1 configuration is studied with four shrouding arrangements, nominally, large-axial-large-radial, small-axial-large-radial, large-axial-small-radial, and small-axial-small-radial. Table 2 and Fig. 11 quantify and illustrate the shroud dimensions, respectively. These four shroud configurations were chosen to be representative of the extrema of the full-shroud NASA Glenn tests detailed further below. The smaller axial and radial shroud clearances gave rise to mesh topology constraints that, in turn, led to poor quality block-structured meshes, unless overset meshes are used, as shown in Fig. 12.

In Fig. 13, the predicted windage loss versus rotation rate is plotted for the unshrouded case, validated and studied above, and

**Table 2 Diab shroud clearances**

| | Minimum | Maximum |
|---|---|---|
| Axial ($/R_{\text{tip}}$) | 0.0044 | 0.1733 |
| Radial ($/R_{\text{tip}}$) | 0.0044 | 0.0970 |

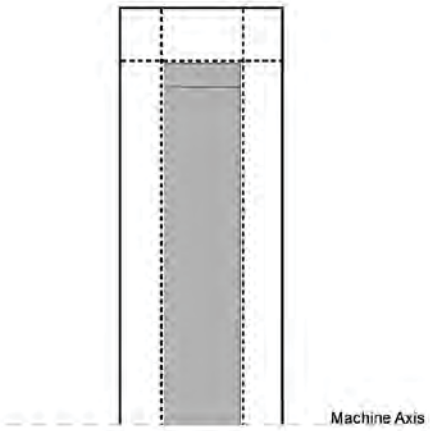



**Fig. 11 Four notional shroud configurations for Diab gear 1 geometry. Figure is to scale. The gray region defines the gear. The solid black lines indicate the positions of the large-axial and large-radial shroud walls. The dashed lines indicate the positions of the small-axial and small-radial shroud walls.**

**Fig. 12 Cross section of overset mesh topology for the small-radial shroud**

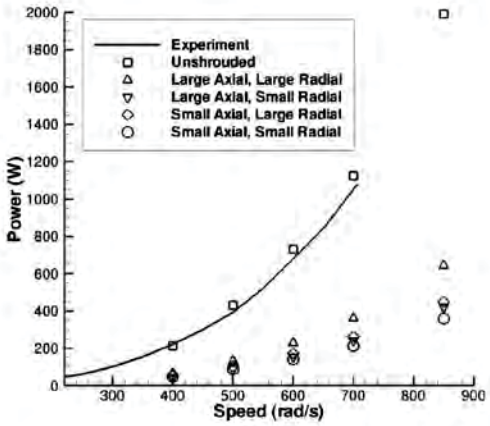



**Fig. 13 Comparison of predicted windage losses between free spinning Diab gear 1 and four shrouded configurations**

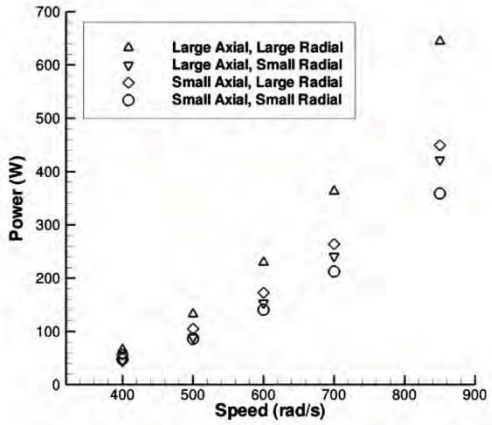



**Fig. 14 Comparison of predicted windage losses between four shrouded Diab gear 1 configurations**

the four shrouded configurations. All four of the shrouds give rise to very significant improvements in windage losses. The large-axial-large-radial shroud provides a 68% decrease in loss at 850 rad/s; the small-axial-small-radial shroud provides approximately an 81% decrease at this speed. Indeed, distinguishing the performance gains between the shrouded cases is facilitated by excluding the free spinning results, as in Fig. 14. Examining this plot, we see that reducing the axial and radial clearances from large to small provide approximately the same level of additional benefit over the large-axial-large-radial case, with the reduction of the radial clearance, providing somewhat more benefit in this particular case. Applying both clearance reductions together provides the maximum benefit.

Figure 15 shows the same pressure coefficient plot reported above but here, for the large-axial-large-radial shroud case, it is again at 850 rad/s. A highly edge loaded $\Delta C_P$ between leading and trailing surfaces can be observed as with the unshrouded case but with a very reduced range of $\Delta C_P$, which, of course, leads to reduced torque. Figure 16 shows contours of pressure and surface shear stress lines on the leading surface for the four shrouded cases at 850 rad/s. It can be seen that the general features of impingement/stagnation near the face remain present for all of the

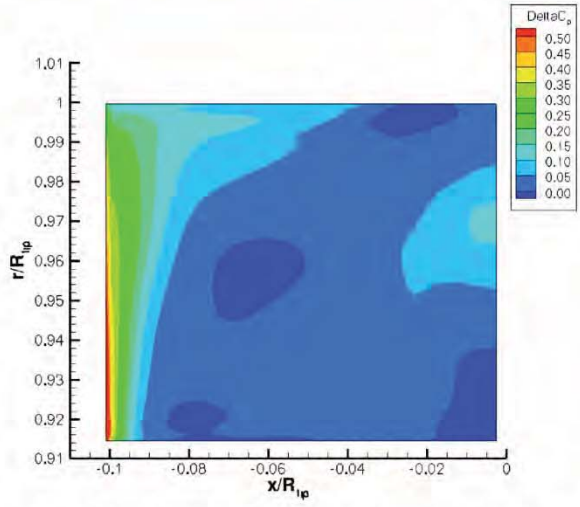



**Fig. 15 $\Delta C_P$ between gear leading and trailing tooth surfaces (one-half of symmetrical gear shown). Diab gear 1, large-axial-large-radial shroud, 850 rad/s.**

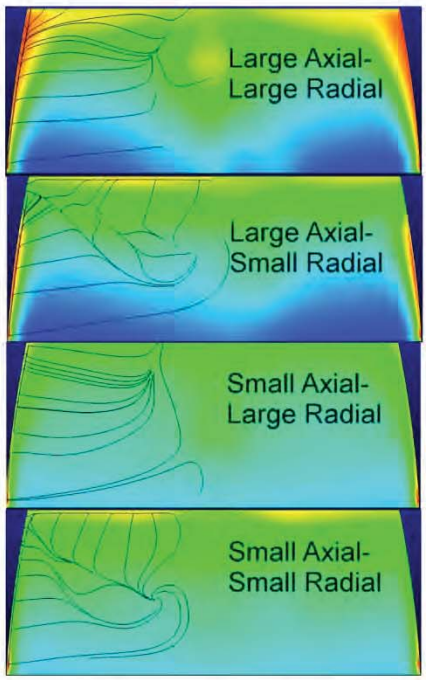



**Fig. 16 Contours of pressure and surface shear stress lines on the leading surface for the four shrouded cases at 850 rad/s**

configurations.

The torque per unit width is plotted versus distance from the gear face for all four shrouds at 850 rad/s in Fig. 17. All four configurations exhibit edge loaded profiles, with their integral consistent with the net loss trend reported in Fig. 14. Comparison with Fig. 10 shows the dramatic reduction in pressure torque that has been achieved for these configurations but with retention of the basic features of the pressure torque distribution.

An exploration of viscous losses was performed on the shrouded Diab gears. Figure 5 illustrated that the viscous losses per unit span for the unshrouded case increased with rotation speed and radius. The viscous losses per unit span are plotted for each of the four shrouded cases and the unshrouded case at 850

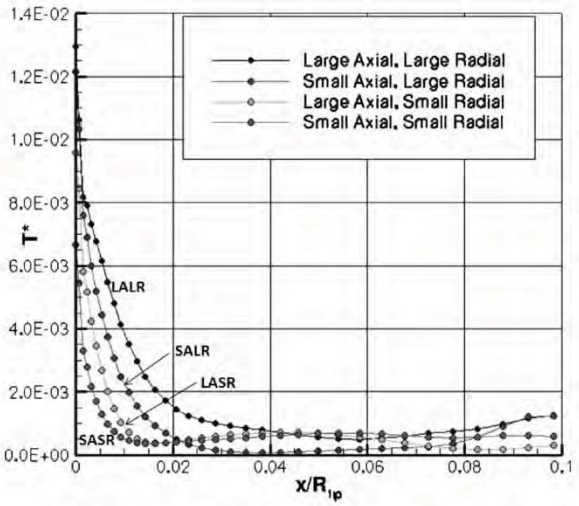



**Fig. 17 Plot of torque per unit width from the gear face for the four shrouded cases**

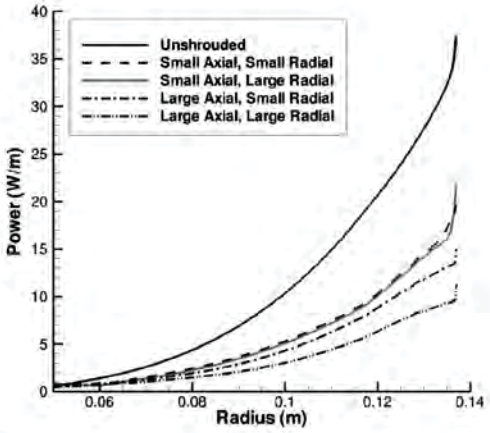



**Fig. 18 Comparison of viscous torque per unit span for Diab gear 1 at 850 rad/s for the unshrouded and four shrouded configurations**

rad/s in Fig. 18. It is observed that face shear is reduced for all of the shrouded cases due to the "couette-flowlike" rotational boundary layer that arises between the gear face and the axial shroud. Also, of interest in this figure are the greater viscous losses for the small-axial shrouds compared with the large-axial shrouds. These losses are still less than the unshrouded case but suggest that viscous losses can increase as a percentage of total loss for very small axial shroud clearances.

## 4 NASA Glenn Gear Windage Test Facility

The NASA Glenn Research Center has recently installed a gear windage test facility [5]. The test facility is designed to parametrize the effects of gear geometry, shroud geometries and sizes, different lubrication system configurations, system pressures and temperatures, and gear meshing on windage loss. A sketch of the test facility is shown in Fig. 19. The facility has a 112 kW (150 hp) dc drive motor that is connected to a 5.7:1 speed-up gearbox. The output of the speed-up gearbox is then connected to a torque meter prior to a coupling connection of the input shaft to the test gearbox. The input and output shafts have hydraulically operated clutches that allow the facility (in single or dual shaft mode) to be disconnected from the power applier and/or magnetic brake attached to the output shaft. With the speed capability of the drive

**Table 3 Basic gear dimensions [5]**

| | |
|---|---|
| No. of teeth | 52 |
| Module/pitch, mm (1/in.) | 6.35 (4) |
| Face width, mm (in.) | 28.4 (1.12) |
| Pitch diameter, mm (in.) | 330.2 (13.0) |
| Pressure angle, deg | 25.0 |
| Outside diameter, mm (in.) | 342.65 (13.49) |

**Table 4 Shroud wall clearances studied [5]**

| | Minimum | Maximum |
|---|---|---|
| Axial, mm (in.) | 0.762 (0.030) | 29.718 (1.17) |
| Radial, mm (in.) | 0.762 (0.030) | 16.64 (0.655) |

motor and speed increasing gearbox and the dimensions of the test specimen, the pitch line velocity can be taken to 280 m/s (55,000 ft/min).

The test gears can be run in the gearbox with or without shrouding. The shrouding clearance can be adjusted both radially and axially. This work reports data from the first series of experiments: air-only, single-gear, and fully enclosed axial and radial shrouds. Details of the gear under test in this report are given in Table 3 and the maximum and minimum shroud clearances are provided in Table 4. The gear has some modest geometric complexities compared with the idealized Diab gear studied above. These include chamfered teeth, filleted teeth roots, and narrower body width between the teeth and the hub. This work reports experimental data CFD results for the four extreme configurations: large-axial-large-radial, large-axial-small-radial, small-axial-large-radial, and small-axial-small-radial. A sketch of the shrouding arrangement is shown in Fig. 20.

Data from the NASA Glenn Gear Windage Test Facility were measured in the following manner. Speed data are measured using inductive pickups that read a 60-tooth disk on the end of each of the shafts. The output from the sensor (pulse/sine wave) is sent to a frequency to voltage converter. The output from the converter is then sent to a National Instruments card and read by LABVIEW. Data were taken at 10 Hz. The facility was operated at a series of increasing drive motor speeds. At each of these conditions, several different data were collected. The drive motor speed, torque applied to rotate the test hardware, internal shroud (fling-off) temperature, and internal shroud static pressure data were taken at steady drive motor speed conditions. Data were taken and then the speed was incremented up a given drive motor speed from 500

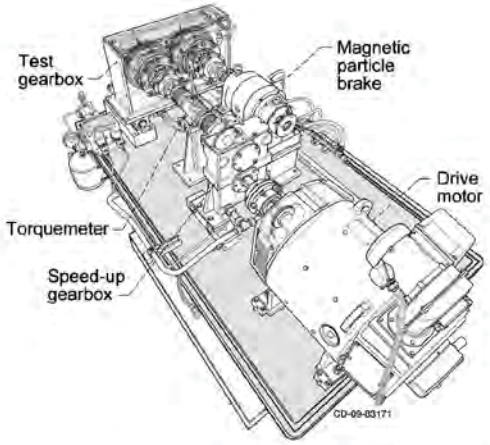



**Fig. 19 Overview sketch of the NASA Glenn Research Center Gear Windage Test Facility [5]**

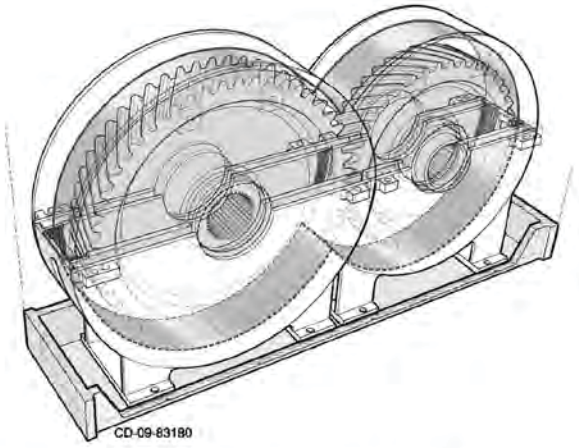



**Fig. 20 Shroud assembly for test facility [5]**

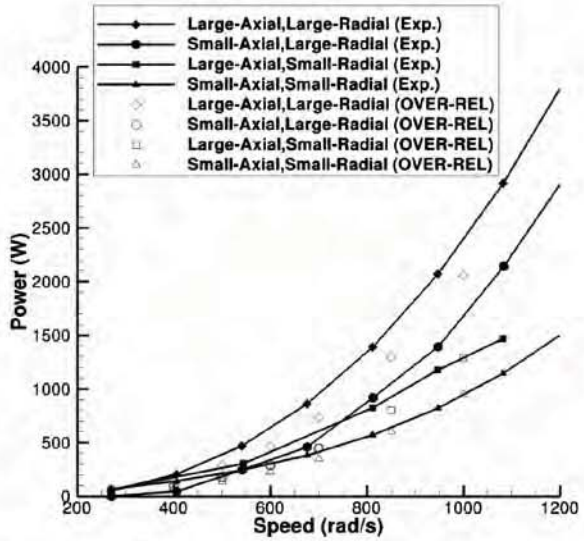



**Fig. 21 Experimental [5] and CFD results for the 13″ pitch diameter NASA spur gear**

rpm to 3125 rpm (or 2587–16,168 rpm of the gear shaft). The data were taken from motor controller speed, a commercially available torque meter for torque, thermocouple for inside shroud temperature (fling-off), and manometer. In order to determine the effects of the gear-only windage, a separate test was conducted with the entire system in place minus the gear.

## 5 Experimental and Computational Results for the Glenn Facility Tests

Grids were developed for the 13 in. (330.2 mm) pitch diameter spur gear configuration using similar topologies as the Diab cases. Four shroud configurations have been selected for study. Table 4 lists the four baffle configurations studied computationally.

Figure 21 shows comparisons of the NASA Glenn data and OVER-REL predictions for the four shroud configurations. The data show the same trends as the idealized Diab gear CFD studies reported above. Specifically, the large-axial-large-radial shrouding exhibits the highest loss levels, and the small-axial-small-radial shrouding exhibits the lowest loss levels. The benefit realized by reducing both clearances is somewhat more substantial than for the Diab case. The CFD results are seen here to provide fairly good agreement with measured values. An interesting observation in the CFD results is that the large-axial-small-radial and small-axial-large-radial results are nearly identical along the entire speed line.

The qualitative correspondence between the NASA Glenn and idealized Diab cases presented earlier suggests that the same physical loss mechanisms are acting. Figure 22 shows a view of predicted relative streamlines colored by pressure for the large-axial-large-radial shroud case at 700 rad/s. This image exhibits several of the features observed for the idealized case, including diversion of the near-face flow into the tooth passage and impingement on the leading surface, strong axial secondary vorticity in the tooth passage, strong ejection of this flow near the tooth centerline, and radial flow of the near-face streamlines below the base of the teeth.

## 6 Design Alternatives

It was demonstrated above that axial and radial shrouding can reduce windage losses. Some of the physics of these loss reduction schemes were studied there. Despite the experimentally and computationally observed differences in loss magnitudes between

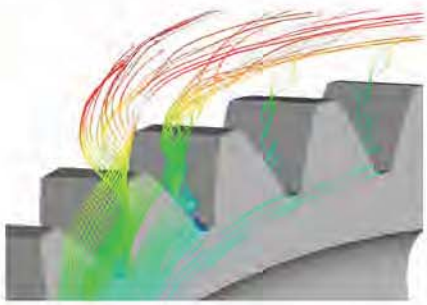


**Fig. 22 Three-dimensional relative frame streamlines colored by static pressure for the NASA 13″ pitch diameter spur gear, large-axial-large-radial shroud, at 700 rad/s**

unshrouded and various shrouded configurations, in all cases, a significant component of the torques associated with windage arose from impingement onto the leading surface of the high velocity relative frame flow drawn into the tooth passage. Accordingly, in this section, we return to the geometrically idealized Diab gear 1 in the large-axial-large-radial shroud configuration, and experiment numerically with four proposed tooth geometry modifications aimed at mitigating this impingement and attendant windage torque. Figures 23–27 show an oblique view of the baseline geometry and the four alternative geometries considered: (1) leading surface tooth-edge rounding, (2) leading+trailing surface tooth-edge rounding, (3) double slots on the top of the teeth, and (4) trailing surface ramp.

Figure 28 shows a comparison of these four simulations with the baseline large-axial-large-radial case. The leading surface rounding and double slot geometries return nearly identical windage loss. The leading+trailing surface rounding returns somewhat higher loss. However, the net loss obtained using the trailing surface ramp is approximately 30% lower than the baseline configuration. The torque per unit width for the five geometries are plotted in Fig. 29. There, it can be seen that the ramp configuration

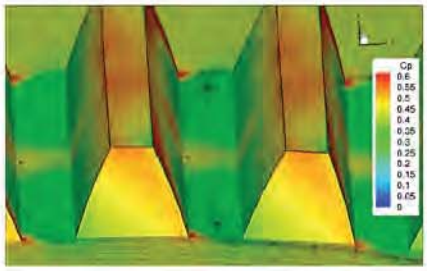



**Fig. 23 Predicted surface pressure coefficient and skin friction lines for baseline tooth geometry**

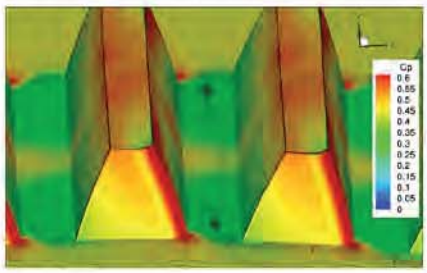



**Fig. 24 Predicted surface pressure coefficient and skin friction lines for tooth geometry alternative: leading surface rounding**

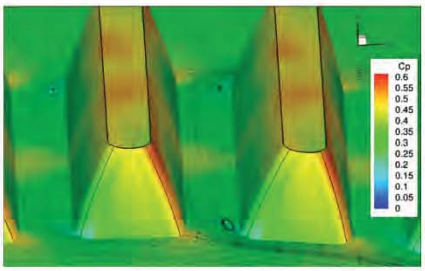



**Fig. 25 Predicted surface pressure coefficient and skin friction lines for tooth geometry alternative: leading+trailing surface rounding**

exhibits much smaller torques within the tooth channel and this clearly results in the reduced integrated loss for the entire gear.

To further elucidate the physics involved in these numerical studies, we return to Figs. 24–27. In each of these figures, predicted surface pressure coefficient contours are plotted along with selected surface skin friction lines. The baseline and two rounded geometries exhibit largely same qualitative flow features, with the rounded cases "smearing" the leading face impingement and trailing face detachment gradients. The tooth slots were conceived to "flush" the peak axial vorticity/low pressure regions with higher velocity incoming relative flow, thereby reducing the axial pressure gradient and thereby diverting the high relative velocity near-face flow into the passage. This appears to not have achieved the desired result. The trailing surface ramp geometry did have a significant impact on the aerodynamics. Specifically, the relative flow near the face is turned away from the gear. This turning induces a local pressure rise on the ramp, which contributes to windage torque. However, this flow has been diverted away from the tooth enough that subsequent diversion of this flow into the tooth passage has been virtually eliminated, resulting in almost no pressure rise on the leading surface. This gives rise to much less net torque shown in Fig. 29, and so the improved net performance of the ramp configuration observed in Fig. 28 is clearly due to the reduced integrated tooth passage torque more than offsetting the increased torque associated with the ramp turning itself.

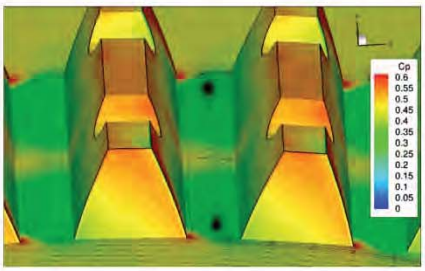



**Fig. 26 Predicted surface pressure coefficient and skin friction lines for tooth geometry alternative: tooth slot**

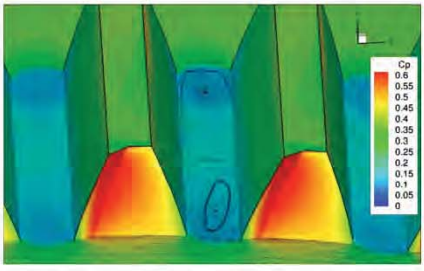



**Fig. 27 Predicted surface pressure coefficient and skin friction lines for tooth geometry alternative: trailing surface ramp**

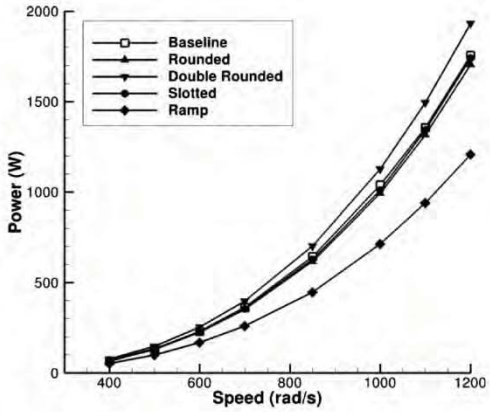



**Fig. 28 Windage loss predictions for the baseline tooth geometry (Diab gear 1 with large-axial-large-radial shroud) and four geometric alternatives**

## 7 Conclusion

This paper has summarized a number of CFD studies focused on the aerodynamics of gear windage losses. The goals of this work have been to validate the numerical and modeling approaches used for these applications and to develop physical understanding of the aerodynamics of gear windage loss. Comparisons are made with experimental data from open literature, and data recently obtained in the NASA Glenn Research Center Gear Windage Test Facility. Parametric shroud configuration studies carried out in the Glenn experiments and the CFD analyses elucidate important physical mechanisms of windage losses as well as mitigation strategies due to shrouding and newly proposed tooth contour modifications.

The following conclusions apply. (1) The NASA Glenn data and CFD analyses show that axial and radial gear shrouding are effective in significantly reducing gear windage losses both independently and when employed together. (2) For all configurations studied, the dominant physical mechanism contributing to wind-

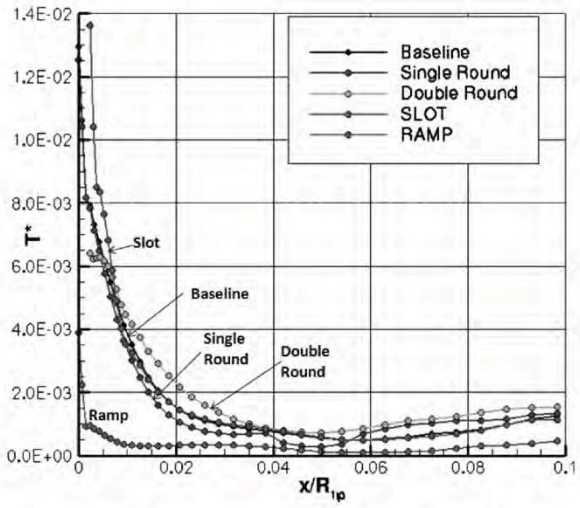



**Fig. 29 Torque per unit width predictions for the baseline tooth geometry (Diab gear 1 with large-axial-large-radial shroud) and four geometric alternatives at 850 rad/s**

age losses is the pressure field associated with diversion and impingement of the high speed relative flow on the leading tooth surface. (3) Shrouding mitigates the magnitude of this mechanism but not its dominance in the loss budget. (4) Although viscous effects have been identified as a secondary source of windage losses, they are not insignificant. Also, it was identified here and in Refs. [15,16] that the sublayer resolved turbulence model used here is still more accurate than a high Reynolds number form with wall-functions but apparently underpredicts viscous losses for the rotational boundary layer studied. Accordingly, future attention to turbulence modeling is important. (5) Viscous losses are reduced by shrouding as well. However, small axial clearances were observed computationally to increase viscous losses compared with larger shroud clearances, suggesting that this effect could become important for even smaller clearances and at higher speeds. (6) The CFD results show good agreement with the Glenn experiments. (7) The NASA Glenn data show quite similar loss trends to the idealized shrouded gear configurations studied computationally. (8) The CFD studies suggested a set of possible geometric tooth modifications to further reduce windage loss. One of these appears quite promising and the authors anticipated studying this in the NASA Glenn facility.

Our continuing work in the gear windage aerodynamics area focuses on (1) further validation for shrouded gears as more Glenn data become available, (2) validate for viscous-only losses with available data on rotating disks [4,23], (3) multiphase flows, (4) overset gridding in the context of gear tooth contact, and (5) further evolution of design guidance for minimizing windage losses.

## Acknowledgment

This work was supported by NASA under NASA Cooperative Agreement No. NNX07AB34A.

## Nomenclature

$\forall$ = volume
$\underline{r}$ = radial vector
$\underline{V}$ = velocity vector in the relative frame of reference
$A_\theta$ = tangential projection of the area of the grid face
$C_p$ = pressure coefficient
$D$ = wall damping coefficient
$d_n$ = measure of the normal distance to the nearest wall
$k$ = turbulence kinetic energy
$L_{\text{ref}}$ = reference length
$p$ = pressure
$p_\infty$ = freestream pressure
$q$ = $\sqrt{k}$
$r$ = radial coordinate of grid face centroid
$S$ = area
$T^*(x)$ = torque per unit width
$V^*$ = local normalized projected relative velocity magnitude
$V_x$ = velocity $x$-component
$V_y$ = velocity $y$-component
$V_{\text{ref}}$ = reference velocity
$C_\mu, C_1, C_2, \alpha$ = turbulence model coefficients
$\text{Pr}_q, \text{Pr}_\omega$ = turbulent Prandtl numbers
$\varepsilon$ = turbulence dissipation rate
$\mu$ = molecular viscosity
$\mu_t$ = eddy viscosity
$\omega$ = $\varepsilon / k$
$\rho$ = density
$\underline{\omega}$ = angular velocity
$\underline{\underline{\tau}}$ = shear stress tensor

## References

[1] Dawson, P. H., 1984, "Windage Loss in Larger High-Speed Gears," Proc. Inst. Mech. Eng., Part A, **198**(1), pp. 51–59.
[2] Dawson, P. H., 1988, "High Speed Gear Windage," General Electric Company (Great Britain) Review, **4**(3), pp. 164–167.
[3] Winfree, D. D., 2000, "Reducing Gear Windage Losses From High Speed Gears," ASME Power Transmission and Gearing Conference, ASME.
[4] Diab, Y., Ville, F., Velex, P., and Changenet, C., 2004, "Windage Losses in High Speed Gears—Preliminary Experimental and Theoretical Results," ASME J. Mech. Des., **126**(5), pp. 903–908.
[5] Handschuh, R. F., and Hurrel, M. J., 2010, "Initial Experiments of High-Speed Drive System Windage Losses," International Conference on Gears.
[6] Anderson, N. E., and Loewenthal, S. H., 1979, "Spur-Gear-System Efficiency at Part and Full Load," NASA Technical Report No. TP 1622 AVRADCOM TR 79-46.
[7] Anderson, N. E., and Loewenthal, S. H., 1981, "Effect of Geometry and Operating Conditions on Spur Gear System Power Loss," ASME J. Mech. Des., **103**(1), pp. 151–159.
[8] Anderson, N. E., and Loewenthal, S. H., 1984, "Efficiency of Nonstandard and High Contact Ratio Involute Spur Gears," Fourth International Power Transmission and Gearing Conference, ASME, Paper No. NASA TM 83725 US-AAVSCOM TR 84-C-9.
[9] Handschuh, R. F., and Kilmain, C. J., 2002, "Preliminary Investigation of the Thermal Behavior of High-Speed Helical Gear Trains," International Conference on Gears, International Federation for the Theory of Machines and Mechanisms, Paper No. NASA/TM-2002-211336 ARL-TR-2661.
[10] Heingartner, P., and Mba, D., 2005, "Determining Power Losses in the Helical Gear Mesh," Gear Technol., Sept./Oct., pp. 32–37.
[11] Al-Shibl, K., Simmons, K., and Eastwick, C. N., 2007, "Modelling Windage Power Loss From an Enclosed Spur Gear," Proc. Inst. Mech. Eng., Part A, **221**, pp. 331–341.
[12] Eastwick, C. N., and Johnson, G., 2008, "Gear Windage: A Review," ASME J. Mech. Des., **130**(3), p. 034001.
[13] Marchesse, Y., Changenet, C., Ville, F., and Velex, P., 2009, "Investigations on CFD Simulation for Predicting Windage Power Losses in Spur Gears," Third International Conference on Integrity, Reliability and Failure.
[14] Imai, H., Goi, T., Arisawa, H., and Nishmimura, M., 2009, "Reduction in Gear Windage and Churning Loss by Optimum Shroud Design With Aid of CFD," JSME International Conference on Motion and Power Transmission, JSME.
[15] Hill, M. J., Kunz, R. F., Noack, R. W., Long, L. N., Morris, P. J., and Handschuh, R. F., 2008, "Application and Validation of Unstructured Overset CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulation," 64th Annual Forum of the American Helicopter Society.
[16] Hill, M. J., 2010, "A Computational Investigation of Gear Windage," Ph.D. thesis, Pennsylvania State University.
[17] Coakley, T. J., 1983, Turbulence Modeling Methods for the Compressible Navier–Stokes Equations, Paper No. AIAA 83-1693.
[18] Chorin, A. J., 1967, "A Numerical Method for Solving Incompressible Viscous Flow Problems," J. Comput. Phys., **2**(12), pp. 12–26.
[19] Roe, P. L., 1981, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," J. Comput. Phys., **43**, pp. 357–372.
[20] van Leer, B., 1979, "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," J. Comput. Phys., **32**, pp. 101–136.
[21] Noack, R. W., 2005, "SUGGAR: A General Capability for Moving Body Overset Grid Assembly," 17th AIAA Computational Fluid Dynamics Conference, AIAA, Paper No. AIAA 2005-5117.
[22] Noack, R. W., 2006, "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," 17th AIAA Computational Fluid Dynamics Conference, AIAA, Paper No. AIAA 2005-5116.
[23] Daily, J. W., and Nece, R. E., 1960, "Chamber Dimension Effects on Induced Flow and Frictional Resistance of Enclosed Rotating Disks," ASME J. Basic Eng., **82**, pp. 648–653.

In the final stages of the NRA program, we turned our attention to further physics interpretation/understanding, and the role of multiphase flow. We discovered and proved that angular momentum confinement near the pitch-line-face region plays an important role in shroud effectiveness. This was a discovery not understood by the community prior to our research. As postulated by other researchers, we showed that tooth channel mass flow rate restriction due to the shrouds is also important in reducing windage loss, and is the dominant mechanism for very small clearance radial shrouds. It was demonstrated that even relatively small leakage paths in the shroud can reduce the shroud's effectiveness due to the loss of angular momentum confinement.

In the multiphase flow modeling area, a plausible set of droplet and film models, separately validated for aerodynamic application, returned significantly increased losses associated with oil lubrication. The multi-phase studies undertaken demonstrated a richness of physics that we have only begun to explore, including significantly larger predicted losses than suggested by homogeneous assumptions, important local droplet concentration effects, and the necessity of incorporating film modeling. The 2012 AHS Forum paper which summarizes this research is included here in its entirety.

# Computational Studies of the Roles of Shrouds and Multiphase Flow in High Speed Gear Windage Loss

Robert F. Kunz[1,2], Matthew J. Hill[1,2,3], Kerrie J. Schmehl[1], Sean M. McIntyre[1,2]
[1] Applied Research Laboratory, The Pennsylvania State University, University Park, PA
[2] Department of Aerospace Engineering, The Pennsylvania State University, University Park, PA
[3] Currently: Bell Helicopter, Fort Worth TX
rfk102@arl.psu.edu

## ABSTRACT

This work, builds on the authors' recent CFD research on high speed gearbox windage loss[1-4]. There, 3D steady and unsteady, single phase simulations of spur gears were presented. Those studies served to elucidate some of the physics of windage losses, validate the methods and models employed against available experimental data, and explore design alternatives to mitigate windage loss. With this validated capability in hand, we have recently studied computationally two critical design relevant components of high speed gearbox aerodynamics: 1) The role of shrouds, widely used in rotorcraft transmission systems for windage loss mitigation, and 2) the role of lubricant transport, i.e., the multiphase flow resident in real gearboxes. Research findings and attendant recommended design guidance are reported in this paper.

## NOTATION

| | |
|---|---|
| $\underline{A}_{wall}$ | area vector of wall face |
| $C_D$, $C_{TD}$ | drag and dispersion coefficients |
| $C_\mu$, $\kappa$, $U^+$, $U^*$, $y^+$, $E$ | turbulence model parameters and scales |
| $D$ | inter-field drag force kernel, droplet diameter |
| $E$ | total energy ($\varepsilon+\frac{1}{2}q^2$) |
| $H$ | stagnation enthalpy |
| $h^{film}$ | film thickness |
| $I$ | rhothalpy ($H-\Omega^2r^2/2$) |
| $k$ | turbulent kinetic energy |
| $\underline{M}$ | inter-field non-drag force kernel |
| $\dot{m}$ | mass flow rate |
| $p$ | pressure |
| $q$ | mean kinetic energy |
| $q''$ | heat flux vector |
| $Re$ | Reynolds number |
| $\underline{r}$ | coordinate vector |
| $\underline{S}$ | area vector |
| $Sc_t$ | turbulent Schmidt number |
| $\underline{V}$ | velocity in relative frame of reference |
| $We$ | Weber number |
| $\alpha$ | volume fraction |
| $\varepsilon$ | internal energy, turbulence dissipation rate |
| $\mu$ | molecular viscosity |
| $\rho$ | density |
| $\sigma$ | surface tension |
| $\Gamma$ | inter-field mass transfer rate |
| $\underline{\Omega}$ | system rotation rate |
| $\underline{\underline{\tau}}$ | viscous stress tensor |
| $\forall$ | control volume |

## INTRODUCTION

Gear windage losses refer to the power losses that are a result of aerodynamic forces (pressure and viscous) acting against the rotation of high speed gears. Windage losses are a source of significant heating and power consumption in rotorcraft and other gearbox systems where high pitch-line velocities are encountered (nominally >80 m/s).

In earlier publications by the present authors[1-4], we studied various aspects of windage losses, using CFD, in the context of single phase flow, and validated our method against open literature data[5], and data recently obtained in the Gearbox Windage Facility at NASA Glenn Research Center[6]. That work showed that the dominant physical mechanism contributing to windage losses in spur gears is the pressure field associated with diversion and impingement of the high speed relative flow on the leading tooth surface adjacent to the face. Shrouding the gear was observed to mitigate the magnitude of this mechanism but not its dominance in the loss budget. Also, this mechanism suggested some passive gear geometry treatments to further reduce windage loss, these were explored computationally and one concept has shown design promise[3]. This earlier work complements a number of CFD studies carried out by other researchers that have appeared in recent years[7,8,9,10], some of which are reviewed in Ref. 3.

In this paper we report on progress in two research tracks. Firstly, a computational exploration of the physical mechanisms associated with shrouding has been carried out and several conclusions are drawn related to "why shrouds work" and "when/why they do not". Secondly, we incorporate non-equilibrium two-phase flow analysis and show that droplet and film flow distributions play an important role in windage loss.

It was shown experimentally by Dawson[11,12] and Winfree[13] that geometric modifications of the near-gear flow path, such as shrouding, can significantly reduce both windage losses and lubricating oil consumption (80% and 40% reductions observed, respectively, in Ref. 13). Accordingly, shrouding is of significant interest in rotorcraft transmission design and is used in numerous systems today. Unfortunately, only a few empirical studies have been made

Presented at the American Helicopter Society 68th Annual Forum, Fort Worth, TX, May 1–3, 2012. 

available to date, and little physical understanding has been proposed. Indeed, Winfree[13] observed that shrouding a bevel gear reduced the ability of the gear to pump fluid from the gear axis and through its teeth, which suggests that the shroud acts to restrict the amount of fluid that the gear draws in and then expels through the tooth channel. Dawson[11,12] also attributed windage losses in spur gears to the drawing of air axially into the tooth channel and its ejection radially at the gear centerline, i.e. that the gear acts as a pump and that windage losses can be interpreted as the rate of work performed on the air in this pumping process. He supported this hypothesis with flow visualization, and an observed reduction in windage losses when the channel entry was blocked – as by an axial or radial shroud. These hypotheses are supported in part below, but the mass flow blockage mechanism itself is shown to be only part of the story, with angular momentum dynamics playing importantly as well.

Accordingly, there is a need for more experimental data and analysis aimed at improved understanding of the physical mechanisms involved in shrouding schemes for loss reduction. The Gearbox Windage Facility at NASA Glenn Research Center[6], has been established in part to produce such data, and we recently validated the CFD method used here against a number of shroud configurations studied in that facility[3]. Below we study several shroud configurations and formulate and support a hypothesis as to why they are effective. These studies also show some instances where they are not effective. Together these observations lead to design guidance proposals.

The important role of lubrication oil in windage losses is well known although not well understood. Most high speed experimental windage studies to date have been performed in air[5,6,13], though forthcoming results from the NASA Glenn Windage Facility will provide 2-phase validation data. The incorporation of the two-phase gearbox environment in predictive analysis is generally quite primitive; all of the correlations in the literature employing, at most, dimensionally consistent mixture density and viscosity parameters[8], obtained using estimated mass loading and droplet homogeneity assumptions. Imai et al.[10] are the only group to the authors' knowledge to perform multiphase CFD analysis for gearbox aerodynamics. They analyzed a two-gear spiral-bevel system using a homogeneous VOF method and porosity model for the two-phase flow and gear meshing, respectively. These authors demonstrated that the VOF method was able to capture details of the macro-physics of lubrication jet breakup and transport, but such a method is inherently incapable of capturing the details of droplet/mist formation and transport at the scales where they contribute to mass loading of the environment (1-300μm).

In our view, the minimum level of modeling required to capture two-phase environmental effects in high speed gearboxes are methods that account for non-equilibrium interfacial dynamics. That is, two-fluid Eulerian or Lagrangian methods can accommodate slip velocity between the droplets and carrier air through the modeled influence of (at a minimum) interfacial drag. The reason that a non-equilibrium approach is needed is that droplet distribution is fundamental to local mass loading which in turn defines local mixture density and viscosity, and thereby loss magnitudes directly. The large shears and centrifugal forces encountered in high speed gearboxes lead to droplet distributions that are highly non-uniform by virtue of the droplet size spectrum that arises (i.e. the majority of droplet mass would have to reside in droplets less that 1μm in diameter for a near homogeneous distribution to arise). Lagrangian methods can capture these physics, but are likely to be very computationally intensive for the gearbox problem, and inherently unable to handle the wall films that arise. Accordingly, we have chosen an Eulerian two-fluid approach, which can be instrumented to model all of the physical mechanisms present in these systems. Specifically, it enables the modeling of droplet dynamics (drag, dispersion), droplet wall impaction and film breakup (both inter-field mass transfer mechanisms), multiple droplet sizes, droplet breakup and coalescence, film shear, and potentially other mechanisms. Most of these are implemented below.

Below we study a shrouded spur gear configuration under multiphase flow conditions. Oil mass loading, droplet size, and number of droplet fields are parameterized. Also, a scheme to properly accommodate wall films is introduced. A number of conclusions are drawn from these studies related to mass loading specification, droplet distributions, the importance of modeling films and boundary conditions. In particular it is shown that simple mixture density and viscosity based loss correlations are unlikely to return reliable quantitative results.

This paper is organized as follows. First, the governing equations, models and numerical procedures are reported. A series of shrouding related numerical studies are then performed and discussed. This is followed by a series of multiphase flow analyses, which demonstrate some of the richness of the multiphase physics that arise in these systems, suggest shortcomings likely to arise in simpler analysis, and point to further modeling refinements to be considered.

## THEORETICAL FORMULATION

### Governing Equations

A two-fluid Eulerian formulation is adopted. The conservation of mass, momentum and energy for each constituent present are written in integral conservation law form for a compressible flow in a steadily rotating coordinate system, in Equations 1-3. $k$, $l$ are constituent indicators, here $k$ = 0, 1, 2 for air, droplets, and films respectively. For all simulations presented in this paper, the carrier air field is assumed a perfect gas; the disperse droplet and continuous film oil fields are treated as incompressible.

$$\int_{\underline{S}} \rho^k \alpha^k \underline{V}^k \bullet d\underline{S} = \sum_{k \neq l} \left( \Gamma^{lk} - \Gamma^{kl} \right) \qquad (1)$$

$$\int_{\underline{S}} \rho^k \alpha^k \underline{V}^k (\underline{V}^k \bullet d\underline{S}) = -\alpha^k \int_{\underline{S}} p d\underline{S} + \int_{\underline{S}} \alpha^k \underline{\underline{\tau}}^k \bullet d\underline{S} =$$
$$-\int_{\forall} \rho^k \alpha^k [\underline{\Omega} \times (\underline{\Omega} \times \underline{r})] - \int_{\forall} 2\rho^k \alpha^k (\underline{\Omega} \times \underline{V}) \qquad (2)$$
$$+\sum_{k \neq l} \underline{M}^{lk} + \sum_{k \neq l} \left[ D^{kl} \left( \underline{V}^l - \underline{V}^k \right) + \Gamma^{lk} \underline{V}^l - \Gamma^{kl} \underline{V}^k \right]$$

$$\int_{\underline{S}} \rho^k \alpha^k I^k (\underline{V}^k \bullet d\underline{S}) =$$
$$\frac{\partial}{\partial t} \int_{\forall} \alpha^k p d\forall + \int_{\underline{S}} \left( \alpha^k \underline{\underline{\tau}}^k \bullet \underline{V}^k \right) \bullet d\underline{S} + q''^k + \sum_{k \neq l} \left( \Gamma^{lk} E^l - \Gamma^{kl} E^k \right) \qquad (3)$$

**Turbulence and Multiphase Modeling**

A high-Reynolds number k-ε turbulence model with wall functions is used in the studies that follow, for the air field only. Although sublayer resolved turbulence modeling can improve the accuracy of the viscous component of loss in gear windage analysis[1,3], its underprediction by high Reynolds number formulations are of secondary importance in the shroud and multiphase flow studies undertaken here. Multiphase turbulence effects are discussed below.

The terms $D^{kl}$, $I^{kl}$ and $\underline{M}^{kl}$ that appear in the continuity and momentum equations represent interfacial drag forces, interfacial mass transfer and interfacial "non-drag" forces, respectively. A broader discussion of the relevance of these terms in general two-fluid simulations, as well as details of their numerical implementation is available in Ref. 14. Here we present the models used for the gearbox windage application.

Droplet drag is modeled in a conventional manner[15]:

$$D^{air-droplet} = \frac{3}{4} \frac{\rho^{air} C_D}{D^{droplet}} \left| \underline{V}^{rel} \right|,$$
$$C_D = 0.4 + \frac{24}{\mathrm{Re}^{droplet}} + \frac{6}{1 + \sqrt{\mathrm{Re}^{droplet}}}, \qquad (4)$$
$$\mathrm{Re}^{droplet} \equiv \frac{\rho^{air} \left| \underline{V}^{rel} \right| D^{droplet}}{\mu^{air}}$$

To accommodate the effect of the carrier field (air) turbulence, a droplet dispersion model due to Lopez de Bertodano[16] is used:

$$\underline{M}^{air-droplet} = C_{TD} \rho^{air} k^{air} \nabla \alpha^{droplet} \qquad (5)$$

The drag and dispersion models in Equations 4 and 5 represent a "minimum" set of interfacial force models in that for low mass loading, they are sufficient to account for the distribution of droplets within the gearbox. We do not model other interfacial dynamics in this work (e.g., lift, virtual mass) as they are likely to be negligible in this high density ratio system ($\rho^{air}/\rho^{droplet} << 1$).

There is no data yet available with which to calibrate droplet size distribution in the gearbox system. In this first multiphase flow study, we account for droplet size by parameterizing its effect on the solution. Droplet coalescence and breakup are not yet modeled.

In the high speed gearbox system, a significant (but unknown) amount of the oil in the gearbox is constituted as films on the gears, shrouds, housing and other components. Although 2-field analysis (air+droplets) can provide some interesting physics and design guidance for relevant mass loadings when shrouds are present, droplets will accumulate locally in a non-physical manner if not allowed to deposit as films on the walls since their interfacial area density and thereby drag is so high. Modeling the mass transfer from droplets to films enables films to be modeled explicitly with relevant (small) interfacial areas densities, and appropriate drag kernels. For droplet deposition to films we incorporate a hybrid impaction and diffusion mass transfer model due to Haworth et al.[17]:

$$\Gamma^{droplet-film} = \Gamma^{droplet-film}_{\text{impaction}} + \Gamma^{droplet-film}_{\text{diffusion}}$$
$$\Gamma^{droplet-film}_{\text{impaction}} = \rho^{droplet} \left( \underline{V}^{droplet} \bullet \underline{A}_{wall} \right) \alpha^{droplet}$$
$$\Gamma^{droplet-film}_{\text{diffusion}} = \frac{\rho^{air} U^{*,air} \left| \underline{A}_{wall} \right|}{Sc_t \bullet U^{+,air}}, \qquad (6)$$
$$U^{*,air} \equiv C_\mu^{3/4} \sqrt{k^{air}}, U^{+,air} \equiv \frac{1}{\kappa} \ln\left( Ey^+ \right)$$

For drag exerted on the (assumed thin) films by the air we have:

$$D^{air-film} = \frac{\mu}{h_{film}} \left| \underline{A}_{wall} \right| / \forall, h_{film} \equiv \frac{\alpha^{film} \forall}{2 \left| \underline{A}_{wall} \right|} \qquad (7)$$

with attendant wall drag on the film:

$$\left| \underline{\underline{\tau}} \right|^{film}_{wall} = \frac{\mu \left( \left| \underline{V} \right|^{film} - \left| \underline{V} \right|^{wall} \right)}{h_{film} / 2} \qquad (8)$$

Lastly, we model film breakup mass transfer to droplets (in order that a physically representative droplet-film equilibrium is attained) following Schadel and Hanratty[18]:

$$\Gamma^{film-droplet} = .001 * \left[ .4 \ln(150 \bullet \mathrm{I}_R We + 1) + 1.4 \sqrt{\mathrm{I}_R We} \right]$$
$$* U^{*,film} \sqrt{\rho^{air} \rho^{film}} \qquad (9)$$

where:

$\Gamma_R \equiv$ roll wave intermittency $\equiv .15 + .75\Gamma_e$

$\Gamma_e \equiv$ excess film flow rate $\equiv \left[\!\left[ 0, \left( \dot{m}_{film,out} - \dot{m}_{film,in} \right)_{cell} \right]\!\right]$

$$We = \frac{\rho^{air} \left| U^{rel} \right|^2 h_{film}}{\sigma}, h_{film} \equiv \frac{\alpha^{film} \forall}{2 \left| \underline{A}_{wall} \right|}$$

Equation 9 is applied when the local Weber number exceeds the critical breakup Weber number of 17.

### CFD Numerics and Code

The CFD code used in this work, NPHASE-PSU[14] is a parallel face-based, cell-centered, arbitrary-element unstructured multiphase flow solver. The baseline algorithm follows established segregated pressure based methodology. Compressibility is accommodated by including a density correction in the formulation of the pressure corrector equation. A number of algorithmic details associated with multiphase flow are relevant here, and those details are available in Ref. 14. The code is instrumented with overset meshing and mesh motion, both used in our earlier windage publications, but all of the simulations carried out here use contiguous meshes which are stationary in the rotating frame of reference. A single tooth passage is simulated and periodic boundary conditions are applied. Since these are spur gears the symmetry of the system was also exploited and only ½ of the domain was modeled. A 517,348 cell mesh was used for the unshrouded analyses presented below, fewer cells are required for the various shrouded cases. The meshes conform to wall spacing requirements for the turbulence model wall functions used ($y^+ \leq 200$). These grids are consistent with those used in our earlier work[1,3].

All of the studies presented in this paper are applied to the same spur gear geometry, a 72-tooth spur gear studied experimentally by Diab et al.[5], which has a pitch diameter of 288mm ($r_{tip}$=1.48m), a tooth width of 30 mm and a module of 4 mm.

## SHROUDING STUDIES

It has been known for some time that axial and radial shrouds can significantly reduce gearbox windage losses[11,12,13]. In Refs. 2-4 it was shown that the favorable impact that shrouding has on windage losses can be predicted quantitatively, including the trends associated with the axial and radial clearance of the shrouding. However, the physics behind why shrouding schemes work has remained only partially understood.

It has been conjectured in the literature[11,12,13] that shrouds restrict the "pumping" ability of the gear teeth, and that this is fundamentally responsible for the reduction in windage loss. To explore this, we study unshrouded and shrouded spur gears computationally. Figure 1 shows a visualization of the unshrouded simulation of the test spur gear rotating at 1000 $s^{-1}$ in the direction indicated by the arrow. There, relative frame streamlines show the drawing in of the near face flow and radial ejection of this flow at the tooth channel centerline, consistent with Dawson's observations[11,12]. A computational slice is taken across the gear top land and contours of radial velocity are plotted with the ejecting relative flow streamlines. When integrated, this net radial mass flow rate provides a quantitative figure for the gear "pumping".

The four shrouds studied are designated large-radial-large-axial (LRLA), large-radial-small-axial (LRSA), small-radial-large-axial (SRLA), and small-radial-small-axial (SRSA). These are shown schematically in Figure 2. The small and large radial shrouds extend .0044*$r_{tip}$ and .1733*$r_{tip}$ radially outward from the tip. The small and large axial shrouds extend .0044*$r_{tip}$ and .0970*$r_{tip}$ axially away from the gear face.

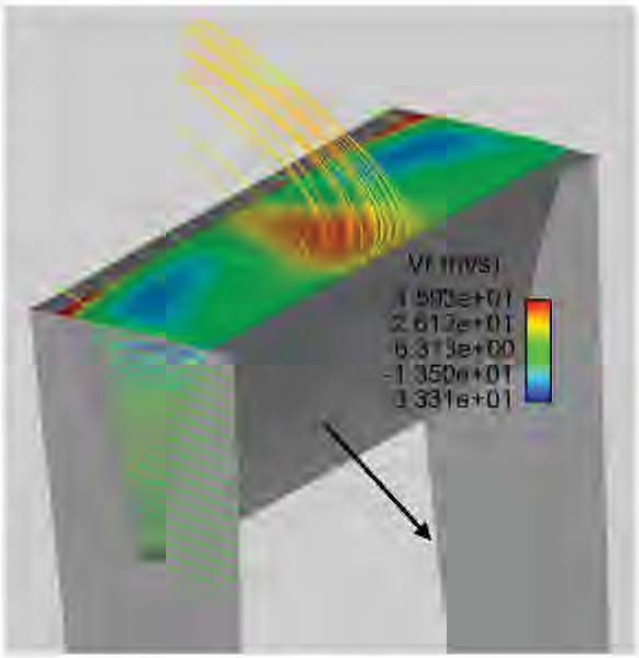



**Figure 1. Relative streamlines and radial velocity contours for an unshrouded simulation of a spur gear.**

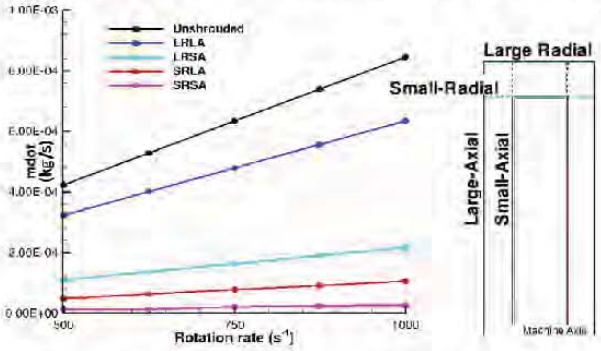



**Figure 2. Predictions of net mass flow ejected radially from gear tooth channel for unshrouded and four shrouded configurations.**

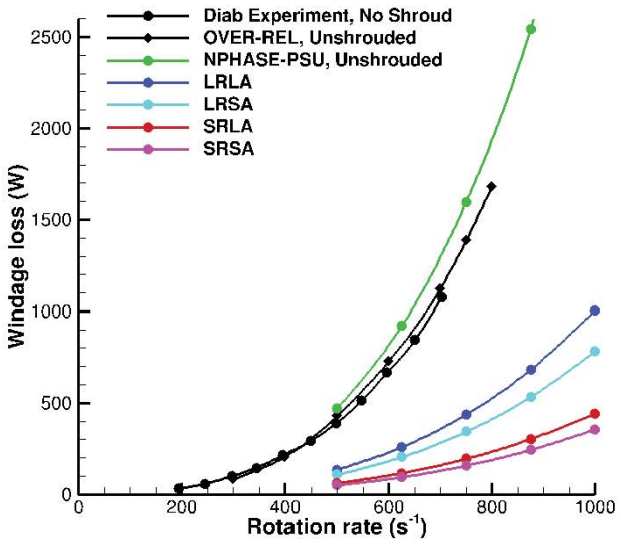



**Figure 3. Windage loss predictions for shrouded and unshrouded spur gears.**

The radial mass flow integration was carried out for four shroud configurations and the results plotted in Figure 2. Figure 3 shows predicted windage loss vs. rotation rate for the unshrouded case and the four shrouded cases. The unshrouded simulation compares well with the experimental measurements due to Diab et al.[5], and our previously published results using another CFD code[3]. (The present results somewhat overpredict the loss compared to experiment and the earlier analyses – the reasons for this have not been studied, but these differences do not impact the conclusions drawn from the comparative studies undertaken.)

Comparing Figures 2 and 3, it is clear that the integrated mass flow rates do not exhibit the same relative trends as the loss. Specifically, considering the LRLA shroud, the reduction in mass flow from the unshrouded gear, ≅25%, is far less than the observed ≅72-74% reduction in windage loss. For the SRSA shroud, the reduction in mass flow from the unshrouded gear, ≅97%, is far more than the observed ≅90% reduction in windage loss. In Figure 4, the ratio of windage loss to integrated mass flow is plotted for all cases. From this figure we see that each configuration exhibits similar trends with gear speed, but that the different shroud schemes exhibit large variations in this ratio. Indeed, the LRLA and SRSA shrouds show a consistent order of magnitude difference in this parameter. So clearly, mass flow restriction is only part of the physics which constitute performance gains with shrouds.

In order to better understand the nature of the shrouding's effect on losses, consider Figure 5a which shows contours of absolute tangential momentum for an unshrouded and the LRLA shrouded analyses of the test gear at a rotation rate of 1000 $s^{-1}$. The contour slice is taken at mid channel. In all gear systems the gear acts to impart significant angular momentum to the fluid. It can be seen that the shrouded gear confines the region of large absolute angular momentum near the teeth whereas this momentum is transported away from the teeth in the unshrouded case.

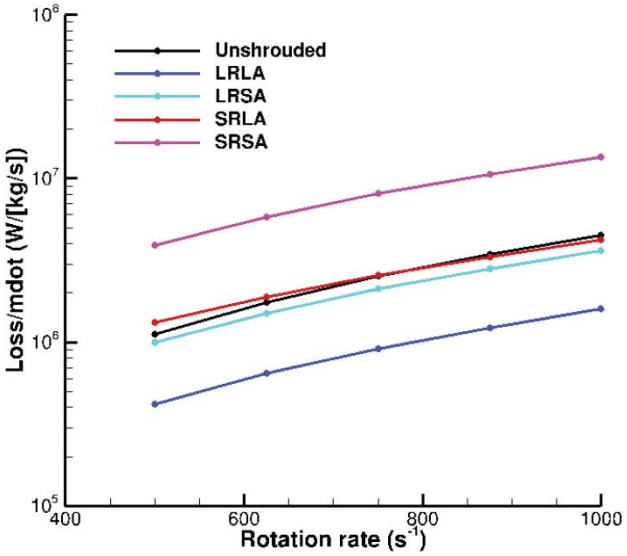



**Figure 4. Predictions of loss/mass flow ejected radially from gear tooth channel, for unshrouded and four shrouded configurations.**

Figure 5b shows a plot of angular momentum along an axial line through the domain at the pitch radius (indicated in Figure 5a) for the unshrouded and all of the shrouded cases. Considering the LRLA case, it is observed that the difference in angular velocity between the teeth and the near-gear-face flow is much lower for the shrouded case due to the confinement of the angular momentum. This leads to a much lower deceleration of the flow that impinges on the leading tooth near the face, in turn leading to lower pressure in the stagnation region where the flow is drawn in, as observed in Figure 5c. So for the LRLA, we have a small effect on loss reduction due to mass flow restriction, and a larger effect due to angular momentum confinement; together these give rise to a net loss reduction across the speed range studied of between 72% and 74%. This budget of mass flow and momentum effects is reflected in Figure 4 where the LRLA data is a factor of approximately 3 lower than the unshrouded budget and a full order of magnitude less than the SRSA budget.

Consider next the SRSA case. Figure 5b shows that the angular velocity between the teeth and the near-gear-face flow is again higher than for the unshrouded case but is lower that the large-axial cases by virtue of the proximity of the axial shroud. So we expect a reduced benefit due to angular momentum confinement for the small-axial shrouds compared to the large axial shrouds. Also, it was observed in Refs. 2, 3, 4 that increases in viscous loss begin to manifest for small axial shrouds. So we have the situation where mass flow rate is dramatically reduced by some 97% for all speeds, but the losses due to reduced angular momentum and viscous effects lead to higher relative losses than the reduced

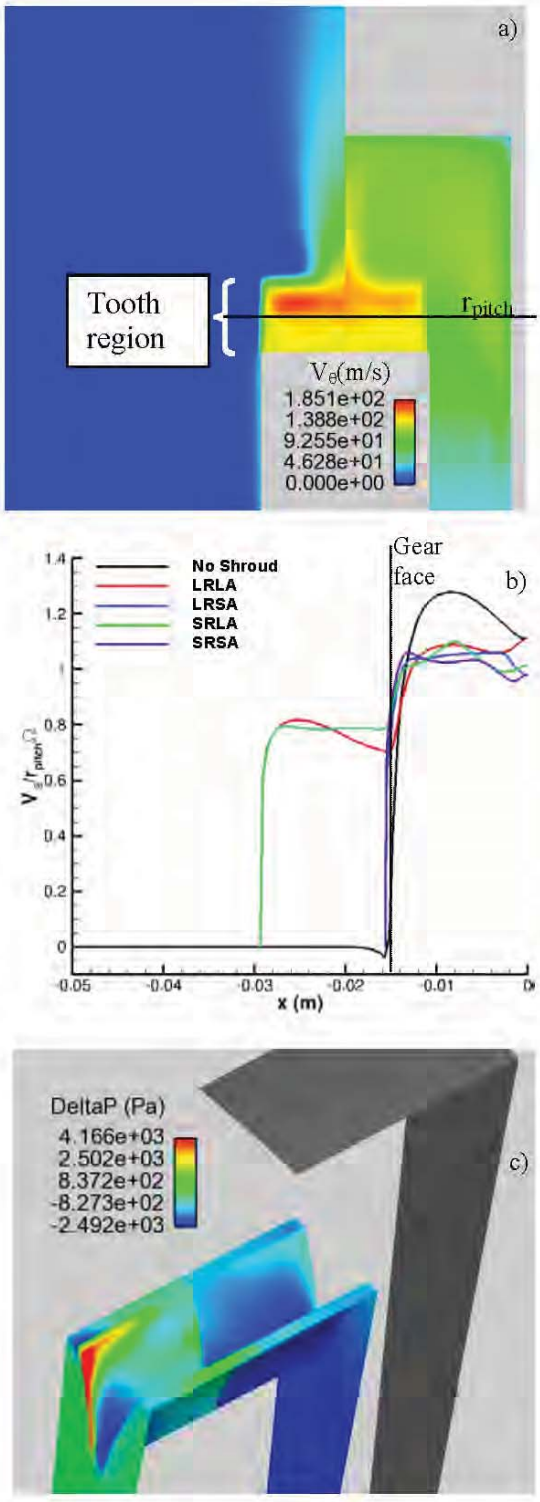



**Figure 5. a) Contours of absolute angular velocity for unshrouded and shrouded (LRLA) simulations of test gear at 1000 $s^{-1}$, b) Angular velocity along pitch line for all configurations at 1000 $s^{-1}$, c) Tooth pressure contours for unshrouded and shrouded (LRLA) simulations at 1000 $s^{-1}$.**

mass flow rate alone would suggest. This budget of mass flow and momentum effects is reflected in Figure 4 where the SRSA data is a factor of approximately 3 higher than the unshrouded budget and a full order of magnitude higher than the LRLA budget.

Further consideration of Figures 4 and 5b lead to a final observation. The LRSA and SRSA shrouds exhibit similar near face angular momentum profiles as do the LRLA and SRLA shrouds. However, the SRSA shroud shows predicted windage losses 53-55% lower than the LRSA shroud across the speed range, and the SRLA shroud shows predicted windage losses 55-56% lower than the LRLA shroud across the speed range. This tracks well with the mass flow rate reductions: SRSA shows predicted mass flow rate reductions 87- 88% lower than the LRSA, SRLA shows predicted mass flow rate reductions 83- 85% lower than the LRLA. So for a given axial shroud location we see that radial shrouding's dominant effect is indeed on mass flow reduction.

The roles of angular momentum confinement and mass flow restriction through the channel can provide significant guidance in the design of gear+shroud systems. In Refs. 2, 3, 4, several gear tooth modifications were explored in order to reduce the mass flow into the channel, a spoiler concept shows promise there. Analogously, CFD can be used to optimize shroud geometry towards reducing channel mass flow and/or to maximizing angular momentum confinement. Considering the latter, in real gearbox systems, holes are required in the shrouding for plumbing, drainage, meshing and hub penetration. Such penetrations can reduce the effectiveness of shrouds by allowing angular momentum to "leak" away from the near-pitchline-face region. To explore this, Figure 6 shows elements of a drainage slot parameterization study. There, the results of three simulations are presented, with the nominal, fully enclosed SRLA shroud explored above, and two slotted-shroud configurations, each rotating at 750 $s^{-1}$.

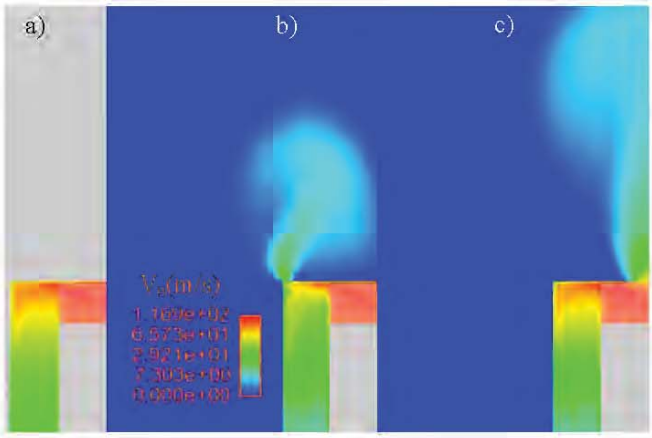



**Figure 6. a) Contours of absolute angular velocity for SRLA configuration at 750 $s^{-1}$. a) Fully shrouded, b) Radial shroud corner slots, c) Radial shroud centerline slot**

Slots were introduced at the radial shroud corners and centerline as shown in the figure, and the computational domain was extended radially and axially into a volume so that the flow can exhaust. Figure 6 shows predicted contours of angular momentum. Both slotted geometries allow some transport of angular momentum into the external volume. As seen in the contours, this comes at the expense of angular momentum in the near-pitchline-face region, for both slotted cases, when compared to the fully enclosed run. Also, the ejected channel mass flow rate increased by 13% for the corner slot, and decreased by 10% for the center slot (perhaps counter-intuitively). Figure 7 shows the loss values associated with these simulations. For these notional slots, windage loss is approximately doubled. This significant loss increase associated with comparatively small shroud penetrations is consistent with data recently published by Handschuh and Hurrell[6]. These simple parametric slot studies indicate that unavoidable penetrations in gear shrouds can significantly hurt the windage loss performance of these systems, and therefore CFD optimization of the size and placement of these penetrations can benefit gearbox performance. Of course, due to the non-periodic nature of real penetrations, such optimization studies will necessarily be much more computationally intensive than the exploratory studies presented here.

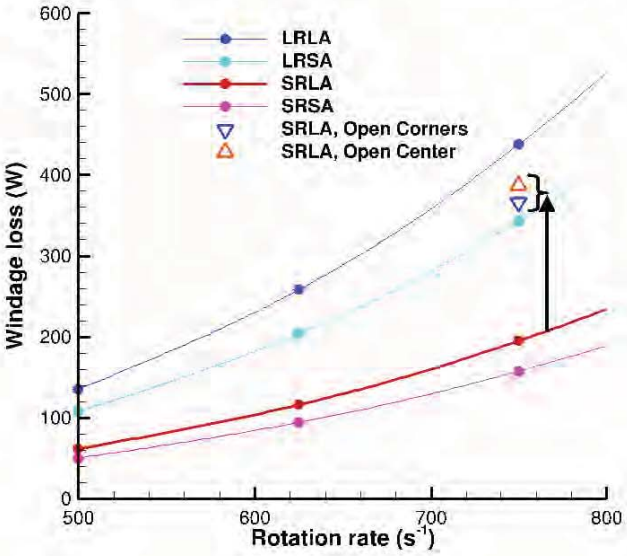



**Figure 7. Windage loss predictions for shrouded gears. Loss values for SRLA slotted configurations are plotted at 750 $s^{-1}$.**

## MULTIPHASE FLOW STUDIES

Windage loss in lubricated systems is intuitively dependent on droplet concentrations that are nominally local to the gear pitchline. Therefore CFD methods that model droplet concentration and size distributions accurately will be required in order that quantitatively accurate loss predictions are returned. This is a significant challenge for a number of reasons including: 1) There is no experimental data yet available with which to calibrate/validate droplet concentration or size distributions in the gearbox system, 2) Accurate and validated models for droplet drag, dispersion, and deposition, film drag and breakup are required – a first set of these models is incorporated here. But also, in order that local droplet *size distributions* be predicted accurately, models for droplet coalescence and breakup, and droplet bouncing and splash will need to be incorporated, as likely will more sophisticated models for the baseline set of interfacial transfer models used here (Equations 4-9), 3) The boundary conditions for these multiphase simulations are difficult to define unless the complete gearbox with all lube injection plumbing, shroud features and gear meshing are modeled. Indeed, as discussed below, even estimating the total oil mass loading for the gearbox is challenging.

In these first multiphase flow studies, we account for droplet size by parameterizing its effect on the solution, and by performing multiple-droplet-field analysis. The droplet models presented in Equations 4-6 (drag, dispersion, deposition) have been partially validated against a high speed aerodynamic data set[19]. Figure 8 shows predicted non-dimensional deposition rate vs. distance along the airfoil surface for an MS-317 airfoil at 0° angle-of-attack, free stream velocity of 90 m/s, and a free stream median volumetric diameter of 92 μm. In the experiment, 10 droplet size "bins" were used to characterize the far-field conditions – free stream liquid water content and droplet diameter were reported for all ten. Accordingly, a 12 field simulation was run (air + film + 10 droplet fields). Figure 8 shows that the NPHASE-PSU results match closely with the models in LEWICE-3D[20], for the case where a droplet splashing model was not-included. LEWICE does a much better job of matching the data when a splashing model is included, suggesting we include splashing in our future work.

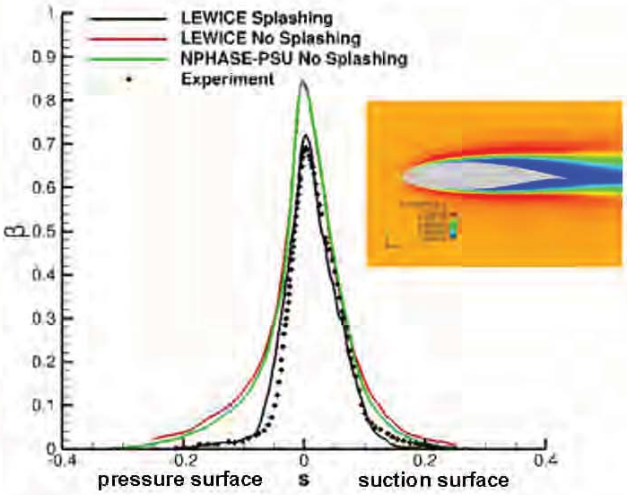



**Figure 8. Non-dimensional deposition rate vs. distance along airfoil surface for an MS-317 airfoil. Contours of predicted total droplet volume fraction are also shown. Data and LEWICE predictions from Refs. 19, 20.**

The single phase simulations presented above included no inflow or outflow boundaries. This is accommodated numerically by fixing the static pressure near atmospheric at a single cell in the computational domain, an approach that is guaranteed to conserve mass. In the multiphase analyses, this cannot be done since there is no mechanism to replenish mass loss errors in a given field during the course of iteration. Accordingly we introduce notional inflow and outflow boundary conditions to the test gear configuration, and abandon the model symmetry. Specifically a small (compared to the pitchline velocity) inlet flow of air and droplets is specified and an outflow boundary is also specified, as shown in Figure 9.

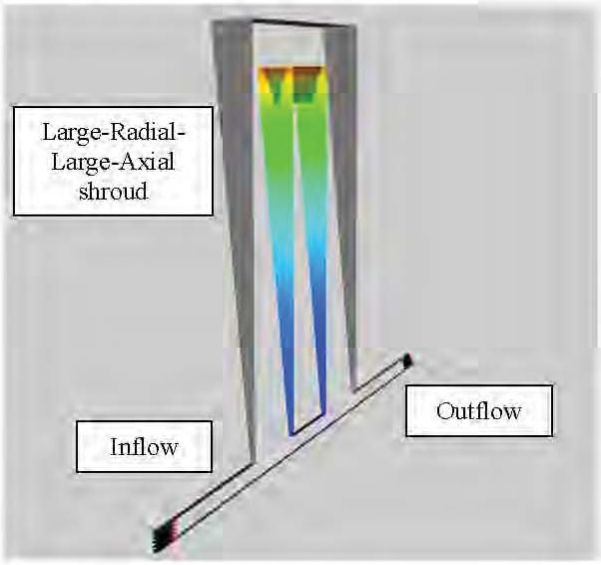



**Figure 9. Computational domain for multiphase analyses.**

The first set of calculations performed employ one droplet field and no film field. Per Equation 1, we adopt a volume fraction form for constituent transport. The air and droplet volume fractions are specified at the inlet in addition to their velocities, and this defines the oil injection mass flow rate. However, although the injection flow rate influences the oil mass loading (≡ [total mass of oil]/[total mass of oil+air]) in the gearbox (or shrouded gear system), *it does not define it.* Specifically, the mass of oil in the computational domain depends on many factors, but is not uniquely defined by inflow and outflow mass flow rate since the steady state oil continuity equation requires:

$$\int_{\underline{A}} \rho^{oil} \alpha^{oil} \underline{u}^{oil} \bullet d\underline{A} = 0, \qquad (10)$$

but says nothing about the distribution of oil in the domain. This has two implications: 1) Measuring lubrication flow rates experimentally is not sufficient for validation – we will need to know local mass fractions inside the enclosure, 2) Computationally we cannot "specify" mass loading, it is an output of the computation. So although increasing inflow flux of oil to the rig (or computational domain) will increase the equilibrium oil mass loading in the system, the relationship will not be linear.

For the 2-field droplet studies we ran a range of fixed droplet sizes, inlet oil flow rates, and gear speeds, all using the test gear with radial shroud extending .101*$r_{tip}$ radially outward from the tip and axial shroud extending .312*$r_{tip}$ axially away from the gear face. Figure 10 shows the results of a cross section of these runs. Several observations are relevant: 1) All multiphase analyses exhibit a noticeable increase in windage loss over the single phase case, 2) Increased mass loading for a given droplet size increases loss, 3) Increasing droplet size for a given mass loading increases loss.

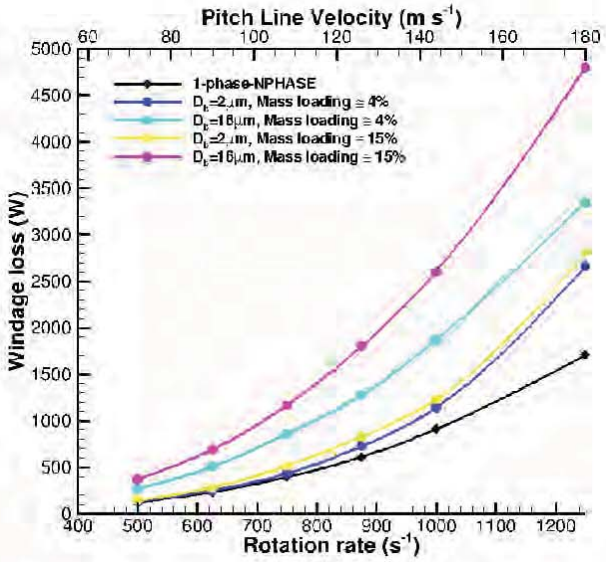



**Figure 10. Predicted windage losses for single phase and two-field air-droplet simulations.**

Here we interrogate the details of the simulations in order to better understand the physics underlying these windage loss trends, and to ascertain how general these results might be. In Figure 11a, contours of droplet volume fraction are plotted for a 2μm droplet field at a rotation rate of 750 $s^{-1}$. The contour slice is taken at mid channel and the tooth region is displayed as partially transparent. Figure 11b shows the same plot for the 16μm droplet field.

As expected, the 2μm droplet distribution is more uniform than the 16 μm case. Nevertheless, even for the smaller droplets, with their attendant high drag coefficient, the high shear rates, centrifugal forces and secondary air flows in this system give rise to a significant accumulation of droplets near the shroud and "starvation" in the tooth channel region. This effect is much more pronounced for the 16 μm droplets which exhibit almost no droplets in the tooth channel region and even more accumulation of droplets near the shroud corners. In both cases a slight asymmetry associated with the non-symmetric boundary conditions is

noticeable. The predictions in Figure 10, 11a, and 11b suggest that a simple mixture density scaling will not be suitable for loss correlations. Droplet diameter should be incorporated in some fashion. Figures 12a and b isolate the low and high mass loading conditions respectively. Included are the single phase results for this shroud and the single phase result scaled by the bulk mixture density based on the mass-loading. Clearly in both cases the predicted losses are much higher than would be returned by simply scaling a 1-phase analysis by the bulk mixture density or, equivalently, by running a homogeneous mixture model. As expected, the 2μm results are closer to the scaled 1-phase loss than the 16μm results, but they are still significantly higher, especially at high speeds.

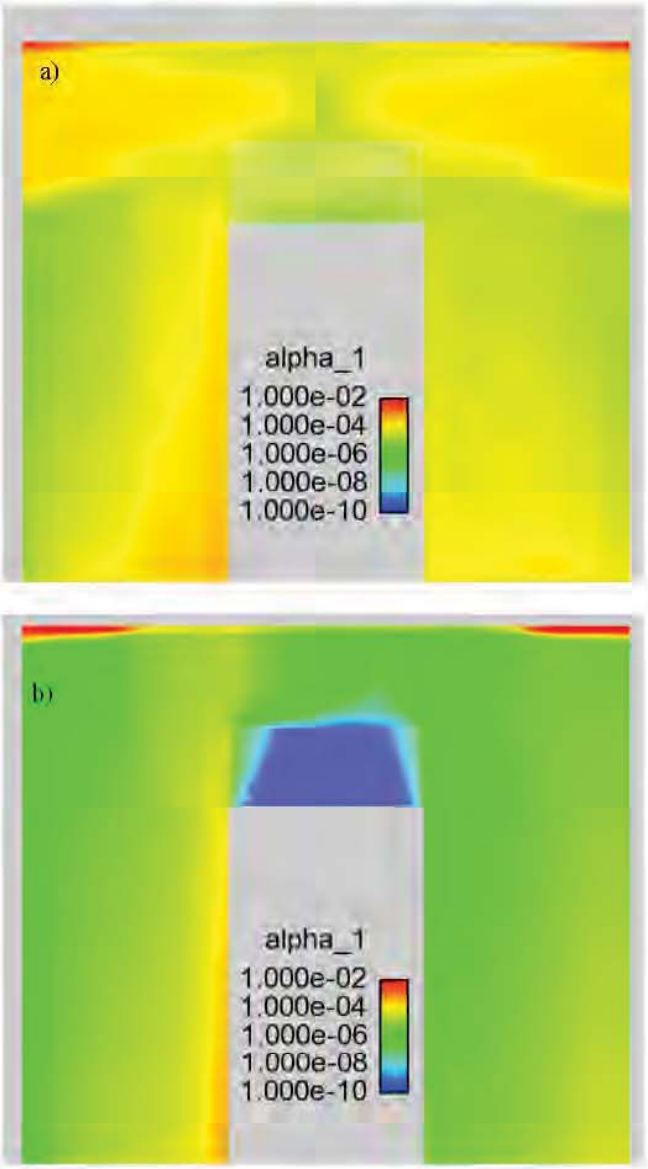



**Figure 11. Predicted droplet volume fraction distribution a) 2μm, b) 16 μm simulations at 750 $s^{-1}$.**

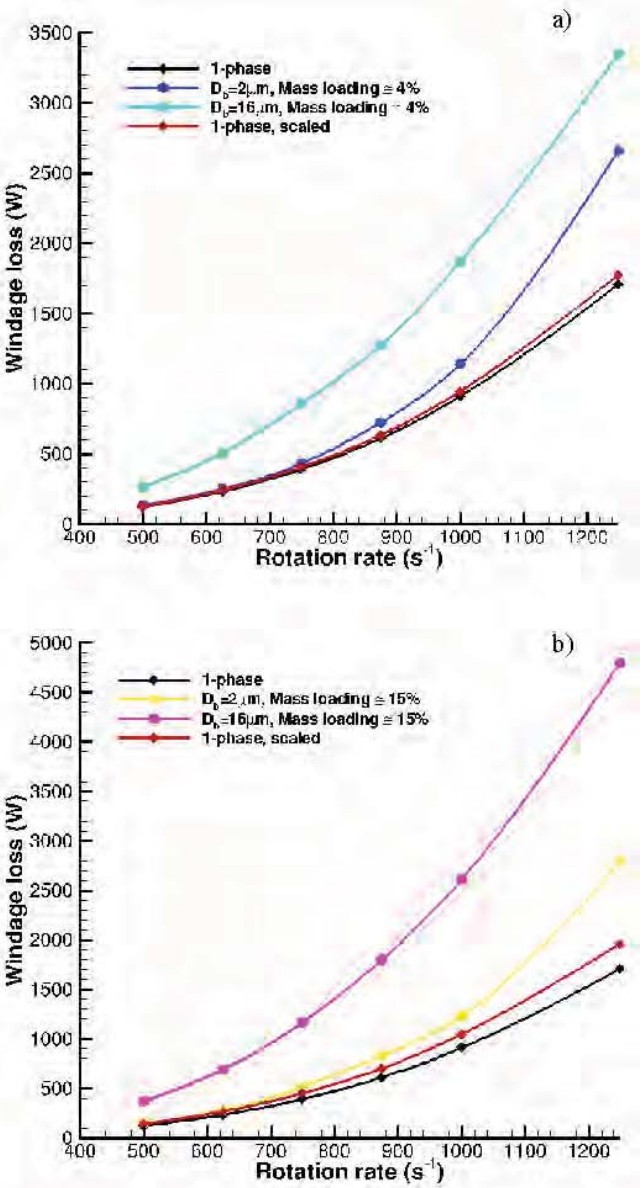



**Figure 12. Predicted windage losses for single phase and two-field air-droplet simulations. Mixture density scaled 1-phase results included. a) Low mass loading, b) High mass loading.**

Apparently, the non-equilibrium interfacial dynamics have led to non-uniformities in the mixture that increase windage losses. In order to better understand the physical reasons for this, consider Figure 13, where the higher mass loading, 750 $s^{-1}$, 2μm and 16μm simulations are compared. Contours of mixture density ($\rho_m \equiv \rho_{air}\alpha_{air} + \rho_{oil}\alpha_{oil}$) are plotted on the gear symmetry plane. In the 2μm simulation, droplets accumulate near the advancing face, not only at the face edge, but near the tip all along the channel, leading to higher local mixture densities and pressures there, and thereby higher net Δp across the channel. The net predicted windage loss for this case was 515 W vs. 392 W for single phase. The 16μm case shows significant mixture density increase

all along the leading face, and a larger pressure magnitude than the single phase case, albeit qualitatively localized like the single phase case. The net predicted windage loss for this case was 1165 W.

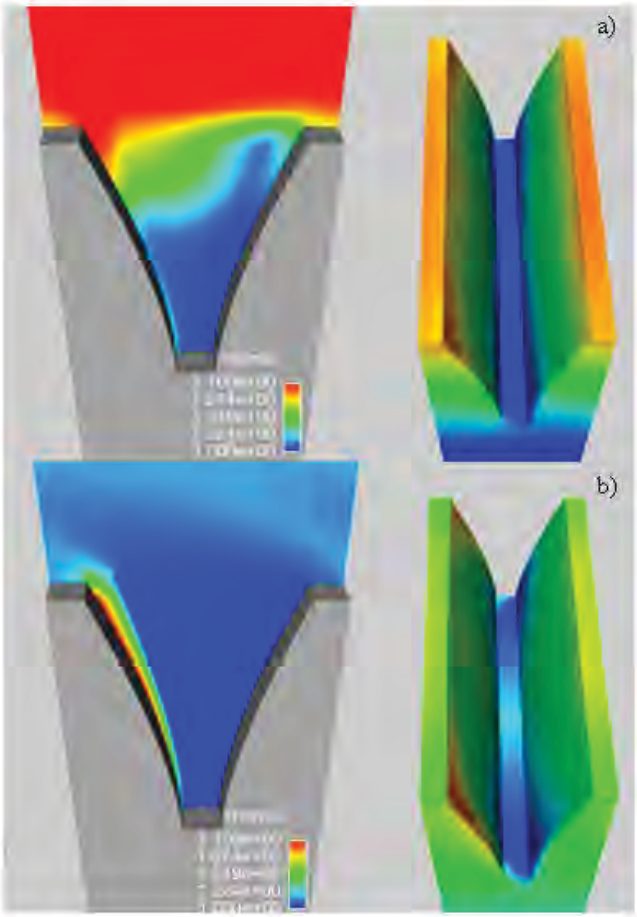



**Figure 13. Elements of higher mass loading, 750 $s^{-1}$ simulations. a) 2μm case. Contours of mixture density at gear centerline (left) and tooth static pressure contours (right). a) 16μm case. Contours of mixture density at gear centerline (left) and tooth static pressure contours (right).**

As discussed above and evidenced in Figure 11, droplets will accumulate locally in a non-physical manner if not allowed to deposit as films on the walls since their interfacial area density and thereby drag is so high. Accordingly we have implemented a film field which will exchange mass with the droplet field (droplet deposition, film breakup [Equations 6 and 9]) and has an associated air-film drag law (Equation 7) which accommodates the much lower interfacial areas density of the film and thereby allows it to flow along walls. The CFD code is instrumented with logic to allow films to be resident only in wall adjacent cells. We present preliminary results in Figure 14, showing the film volume fraction for a 3-field, 2μm, 750 $s^{-1}$, low mass loading run. Films develop along the walls, with maximum volume fraction (i.e., thickness) on the leading face and shroud corner, and mimima near the channel centerline. Droplet volume fraction contours for the same case shows that the droplets exhibit maximum concentrations near the shroud corners, but do not accumulate locally in a non-physical manner (compare Figure 11), since they are locally deposited as film which is transported, thickens and exchanges mass again with the droplet field along the axial shroud. These admittedly preliminary results exhibit the desired/required droplet film behavior as we move forward with model refinement and validation.

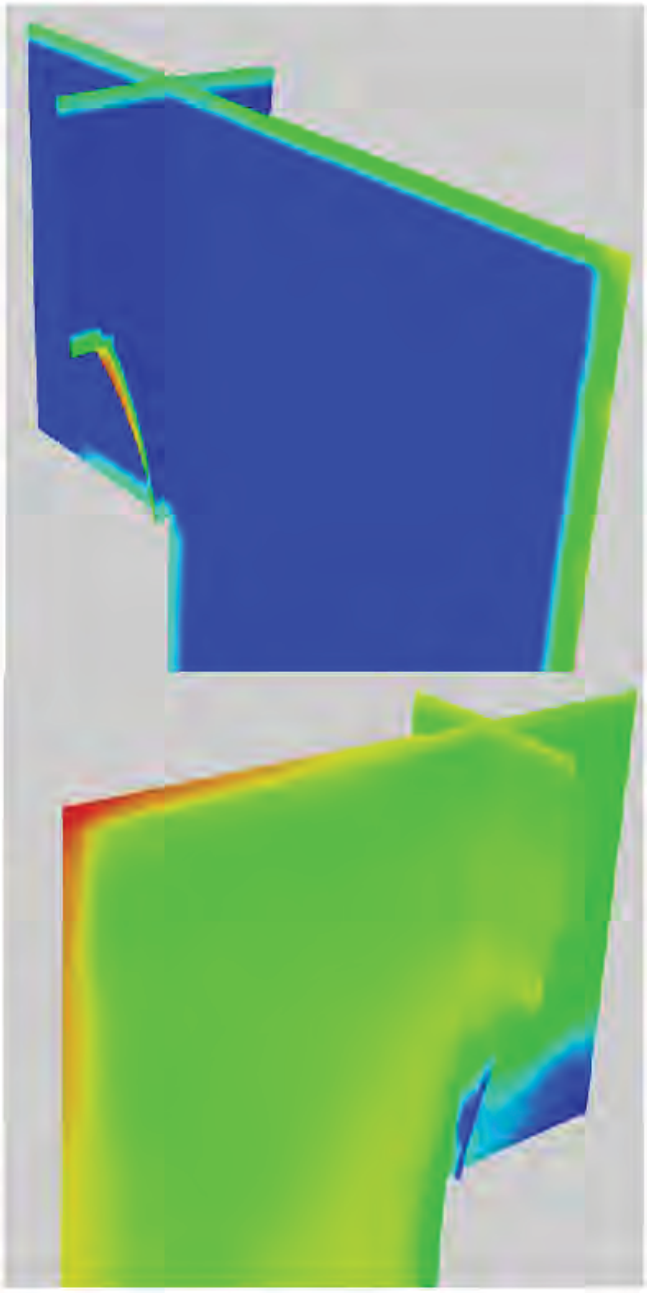


**Figure 14. Elements of 3-field, 2μm droplet, lower mass loading, 750 $s^{-1}$ simulation. Contours of a) Film volume fraction, b) Droplet volume fraction.**

## CONCLUSIONS

This paper has summarized a number of CFD studies focused on the aerodynamics of high speed gear windage losses. The goals of this work have been to explore the roles of shrouds in these systems towards improved physical understanding and design guidance. The following conclusions apply: (1) Angular momentum confinement near the pitch-line-face region plays an important role in shroud effectiveness. (2) As postulated by other researchers, tooth channel mass flow rate restriction due to the shrouds is also important in reducing windage loss, and is the dominant mechanism for very small clearance radial shrouds. (3) Even relatively small leakage paths in the shroud can reduce the shroud's effectiveness due to the loss of angular momentum confinement. (4) A plausible set of droplet models, separately validated for aerodynamic application, return significantly increased losses associated with oil lubrication. (5) The preliminary multi-phase studies undertaken demonstrate a richness of physics that we have only begun to explore, including significantly larger predicted losses than suggested by homogeneous assumptions, important local droplet concentration effects, and the necessity of incorporating film modeling.

Current and future work is focused on: 1) improvement and validation of two fluid models (e.g., splashing), 2) larger time accurate models of shrouds with realistic penetrations, 3) validation of multiphase analysis with forthcoming data from NASA Glenn's Gearbox Windage Facility, 4) further evolution of film modeling, and, 5) incorporation of gear meshing dynamics and thermodynamics.

## ACKNOWLEDGMENTS

This work was supported by NASA under NASA Cooperative Agreement No. NNX07AB34A. Dr. Robert Handschuh is the Technical Monitor.

## REFERENCES

1. Hill, M.J., Kunz, R.F., Noack, R.W., Long, L.N., Morris, P.J., Handschuh, R.F., "Application and Validation of Unstructured Overset CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulations," American Helicopter Society 64th Annual Forum, Montreal, Canada, April, 2008.

2. Hill, M.J., Kunz, R.F., Medvitz, R.B., Handschuh, R.F., Long, L.N., Noack, R.W., "CFD Analysis of GearWindage Losses: Validation and Parametric Aerodynamic Studies," American Helicopter Society 66th Annual Forum, Phoenix, AZ, May, 2010.

3. Hill, M.J., Kunz, R.F., Medvitz, R.B., Handschuh, R.F., Long, L.N., Noack, R.W., Morris, P.J., "CFD Analysis of GearWindage Losses: Validation and Parametric Aerodynamic Studies," ASME Journal of Fluids Engineering, Vol. 133, March, 2011.

4. Hill, M.J. "A Computational Investigation of Gear Windage, PhD Thesis, Department of Aerospace Engineering, Penn State University, August, 2010.

5. Diab, Y., Ville, F., Velex, P., and Changenet, C., 2004, "Windage Losses in High Speed Gears—Preliminary Experimental and Theoretical Results," ASME Journal of Mechanical Design, 126(5), pp. 903–908.

6. Handschuh, R. F., and Hurrel, M. J., 2010, NASA/TM-2011-216925, prepared for the International Conference on Gears, Garching, Germany, October 4-6.

7. Al-Shibl, K., Simmons, K., and Eastwick, C. N., 2007, "Modelling Windage Power Loss From an Enclosed Spur Gear," Proceedings of the Institution of Mechanical Engineers, Part A, 221, pp. 331–341.

8. Eastwick, C. N., and Johnson, G., 2008, "Gear Windage: A Review," ASME Journal of Mechanical Design, 130(3), pp. 03400-1-6.

9. Marchesse, Y., Changenet, C., Ville, F., Velex, P., 2011, "Investigations on CFD Simulations for Predicting Windage Power Losses in Spur Gears," ASME Journal of Mechanical Design, 133(2), pp. 024501-1-7.

10. Imai, H., Goi, T., Arisawa, H., Nishmimura, M., 2009, "Reduction in Gear Windage and Churning Loss by Optimum Shroud Design With Aid of CFD," JSME International Conference on Motion and Power Transmission.

11. Dawson, P.H., 1984, "Windage Loss in Larger High-Speed Gears," Proceedings of the Institution of Mechanical Engineers, Vol. 198A, No. 1, pp. 51-59.

12. Dawson, P. H., 1988, "High Speed Gear Windage," General Electric Company (Great Britain) Review, 4(3), pp. 164–167.

13. Winfree, D.D., 2000, "Reducing Gear Windage Losses From High Speed Gears," Proceedings ASME Power Transmission and Gearing Conference, Baltimore, Maryland, September.

14. Kunz, R.F., Yu, W.S., Antal, S.P., Ettorre, S.M., 2001, "An Unstructured Two-Fluid Method Based on the Coupled Phasic Exchange Algorithm," AIAA Paper 2001-2672, presented at the 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, June.

15. Michaelides, E. E., 2006, Particles, Bubbles & Drops: Their Motion, Heat And Mass Transfer, World Scientific Publishing Company.

16. Lopez de Bertodano, M., 1992, "Turbulent Bubbly Two-Phase Flow in a Triangular Duct," Ph.D. Thesis, Nuclear Engineering, Rensselaer Polytechnic Institute.

17. Haworth, D.C., Kunz, R.F., Leemhuis, L.S., Davison, A.C., Kriete, A., 2002, "Eulerian Multiphase CFD Analysis of Particle Transport and Deposition in the Human Lung," 55th Annnal Meeting of the APS Division of Fluid Dynamics, Austin, TX, November.

18. Schadel, S.A., Hanratty, T.J., 1989, "Interpretation of Atomization Rates of the Liquid Film Iin Gas-Liquid Annular Flow," International Journal Multiphase Flow,' 15(6) 15, pp. 893-900.

19. Papadakis, M., Hung, K, Vu, Giao, Yeong, H.W., Bidwell, C., Breer, M., Bencic,T., 2002, "Experimental Investigation of Water Droplet Impingement on Airfoils, Finite Wings, and an S-Duct Engine Inlet" NASA Gleun Research Center, NASA/TM—2002-211700, October.

20. Wright, W., 2005, "Validation Results for LEWICE 3.0," AIAA Paper 2005-1243. 43rd AIAA Aerospace Sciences Meeting and Exhibit, 10-13 January, Reno, Nevada.

## Summary

As of the writing of this final report, our group's research into gearbox windage aerodynamics continues. The physics understanding, and spoiler modification that it led to, have resulted in a US Patent. This patent has interested two major US rotorcraft manufacturers, Bell and Boeing, to invest in sponsored research of our group to explore its applicability to helical and spiral-bevel systems. Both of these efforts led to CFD demonstration that such modifications should improve the loss performance of rotorcraft transmissions. The exposure of our NASA sponsored research in this area to the wider aerospace community has led directly to sponsorship from Pratt and Whitney for exploration of the aerodynamics in their fan-drive gearbox. The multiphase modeling research supported in this NRA has similarities to some of the requirements of rotorcraft (and fixed wing) icing modeling (specifically, droplet and film physics), and this has benefitted our separate research in icing CFD modeling (to be published in this summer's Asian Rotorcraft Conference). Lastly, our progress has led directly to our current 5 year VLRCOE project on rotorcraft loss-of-lubrication modeling, of which windage is an important component.

# Appendix

# N-PHASE-PSU Version 3.16: Theory, User's and Reference Manual

Robert F. Kunz
The Pennsylvania State University
University Park, Pennsylvania 16804

*Rev. 3.0, 12/28/05*
*Rev 3.1, 1/27/06*
*Rev 3.2, 8/11/06*
*Rev 3.3, 10/7/06*
*Rev. 3.4, 10/18/06*
*Rev. 3.5, 10/29/06*
*Rev. 3.6, 12/8/06*
*Rev 3.7, 12/15/06*
*Rev 3.8, 1/11/07*
*Rev. 3.9, 1/18/07*
*Rev. 3.10, 3/30/07*
*Rev. 3.11, 5/12/08*
*Rev. 3.12, 10/23/08*
*Rev. 3.13, 5/18/09*
*Rev. 3.14, 8/10/11*
*Rev. 3.15, 3/12/13*
*Rev. 3.16, 5/31/13*

## Table of Contents

## Executive Summary

This document represents the evolving formal documentation of the NPHASE-PSU computer code. Version 3.15 is being delivered along with the software to NASA in 2013.

Significant upgrades to the NPHASE-PSU have been made since the first delivery of draft documentation to DARPA and USNRC in 2006. These include a much lighter, faster and memory efficient face based front end, support for arbitrary polyhedra in front end, flow-solver and back-end, a generalized homogeneous multiphase capability, and several two-fluid modelling and algorithmic elements. Specific capability installed for the NASA Gearbox Windage Aerodynamics NRA are included in this version: · Hybrid Immersed Overset Boundary Method (HOIBM) [Noack et. al (2009)] · Periodic boundary conditions for multiple frames of reference, · Fully generalized immersed boundary method, · Fully generalized conjugate heat transfer, · Droplet deposition, bouncing, splashing models, and, · Film transport and breakup.

## Overview of NPHASE-PSU

NPHASE is a CFD code developed by Robert Kunz at the Penn State University Applied Research Laboratory (PSU-ARL) and Steve Antal at Rensselaer Polytechnic Institute (RPI). The code has been under development since 1998. Since NPHASE Version 2.0 was established in 2000, two separate versions of the code have been developed independently by Kunz and Antal. This document applies to the version developed by Kunz at ARL Penn State, NPHASE-PSU which is distributed for free to three sponsoring US government agencies: NASA, DARPA and the USNRC.

NPHASE-PSU is not a commercial CFD code, nor is it used for commercial consulting. NPHASE-PSU is open source. The mission of the software developer is to support government and industrial sponsors of programs related to PSU-ARL's core research activities. The software is owned by the Pennsylvania State University and is distributed freely to research sponsors (including all source, tutorials, documentation, and front/back-end processing tools).

NPHASE-PSU is written in standard ANSI-C, with a smattering of C++, and compiles under (at least) the open-source GNU C compiler, gcc/g++ and (in 2013) the intel LINUX compiler, icc. NPHASE-PSU refers to the CFD code itself, but employs several front-end and back-end processing tools for domain decomposition and reassembly, grid readers for standard COTS formats, pointer topology construction, and writers to standard postprocessing software file formats. These processing codes are written in FORTRAN 77/90, ANSI-C, and C++. Accordingly, if the user wishes to modify these front/back-end programs they must also have access to a FORTRAN 90 compiler. NPHASE-PSU also requires several open source software libraries including MPI2, PETSC, METIS and SUGGAR/DirtLib each of which must be installed with NPHASE-PSU on the system. Although the code has in the past been installed on Windows and SGI systems, the present delivered version, V3.15, is verified to install and run only on desktops and clusters running LINUX.

NPHASE-PSU has the following characteristics, features, and capabilities:

- Arbitrary number of fields and/or species, where different species are assumed to be in dynamic and thermodynamic equilibrium, and different fields are not (i.e. have different velocities and enthalpies). Mass fraction and volume fraction transport options are available for species/field transport.
- Numerous interfacial mass, momentum, energy and turbulence exchange models associated with multiphase flow simulations.
- 3D unstructured: Arbitrary polyhedral formulation with front-back ends supporting 4 standard element types: tetrahedral, hexahedra, pyramids, prisms, as well as completely arbirtrary element types (n-faced polyhera).
- Overset mesh capability, utilizing open source Suggar and DirtLib software.
- Moving and deforming mesh capability (Geometric Conservation Law satisfying).
- Fully matrix level parallelized using MPI2 and domain decomposition.
- METIS used for domain decomposition embedded in front end.
- PETSC and simple point linear equation solvers.
- All-Mach number formulation: incompressible, weakly compressible, strongly compressible flows. Isothermal, Boussinesq and perfect gas single-phase compressible state relations are available.
- Segregated pressure based algorithm and CPE algorithm for multiphase flow.
- Face based finite volume scheme: $1^{st}$ through $3^{rd}$ order accurate convection discretization schemes, $2^{nd}$ order accurate viscous term discretization.
- $1^{st}$ and $2^{nd}$ order, dual time based temporally accurate formulation.
- Several low and high Reynolds number form 1-equation, 2-equation, and v2f turbulence models.
- Structural mechanics coupling to NASTRAN.
- Radiation heat transfer coupling to RADTHERM.
- Numerous “specialty” face and volume elements (conducting solid regions, porous regions, various quasi-1D conjugate heat transfer boundaries).
- Full turbomachinery capability (rotating and non-rotating reference frames) including rotor-stator interaction and body force modeling.
- “Light” face based file formats supported in front end.
- ENSIGHT file format supported in back end.
- Coded purely in ANSI-C, and C++ with some front and back end utilities coded in F77, F90.

Partial development (features that are not fully implemented but are in source code in various stages of completion):

- Non-isotropic mesh adaption
- Full Reynolds Stress modeling
- Conformation tensor transport
- VOF for discrete interfaces
- 6DOF dynamics
- Fully coupled parallel algorithm

NPHASE-PSU has been applied to and validated against a broad range of complex single-phase and multiphase configurations including:

- Gas-particle flows through a branching pipe junctions and human lung geometries
- Bubble column reactors
- Full-annulus rotor-stator pump and turbine stage analyses, including rotor-stator interactions
- High Reynolds number submarine configurations at a range of angles of attack
- Power plant cooling ponds
- Microbubble drag reduction applications of relevance to DARPA
- Geometrically complex UUV (MRUUV) and SEAL deliverivery vehicles (ASDS)
- Several surface ship configurations (5415, Athena)
- High speed maritime lifting pod
- Micro-flows of biological cell systems
- Numerous multiphase flows of relevance to the NRC (thermally driven counter-current reactor flows, 2-phase duct and pipe flows)
- DES simulations of urban/atmospheric dispersion
- Bubbly surface ship wakes
- Thermal management of tank engine compartment
- Thermal management of eco-friendly structures.
- Droplet impingement in compressible aerodynamics flows for icing simulation
- Gear aerodynamics

Documentation of many of these cases appears in Kunz et al. (2001, 2003, 2007, 2011) or can be obtained from the author.

# NPHASE-PSU Theory Manual

## *Governing Equations*

The single-pressure ensemble averaged continuity and momentum equations are cast in conservation law form as:

$$\frac{\partial \alpha^k \rho^k}{\partial t} + \frac{\partial \alpha^k \rho^k u_j^k}{\partial x_j} = \sum_{k \neq l} \left( \Gamma^{lk} - \Gamma^{kl} \right) \tag{1}$$

$$\frac{\partial \alpha^k \rho^k u_i^k}{\partial t} + \frac{\partial \alpha^k \rho^k u_i^k u_j^k}{\partial x_j} = -\alpha^k \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[ \alpha^k \mu_t^k \left( \frac{\partial u_i^k}{\partial x_j} + \frac{\partial u_j^k}{\partial x_i} \right) \right] + \rho^k \alpha^k g_i + M_i^{kl} + \sum_{k \neq l} \left( D^{kl}\left[ u_i^l - u_i^k \right] + \Gamma^{lk} u_i^l - \Gamma^{kl} u_i^k \right) \tag{2}$$

Superscripts k and l designate donor and receptor fields for mass transfer ( $\Gamma^{kl}$ ), and drag ( $D^{kl}$ ) and non-drag ( $M_i^{kl}$ ) interfacial forces. In general each field, k, will have a different density, volume fraction, velocity and viscosity. For single phase flow, equations (1)-(2) reduce to:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0 \tag{3}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[ \mu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \rho g_i \tag{4}$$

For homogeneous multiphase flow, it is assumed that the fields are in dynamic and thermodynamic equilibrium, and equations (1)-(2) reduce to:

$$\frac{\partial \alpha^k \rho^k}{\partial t} + \frac{\partial \alpha^k \rho^k u_j^k}{\partial x_j} = \sum_{k \neq l} \left( \Gamma^{lk} - \Gamma^{kl} \right) \tag{5}$$

$$\frac{\partial \rho^m u_i^m}{\partial t} + \frac{\partial \rho^m u_i^m u_j^m}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[ \mu_t^k \left( \frac{\partial u_i^m}{\partial x_j} + \frac{\partial u_j^m}{\partial x_i} \right) \right] + \rho^m g_i \tag{6}$$

where the set of momentum equations is reduced to a single equation for the mixture. Superscript m represents mixture quantities. In equations (1)-(6), a high Reynolds number form viscous term is assumed with dilatation and turbulence energy terms neglected (although these terms are available in NPHASE-PSU). Energy and turbulence equations are considered below.

### Physical Modeling

#### *Generalize Field Transport*

The generalized n-field formulation in equations (1)-(2) can be applied to non-equilibrium multiphase flows in two ways. The more fundamental approach involves solving mass and momentum equations for each field that is present. For example, in the context of disperse bubbly flows, one could solve a single continuous liquid field and a

number of disperse fields, "binned" by size. In this approach each bubble field exchanges momentum with the continuous field through drag and non-drag interfacial forces which depend in magnitude on the local interfacial area density of that field, $A_{int}=6\alpha^{gas}/D_b$ (for spherical bubbles). This approach was used in our earlier work [Kunz et al., (2003, 2007)], where up to 11 bubble fields were solved.

*Interfacial Area Density Transport*

An alternative is to solve a single mass and momentum equation for each *phase* that is present and to accommodate the variation in dynamics due to phase interface evolution by modelling and solving for interfacial area transport. For example, in the context of disperse bubbly flows, a single gas field continuity and momentum equation would be solved, and an interfacial area density transport (IADT) equation would also be solved to determine a local mean characteristic diameter for the bubbles. This approach significantly reduces the model's CPU requirements compared to solving an (N+1)-field system (N bubble fields). The numerical complexity associated with interfield transfer terms is also reduced considerably.

Since mass transfer can be fully accommodated in the context of IADT (details presented below), the physical appropriateness of employing IADT rests on whether the interface dynamics can be sufficiently captured using a single local mean inter-phase interfacial area, with an assumed/modeled distribution of characteristic size/shape about that mean. This is demonstrated to be the case for an example calculation below. Currently in NPHASE-PSU, a generalized IADT formulation is available, with physical models in place to accommodate disperse bubbly flows related to Microbubble Drag Reduction (hereafter MBDR). In this context mass transfer corresponds to coalescence and breakup (between bubbles of different sizes). The IADT formulation in NPHASE-PSU is presented here, in that context, although any interface evolution (e.g., annular flow, droplet laden gas flows) can be modeled through addition of subroutines corresponding to those available for the disperse bubbly flow models currently available in V3.1.

Following Hibiki et al. (2001), the IADT equation with source terms for breakup and coalescence can be written:

$$\frac{\partial a_i}{\partial t}+\frac{\partial\left(a_i u_{g,j}\right)}{\partial x_j}=\Phi_B+\Phi_C \tag{7}$$

where $a_i$ is the interfacial area density, $u_{g,j}$ are the gas phase velocity components, and $\Phi_b$ and $\Phi_c$ are source terms for breakup and coalescence, respectively. The interfacial area density is defined as:

$$a_i=\frac{6\alpha_g}{\overline{D}} \tag{8}$$

where $\alpha_g$ is the volume fraction of the gas phase and $\overline{D}$ is the mean bubble diameter. The source terms are rates of change of interfacial area concentration, written as:

$$\Phi_B=\frac{1}{3\psi}\left(\frac{\alpha_g}{a_i}\right)^2\phi_b,\Phi_C=\frac{-1}{3\psi}\left(\frac{\alpha_g}{a_i}\right)^2\phi_c \tag{9}$$

where $\phi_b$ and $\phi_c$ are the rates of change of bubble number density ($1/m^3s$) due to breakup and coalescence, respectively. The factor $\psi$ depends on the bubble shape, here taken as spherical, so $\psi=1/(36\pi)$. The particular models used for breakup and coalescence for MBDR are presented below.

Figure 1 illustrates that the dynamics of MBDR can be sufficiently captured using a single local mean gas-liquid interfacial area, with an assumed/modeled distribution of bubble size about that mean. Three MBDR cases are considered, corresponding to three gas injection rates at injector plates near the leading edge of a very high Reynolds number flat plate flow (see "HIPLATE" tutorial below). First, each case was run with three bubble fields using an approximation to the experimentally measured bubble size distribution. Then each case was run using a single gas field and interfacial area density as described above. For these comparisons no coalescence or breakup was incorporated so as to isolate the effect of the different interfacial dynamics modeling approaches. Details of the HIPLATE simulations are provided below, but Figure 1 serves to illustrate that incorporating interfacial area density has only a small impact on accuracy of drag reduction and bubble velocity predictions for MBDR.

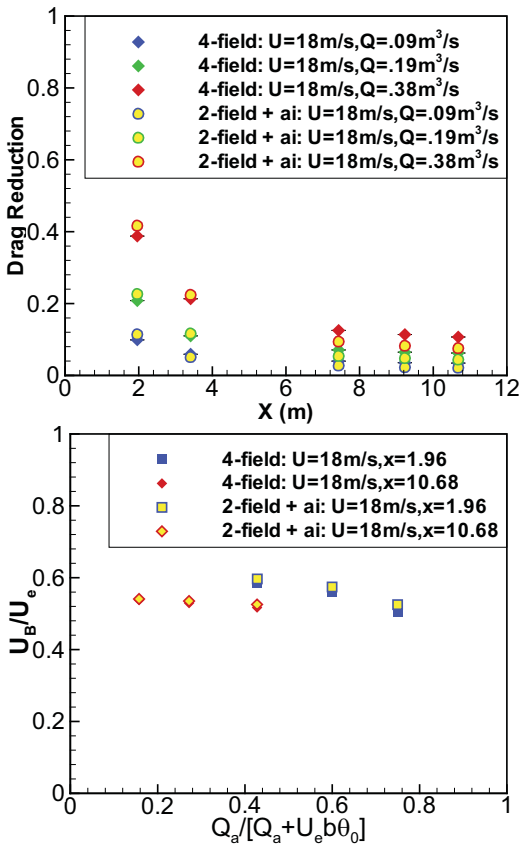



**Figure 1. Comparison of 2-field and 4-field simulations for $U_\infty$=18 m/s HIPLATE cases. (top) Drag reduction vs. x, (bottom) Normalized bubble velocity vs. normalized flow rate.**

*Interface Dynamics*

**Overview**

The structure of NPHASE-PSU supports arbitrary forms for drag ( $D^{kl}$ ) and non-drag ( $M_i^{kl}$ ) interfacial dynamics models that appear in equation (2). The focus of two-fluid NPHASE-PSU research performed to date has been in the context of disperse bubbly flows (where the single continuous field is liquid) and disperse particle flows (where the single continuous field is gaseous). Accordingly the physical model set that currently resides within V3.1 are appropriate for these interface dynamics.

A suite of bubble dynamics models have been developed, adapted from the open literature, and calibrated over the course of the development of NPHASE-PSU. These interfacial force models are summarized here. These models have been used in the context of a full-up n-bubble-field formulation where the bubble diameters that appear and hence the implied interfacial area between that field and the liquid are unique to and representative of that field. As indicated in the previous section, these models are also implemented in the context of a single gas field represented by a mean bubble diameter and attendant interfacial area density, which is transported and evolved with the flow.

**Drag**

In the context of particles, a conventional corrected Stokes drag law is incorporate

$$D^{kl} = \frac{1}{8}\rho^{gas} C_D \left| u_i^l - u_i^k \right| a_i, a_i = \frac{6\alpha^{gas}}{D_p}, C_D = \frac{24}{Re_p} f_D\left(Re_p\right) \tag{10}$$

where the local particle Reynolds number is $Re_p = \rho^{gas}\left|\underline{V}^{rel}\right| D_p / \mu^{gas}$. The solid particle drag-law correction used [Loth (2000), for example] is:

$$\begin{aligned} f_D &= 1. + 0.1875\, Re_p && \text{for } Re_p \le 1 \\ f_D &= 1. + 0.1935\, Re_p^{.6305} && \text{for } 1 < Re_p \le 285 \\ f_D &= 1. + 0.015\, Re_p + 0.2283\, Re_p^{.427} && \text{for } 285 < Re_p \le 2000 \\ f_D &= 0.44\, Re_p / 24. && \text{for } 2000 < Re_p \le 3.5x10^5 \end{aligned} \tag{11}$$

In the context of bubbles, drag models have been implemented for spherical bubbles in seawater, clean fresh water and contaminated (tap) water. Again, a corrected Stokes drag law is employed:

$$D^{kl} = \frac{1}{8}\rho^{liq} C_D \left| u_i^l - u_i^k \right| a_i, a_i = \frac{6\alpha^{liq}}{D_b}, C_D = \frac{24}{Re_b} f_D\left(Re_b\right) \tag{12}$$

where the local bubble Reynolds number is $Re_b = \rho^{liq}\left|\underline{V}^{rel}\right| D_b / \mu^m$.

For fresh water without impurities, the drag-law correction [Loth (2000), for example] is:

$$\begin{aligned} f_D &= 1. + 0.1875\, Re_b && \text{for } Re_b \le 0.1 \\ f_D &= 1. + 0.0565\, Re_b^{.525} && \text{for } 0.1 < Re_b \le 500 \end{aligned} \tag{13}$$

For contaminated (tap) water, the drag-law correction for solid spheres equation (11), is used. For seawater, a drag-law correction due to Detch (1991) is available.

In addition to water purity, locally high gas volume fraction and bubble deformation can influence the drag, so corrections to the spherical bubble, disperse flow models in equations (11) and (13) may be appropriate. For uniformly disperse flows, an increased drag coefficient is appropriate [Richardson-Zaki (1954), for example], and for flows where gas structures are streamlined (bubble columns, sheets) a reduced drag coefficient is appropriate. This latter effect likely is important in the near injector region of MBDR flows, where application of the standard disperse flow model gives rise to too much local drag, thereby inhibiting the penetration of the injected gas into the boundary layer. This observation became clear in the course of the HIPLATE validation studies, where a significant defect in measured bubble velocity could not be obtained unless a "cluster" drag model was incorporated. Specifically, a model proposed by Johansen and Boysan (1988) has been adapted to an Eulerian framework:

$$C_D = C_{D0}\left(1 - 1.54\left[\mathrm{MIN}\left(.5157, \alpha^{gas}\right)\right]^{2/3}\right) \tag{14}$$

where $C_{D0}$ is the original drag coefficient in equations (11) or (13), $\alpha^{gas}$ is the total gas volume fraction and the MIN function is provided to ensure that the corrected drag coefficient does not drop to below 1% of the uncorrected value. The importance of incorporating such a cluster drag form is demonstrated in Kunz et al. (2007).

For air-water droplet flows of aerodynamic interest, a droplet drag model is implemented Michaelides (2006) is employed:

$$C_D = 0.4 + \frac{24}{Re_d} + \frac{6}{1.0 + \sqrt{Re_d}}$$

**Virtual Mass**

Virtual mass is modeled following Lahey and Drew (2000):

$$\underline{M}_{VM}^{liq-gas} = \alpha^{gas}\rho^{liq}C_{VM}\left[\frac{D\underline{V}^{gas}}{Dt} - \frac{D\underline{V}^{liq}}{Dt}\right] \tag{15}$$

**Lift**

The lift model employed in the NPHASE-PSU also follows Lahey and Drew (2000):

$$\underline{M}_{LIFT}^{liq-gas} = \alpha^{gas}\rho^{liq}C_l\underline{V}^{rel} \times \nabla \times \underline{V}^{liq} \tag{16}$$

**Wall Lift**

An empirical turbulent near-wall bubble lift force has been implemented based on the formulation of Kawamura and Yoshiba (2004). This force can be thought of as a repulsive force due to wall collisions. The form of the wall-lift force used is:

$$\begin{aligned} F_{WL} &= C_{WL}\left(\pi D_b^3/6\right)\rho^{liq}\left(k/D_b\right)F_{damp} \\ F_{damp} &= 0.5\left[1 - \tanh\left(y_{wall}/D_b - 1.5\right)\right] \end{aligned} \tag{17}$$

where $F_{damp}$ decays the force to zero away from the wall and the model constant used here, $C_{WL} = 0.012/\sqrt{1 + St_k}$, is significantly smaller than that proposed by Kawamura and Yoshiba. The Stokes number is defined as $St_k = D_B^2\rho^{liq}\varepsilon/(18k\mu_m)$.

**Turbulence Dispersion**

The homogeneous turbulence dispersion model is implemented in the framework of the Carrica, et al. (1999) gradient diffusion force model. The general expression for the dispersive force per unit volume ($N/m^3$) may be written as:

$$\underline{M}_i^{k,TD} = -\rho^{liq} \frac{\nu_t}{Sc} C_{TD} \frac{\partial \alpha^k}{\partial x_i} \tag{18}$$

where $C_{TD}$ is the turbulent dispersion coefficient (units $s^{-1}$). For the Carrica, et al. (1999) model, $C_{TD}$ is defined as:

$$C_{TD} = \frac{3}{8} \frac{C_D}{R_b} \left| \underline{u}_{rel}^k \right| \tag{19}$$

where $\underline{u}_{rel}^k$ is the relative velocity between the continuous phase and disperse phase "k".

At the high gas volume fractions, dispersion is enhanced by collisions among bubbles. A new dispersion model has been developed, based on the collision frequency from the Prince-Blanch (1990) coalescence model. This dispersion mechanism is used in addition to one of the homogeneous turbulence dispersion models discussed above. Since DNS computations [Maxey, et al. (2005)] show a significant effect of collision on dispersion for high gas volume fractions, a heuristic dispersion model based on the bubble collision rate has been formulated. The collision-induced dispersion model is implemented in the framework of the [Carrica, et al. (1999)] gradient diffusion force model, equation (18).

We assume the dispersion model coefficient is an unknown function of the "dispersive collision rate", which excludes bubbles that coalesce. To properly formulate the coefficient relationship, the collision rate must be normalized. For that purpose we choose a turbulent characteristic bubble response time ($\tau_{BC}^j$) defined as:

$$\tau_{BC}^j = \frac{4}{3} \frac{D_j}{C_D \left| \underline{u}_{rel}^j \right|} \tag{20}$$

where *k* is the turbulent kinetic energy. Note that this is the bubble response time normally used to define the Stokes number,

$$St = \frac{\tau_{BC}^j}{\tau_c} \tag{21}$$

Alternative characteristic times were evaluated with some success; however, the above relation is a reasonable choice with physical basis.

The normalized dispersive collision rate ($\theta_{ij}^{TD}$) for bubbles "i" and "j" with an equivalent volume $V_{ij}$ is written as:

$$\bar{\theta}_{ij}^{TD} = \theta_{ij}^T V_{ij} \left(1 - \lambda_{ij}\right) \tau_{BC}$$
$$V_{ij} = \left(V_i + V_j\right)/2 \tag{22}$$

The turbulent dispersion coefficient (for a bubble "j") $C_{TD,j}^{coll}$ is chosen to be proportional to the square root of the dispersive collision rate (normalized by a representative time scale, $\tau_{BC}^{j}$).

$$C_{TD}^{coll} = \frac{\hat{C}_{TD}}{\tau_{BC}} \sum_{i} \left[\overline{\theta}_{ij}^{TD}\right]^{1/2} \quad (23)$$

where $\hat{C}_{TD}$ is a constant to be determined. Note that the square root is chosen to obtain a consistent relation with the collisional pressure identified by Maxey, et al. (2005). The Brown DNS calculations confirmed the functional dependence of collisional pressure on volume fraction.

Further modifications to the dispersion model are required to treat other conditions, especially limiting cases with high gas volume fraction. A heuristic model as been implemented for the dispersion models and bubble lift model.

The dispersion in NPHASE-PSU is modeled by summing the two contributions discussed above (equations (19) and (23)), i.e.,

$$\underline{M}_i^{k,TD} = -\rho^{liq} \frac{\nu_t}{Sc} \left(C_{TD} + C_{TD}^{coll}\right) \frac{\partial \alpha^k}{\partial x_i} \quad (24)$$

In general this relation applies to each bubble field "k".

*Deposition*

Deposition of droplets or particles can be modelled as inter-field exchange through mass transfer in the context of full-two fluid modelling or interfacial area density. However, in some applications, including particle deposition within the lung or droplet deposition onto an airfoil, it may be suitable to model the process as a mass sink. Specifically, once can assume that when the mass deposits on the wall it "leaves the domain" as in a transpiration condition. In NPHASE-PSU this is handled using deposition models that simply extract mass and attendant momentum from the wall adjacent cells.

For droplet deposition a simple inviscid impaction model is adopted wherein, all droplet momentum oriented into the wall is extracted along with its mass:

$$\dot{m}_{deposition} = \int_{\underline{A}_{wal/}} \rho^d \alpha^d \underline{V}^d \bullet d\underline{A}_{wall}$$

where the droplet velocity is taken as the nodal value adjacent to the wall. This flux is consistently extracted from the droplet continuity and momentum equations. Other models are available in the code including diffusion based deposition for co-flowing systems, but this is not yet documented here.

## *Breakup and Coalescence - Multi-Bubble-Field Formulation*

**Overview**

A general formulation for discrete bubble size distributions based on the approach of Kumar and Ramkrishna (1996) has been implemented in NPHASE-PSU. The approach allows one to rigorously conserve two functions of the bubble distribution function (or kernel) regardless of the discrete bubble sizes (bins) selected. There is a unique formulation for coalescence and breakup of bubbles. In both cases we have chosen to conserve volume moments of the bubble number density distribution function, *n(v,t)*, i.e.,

$$M_{\mu} = \int_{0}^{\infty} v^{\mu} n(v,t)\, dv \qquad (25)$$

where *v* denotes the bubble volume and *t* is time. Of course, the distribution function is a function of spatial location as well. We have chosen the zero-th (μ=0) and first (μ=1) moments at present, though the coding permits arbitrary moments to be conserved. The rational for this choice is the conservation of the number of bubbles and the volume of bubbles during the coalescence and breakup process. Though the interfacial area of the bubbles is an important quantity in two-phase bubbly flows, it is not conserved in general. The correct interfacial area will be preserved by conserving the number of bubbles and the volume of the bubbles.

It should be noted that other investigators [e.g., Carrica, et al (1999)] have used formulations based on bubble mass, since in cases with significant gas compressibility the bubble mass is conserved while the volume changes (in the absence of either coalescence or breakup). The present implementation of the Kumar-Ramkrishna scheme in NPHASE-PSU easily permits the use of bubble mass as the bubble size metric rather than bubble volume, if necessary.

**Prince-Blanch Coalescence Model**

For coalescence, the rate kernel employed is due to Prince and Blanch (1990) and Williams and Loyalka (1991). The latter text offers a fairly complete description of the physics of coalescence and various mathematical approaches for modeling the various coalescence mechanisms. Three primary mechanisms may be included in the coalescence kernel – (1) turbulent diffusion, (2) “laminar” shear, and (3) buoyancy. The so-called “laminar” shear contribution is modeled as a function of the local velocity gradient, and is relevant only for laminar flow and therefore, is not considered. The formulation of Prince and Blanch models the effect of turbulent diffusion due to “small” eddies, while the formulation of Williams and Loyalka also purports to model the effect of small eddies, though with an approach that relies on a bubble scale that is small compared to the scale of the turbulence. Hence the Williams and Loyalka formulation may not apply to the bubble sizes expected to be present in microbubble drag reduction applications.

The turbulent diffusion contribution is due to a statistical average of the fluid velocity fluctuations. However, a general turbulent flow also has a mean velocity gradient which has an effect on collisions. Williams and Loyalka discuss the impact of a laminar flow velocity gradient on coalescence. For the present application their formulation was adapted to treat the mean velocity gradient effect in turbulent flow.

The coalescence model considering turbulent diffusion due to small-scale turbulence and mean-shear is operational in NPHASE-PSU. The effect of buoyancy on coalescence has been neglected.

The turbulent collision rate is a dominant factor in bubble coalescence according to both Prince and Blanch (1990), and Williams and Loyalka (1991) . For small eddies the turbulence is assumed to be isotropic (at least on the scale of the bubble diameter) and the bubble size is assumed to lie in the inertial subrange. The same assumption is made in the breakup model formulation discussed below. Following Prince and Blanch, the collision frequency [ $\theta_{ij}^{T}$ ,1/( $m^3$ s)] between bubbles *i* and *j* due to turbulent motion may be written:

$$\theta_{ij}^{T} = n_i n_j S_{ij} \left( \overline{u_i^2} + \overline{u_j^2} \right)^{1/2} \quad (26)$$

where $n_i$ and $n_j$ are the number densities ($m^{-3}$) of bubbles with diameters $D_i$ and $D_j$, respectively. Also, $\overline{u_i^2}$ is the root mean square of the fluctuating velocity of bubble *i* and $S_{ij}$ is the collision cross-sectional area defined by Prince and Blanch:

$$S_{ij} = \frac{\pi}{16} \left( D_i + D_j \right)^2 \quad (27)$$

The required fluctuating velocity in the inertial subrange for isotropic turbulence is given by Prince and Blanch:

$$u_i = \sqrt{2} \left( \varepsilon D_i \right)^{1/3} \; u_i = \sqrt{2} \left( \varepsilon D_i \right)^{1/3} \quad (28)$$

where the relevant turbulence length scale is assumed to be the bubble diameter. The leading constant in equation (28) is not universally agreed upon in the literature and the turbulence length scale also appears in several different forms, although always as a function of bubble diameter.

Combining the above expressions yields the desired relation for the collision frequency,

$$\theta_{ij}^{T} = n_i n_j \frac{\left( \sqrt{2}\pi \right)}{16} \varepsilon^{1/3} \left( D_i + D_j \right)^2 \left( D_i^{2/3} + D_j^{2/3} \right)^{1/2} \quad (29)$$

The probability that a collision results in coalescence is required to complete the rate kernel formulation. Again, following Prince and Blanch, this probability is termed the collision efficiency and is a function of the contact time between bubbles and the time required for bubbles to coalesce. For a pair of bubbles, this efficiency ( $\lambda_{ij}$ ) is written as [following Coulaloglou and Tavlarides (1977)]:

$$\lambda_{ij} = \exp\left( -t_{ij} / \tau_{ij} \right) \quad (30)$$

where $t_{ij}$ is the time required for bubbles of diameters $D_i$ and $D_j$ to coalesce and $\tau_{ij}$ is the contact time for the two bubbles. From other literature, Prince-Blanch presented the following expression for the coalescence time ( $t_{ij}$ ):

$$t_{ij} = \left( \frac{\left( 0.5 D_{ij} \right)^3 \rho^{liq}}{16\sigma} \right) \ln\left( \frac{h_0}{h_f} \right) \quad (31)$$

where $h_o$ is an initial film thickness between two bubbles as they just come into contact and $h_f$ is a final critical film thickness where rupture occurs and the bubbles coalesce. The quantity $D_{ij}$ is an equivalent diameter for bubbles of unequal size and is given by:

$$D_{ij} = \left( \frac{2}{D_i} + \frac{2}{D_j} \right)^{-1} \quad (32)$$

For air-water systems, the film thickness values quoted by Prince-Blanch (from other sources) are

$$h_0 = 10^{-4}\,\mathrm{m}, h_f = 10^{-8}\,\mathrm{m} \quad (33)$$

Finally, an estimate of the contact time ( $\tau_{ij}$ ) for bubbles in turbulent flow was made by Levich (1962) from dimensional analysis. A modification due to the relative velocity between the bubbles is noted by Carrica, et al. (1999), resulting in the following expression:

$$\tau_{ij} = \frac{D_{ch}}{u_{rel,ij} + 2\left(0.5 D_{ch} \varepsilon\right)^{1/3}} \quad (34)$$

where $D_{ch}$ is a characteristic length related to the bubble sizes and $u_{rel,ij}$ is the mean relative velocity between the colliding bubbles. The characteristic length ( $D_{ch}$ ) in equation (34) may be taken as an adjustable parameter in this model. In the absence of better information, $D_{ch}$ will be taken as the average of the inverse of the bubble diameters, $2D_{ch} = \left(D_i^{-1} + D_j^{-1}\right)^{-1}$, as suggested by Carrica, et al. (1999), and Prince and Blanch (1990).

Furthermore, all quantities in the model are assumed to be statistical averages for a turbulent flow, thus further uncertainties in the model may result. There appears to be very little data or analysis in the literature addressing these complex issues.

**Lehr-Mewes Breakup Model**

For breakup, one rate kernel investigated is due to Lehr and Mewes (2001). This kernel has some important properties that allow the formation of a small bubble and a large bubble when a large bubble breaks up. The breakup mechanism considered is due to small-scale turbulent eddies. The Kumar-Ramkrishna (1996) formulation for breakup requires evaluation of volume integrals of the rate kernel, and the form of the kernel has some characteristics that can lead to numerical problems if not addressed carefully.

The Lehr-Mewes rate kernel for the (binary) breakup of a "mother" bubble with non-dimensional volume $\hat{x}_k$ into daughter bubbles with non-dimensional volumes $\hat{v}$ and $(\hat{x}_k - \hat{v})$ is given by:

$$r_1(v, x_k) = C_{LM} \frac{\hat{x}_k^{1/3}}{\hat{v}^{4/3}} \left[ F_{min}(\hat{v}) - \frac{1}{\hat{x}_k^{7/9}} \right] \quad \text{for } \hat{v} \le \frac{\hat{x}_k}{2} \quad (35)$$

The non-dimensional daughter bubble volume $\hat{v}$ is a defined as:

$$\hat{v} = \frac{v}{v_{st}} \tag{36}$$

and $v_{st}$ is related to the maximum stable bubble, $v_{stable}$ , size by:

$$v_{stable} = 2^{3/5} \frac{\pi}{6} \frac{\sigma^{9/5}}{\rho_l^{9/5} \varepsilon^{6/5}} = 2^{3/5} v_{st} \tag{37}$$

where $\sigma$ is the surface tension (N/m) between the gas and liquid phases, $\rho_l$ is the liquid density (kg/m$^3$) and $\varepsilon$ is the turbulence energy dissipation rate (m$^2$/s$^3$). The function $F_{\min}$ is given by:

$$F_{min}(\hat{v}) = \hat{v}^{7/6} \quad \text{for } \hat{v} \le 1$$
$$F_{min}(\hat{v}) = \frac{1}{\hat{v}^{7/9}} \quad \text{for } \hat{v} > 1 \tag{38}$$

Note that the rate kernel is symmetric about $\hat{v} = \hat{x}_k / 2$ which allows its evaluation for $\hat{v} > \hat{x}_k / 2$ using equation (35). Since the rate kernel must be non-negative, equation (35) must be restricted. This leads to a minimum value for a minimum daughter bubble size given by:

$$\hat{v}_{min} = \hat{x}_k^{-2/3} \tag{39}$$

where the rate kernel becomes zero. Also, the rate kernel has a slope discontinuity at $\hat{v} = \hat{v}_{\min}$.

The form of the function $F_{\min}$ also gives rise to a slope discontinuity in the rate kernel at $\hat{v} = 1$, and possibly at $\hat{v} = \hat{x}_k / 2$. Furthermore, the kernel has very large gradients for large non-dimensional bubble sizes. This is shown in Figure 2, where the normalized daughter size distribution for the L-M rate kernel is shown for several values of a volume ratio parameter, *VR*, defined by:

$$VR = \frac{\hat{x}_k}{\hat{v}_{stable}} = \frac{x_k}{v_{stable}} \tag{40}$$

where $v_{stable}$ is the maximum stable bubble volume. As a result of equation (37), the non-dimensional bubble volume is a function of the local flow properties, thus it will vary throughout the flow field.

In general the required moment integrals of the rate kernel required for the K-R formulation cannot be evaluated analytically. The zero-th moment is an exception.

All moments can be evaluated numerically; however for large values of *VR,* accuracy has been shown to be poor unless caution is exercised in selecting the integration step size. This is due to the large gradients shown in Figure 3. An approximate analytical evaluation of the kernel integrals for large *VR* was explored, but proved to be impractical

and did not reduce CPU time. Thus an adaptive procedure was implemented for selecting the integration step size based on a prescribed accuracy in resolving the L-M rate kernel. For a very wide range of mother bubble sizes, this approach requires only a moderate number of integration steps (< 1000) to determine the necessary moments very accurately (< 0.01% error).

**Martinez-Bazan Breakup Model**

Another rate kernel investigated is due to Martinez-Bazan, et al. (1999a, b). This kernel is much different than the Lehr-Mewes kernel in that the formation of a small bubble and a large bubble from a bubble breakup has very low probability. The breakup mechanism considered is due to turbulent eddies and a phenomenological model for the breakup kernel (frequency) was developed using experimental data from a high-Reynolds number water jet flow with bubble injection. The experiments were conducted very carefully to insure that the turbulence in the jet was locally homogeneous, isotropic and in near-equilibrium. The model assumes that the initial bubble size, $D_0$ is in the inertial subrange, i.e., $\eta << D_0 << L_x$ , where $\eta$ is the Kolmogorov microscale and $L_x$ is the integral scale of the turbulence.

$$\eta = \left( \frac{\upsilon^3}{\varepsilon} \right)^{1/4} \tag{41}$$

$$L_x = \frac{\pi E_{11}(k_1 = 0)}{2\overline{u'^2}} \tag{42}$$

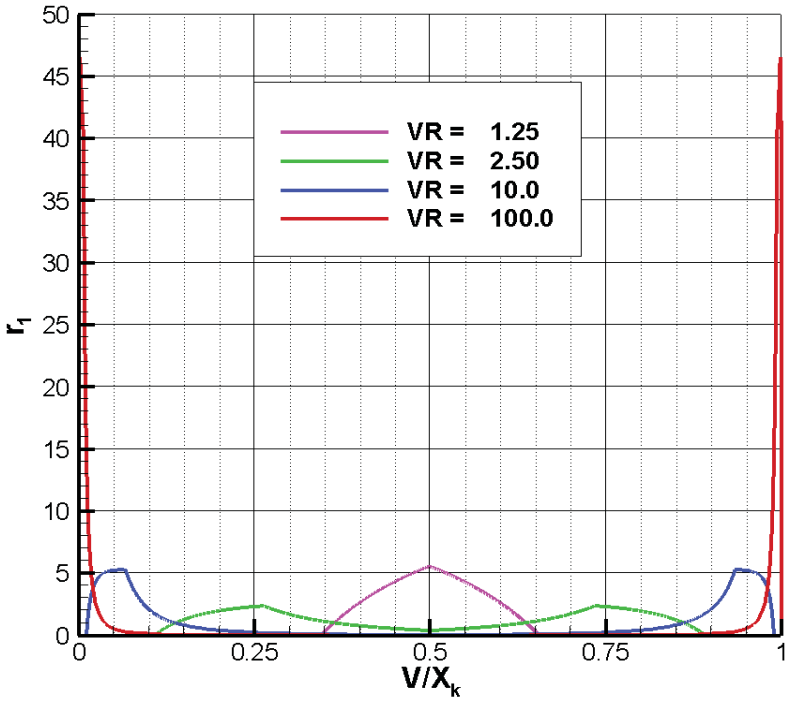



**Figure 2. Normalized daughter size distribution for Lehr-Mewes rate kernel.**

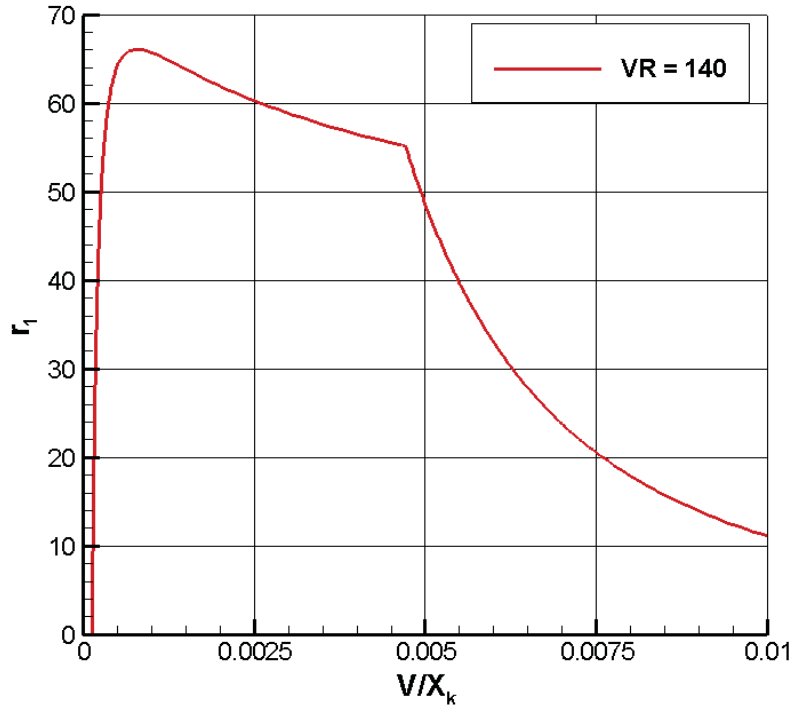



**Figure 3. Normalized daughter size distribution for Lehr-Mewes rate kernel near V/Xk=0, for VR=140.**

where $u'$ is the fluctuating component of axial velocity, $k_1$ is the turbulence wave number in the axial direction and the tensor $E$ is the turbulence energy-spectrum function [Hinze (1975)].

It should be noted that the experimental technique of Martinez-Bazan, et al. (1999a, b) had a minimum measurable bubble size of 83 μm, which Martinez-Bazan states did not affect their breakup frequency results.

A critical bubble diameter $D_c$ exists and if $D \le D_c$ the bubbles will never breakup:

$$D_c = \left(\frac{12\sigma}{\beta\rho}\right)^{3/5} \varepsilon^{-2/5} \quad (43)$$

A minimum diameter exists below which there is insufficient turbulence induced stress to result in bubble breakup.

$$D_{min} = \left(\frac{12\sigma}{\beta\rho D}\right)^{3/2} \varepsilon^{-1} \quad (44)$$

The breakup frequency (1/s) is given by:

$$g(\varepsilon, D) = K_g \frac{\sqrt{\beta(\varepsilon D)^{2/3} - 12\sigma/(\rho D)}}{D} \quad (45)$$

where $\beta = 8.2$ [Batchelor (1956)] and $K_g = 0.25$ was determined experimentally by Martinez-Bazan, et al. (1999a).

The Martinez-Bazan, et al. (1999a) breakup model was developed for conditions more representative of MBDR flows than the Lehr-Mewes model, thus the former has been utilized in the present work.

**Kumar-Ramkrishna Partitioning-Breakup**

The Kumar-Ramkrishna particle bin size representation is shown schematically in Figure 4. Here $x_i$ is the representative bubble bin volume (e.g. average or mid-point volume) due to breakup of mother with volume $x_k$

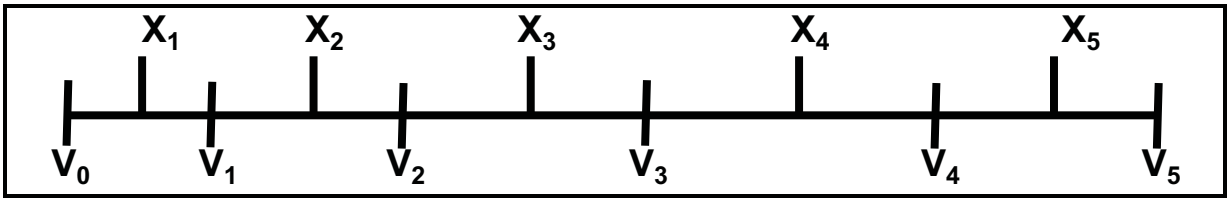



**Figure 4. Kumar-Ramkrishna particle bin size representation.**

Conservation leads to the following relation with the breakup rate is written as:

$$R_{Bb} = \sum_{k=1}^{N_B} n_{i,k} \Gamma_B(x_k) N_k(t) \quad (46)$$

where $N_k$ is the total number of particles in bin "k" and $\Gamma_B(x_k)$ is the breakup frequency (kernel) for mother particle $x_k$ and $n_{i,k}$ is the contribution to the population of the "*i*-th" bin size ($x_i$) due to breakup of particle $x_k$.

$$n_{i,k} = \frac{B^{\mu}_{i,k} x^{\nu}_{i+1} - B^{\nu}_{i,k} x^{\mu}_{i+1}}{x^{\mu}_{i} x^{\nu}_{i+1} - x^{\nu}_{i} x^{\mu}_{i+1}} + \frac{B^{\mu}_{i-1,k} x^{\nu}_{i-1} - B^{\nu}_{i-1,k} x^{\mu}_{i-1}}{x^{\mu}_{i} x^{\nu}_{i-1} - x^{\nu}_{i} x^{\mu}_{i-1}} \quad (47)$$

$$B^{\xi}_{i,k} \equiv \int_{x_i}^{x_{i+1}} v^{\xi} \beta(v, x_k) dv \quad (48)$$

The death rate due to breakup of a particle of size $x_k$ is:

$$R_{Db} = \Gamma(x_k) N_k(t) \quad (49)$$

**Kumar-Ramkrishna Partitioning-Coalescence**

The coalescence formulation is simpler than that for breakup. The birth rate of particles due to the coalescence of particles in bins j and k is given by:

$$R_{Cb} = \sum_{\substack{j,k \\ x_{i-1} \le (x_j + x_k) \le x_{i+1}}}^{j \ge k} \left(1 - 0.5\delta_{j,k}\right) \eta_{j,k} \Gamma^{i,k} N_j N_k \quad (50)$$

where the distribution function due to the coalescence ($\eta_{j,k}$) is given by:

$$\eta_{j,k} = \frac{v^{\mu} x^{\nu}_{i+1} - v^{\nu} x^{\mu}_{i+1}}{x^{\mu}_{i} x^{\nu}_{i+1} - x^{\nu}_{i} x^{\mu}_{i+1}}, \quad x_i \le v \le x_{i+1}$$

$$\eta_{j,k} = \frac{v^{\mu} x^{\nu}_{i-1} - v^{\nu} x^{\mu}_{i-1}}{x^{\mu}_{i} x^{\nu}_{i-1} - x^{\nu}_{i} x^{\mu}_{i-1}}, \quad x_{i-1} \le v \le x_i$$

where $v = x_j + x_k$ (51)

where $v=x_j+x_k$, and $\Gamma^{i,k}$ is the coalescence rate (kernel) due to the coalescence of particles in bins j and k.

The death rate of particles due to the coalescence of particles in bins j and k is given by:

$$R_{Cd} = N_i \sum_{k=1}^{N_B} \Gamma^{i,k} N_k \quad (52)$$

*Breakup and Coalescence – Interfacial Area Density Transport Formulation*

An approximate formulation including bubble breakup and coalescence within the interfacial area framework was proposed by Lehr and Mewes (2001). Lehr and Mewes solved the population balance equation "to describe the evolution of bubble sizes in two-phase flow." To reduce the numerical complexity due to a large number of equations and

strong coupling, they formulated an equation for average bubble volume (equivalent to the interfacial area transport equation) using an approximate analytical approach. A summary of the Lehr-Mewes approach follows. Source terms in the population balance equation involve breakup and coalescence kernel functions that are a function of the bubble volume, v. By assuming that an arithmetically averaged bubble volume ( $\bar{v}$ ) may be used in the kernel functions, a simplified solution for the bubble number-density distribution function, $f(v)$, results:

$$f(v) = \frac{\alpha_g}{\bar{v}^2} \exp\left(-\frac{v}{\bar{v}}\right) \quad (53)$$

$$n_B = \int_0^\infty f(v')dv' = \frac{\alpha_g}{\bar{v}} \quad (54)$$

Lehr and Mewes obtained a transport equation for average bubble volume with simplified source terms due to breakup and coalescence (equivalent to the source terms $\Phi_B$ and $\Phi_C$ and in equation (7). The bubble number-density PDF implies a bubble size distribution consistent with the above noted assumptions. We use this PDF to evaluate bubble number densities for discrete "bins." The bins are defined as shown in Figure 4.

Here $x_i$ is the representative bubble bin volume (e.g. average or mid-point volume) of bin "i" and $v_{i-1}$ and $v_i$ are the lower and upper bin volumes of bin "i", respectively. The number density PDF of bubbles in bin "i" is then

$$N_{B(i)} = \frac{\alpha_g}{\bar{v}}\left(e^{-v_{i-1}/\bar{v}} - e^{-v_i/\bar{v}}\right) \quad (55)$$

This result approaches the number density PDF for a sufficiently large number of bins and a sufficiently large maximum bin volume. Also the first bin is assumed to contain all bubbles from zero bin volume to the uppermost volume of this bin (i.e. $v_1 = 0$). Further to prevent errors due to an insufficiently "large" maximum volume, the distribution must normalized such that $\sum_{\text{all bins}} N_{B(i)} = 1$.

As in the N-bin formulation, we use the Prince and Blanch (1990) rate kernel for coalescence and the Martinez, et al (1999a, b) rate kernel and daughter size distribution for breakup. A complete description of these models is included above; only the essentials are summarized here under the assumption that the rates may be evaluated using the mean bubble diameter.

$$\phi_C = n_B^2\left(\frac{\pi}{2}\right)\varepsilon^{1/3}\bar{D}^{7/3}\exp\left(-t_B/\tau_B\right) \quad (56)$$

where $n_B$ is the bubble number density, $\varepsilon$ is the turbulence energy dissipation rate, $t_B$ is the time required for two bubbles of diameter $\bar{D}$ to coalesce and $\tau_B$ is the contact time for the two bubbles. In the interfacial area density formulation, the bubble number density is given by

$$n_B = \frac{a_i}{\pi \overline{D}^2} \tag{57}$$

As in the prior section the time required for two bubbles to coalesce is given by:

$$t_B = \left( \frac{\left(0.5\overline{D}\right)^3 \rho^{liq}}{16\sigma} \right)^{1/2} \ln\left(\frac{h_0}{h_f}\right) \tag{58}$$

where $h_o$ is an initial film thickness between two bubbles as they just come into contact and $h_f$ is a final critical film thickness where rupture occurs and the bubbles coalesce. For air-water systems, the film thickness values quoted by Prince and Blanch (from other sources) are

The contact time for bubbles in turbulent flow [Levich (1962)] with a modification due to the relative velocity between the bubbles [Carrica, et al. (1999)], is given by:

$$\tau_B = \frac{D_{ch}}{U_{B,rel} + 2\left(0.5 D_{ch}\varepsilon\right)^{1/3}} \tag{59}$$

where, as before, $D_{ch}$ is a characteristic length related to the bubble sizes and $u_{B,rel}$ is the mean relative velocity between the colliding bubbles. The characteristic length ( $D_{ch}$ ) is taken as $D_{ch} = \overline{D}$ .

*Enthalpy Transport*

For compressible flows and flows with heat transfer, it is necessary to solve for an energy equation. NPHASE-PSU incorporates an enthalpy transport equation for each field:

$$\frac{\partial}{\partial t}\left(\alpha^k \rho^k h^k\right) + \frac{\partial}{\partial x_j}\left(\alpha^k \rho^k u_j^k h^k\right) = \alpha^k \frac{D^k p}{Dt} + \frac{\partial}{\partial x_j}\left[\alpha^k \left(\mu^k + \frac{\mu_t^k}{\sigma_h^k}\right)\frac{\partial h^k}{\partial x_j}\right] + \tau_{ij}^k \frac{\partial u_i^k}{\partial x_j} \tag{60}$$

An available alternative that is better suited for higher Mach number compressible flows (transonic, supersonic) in the context of the segregated solution strategy employed in NPHASE-PSU is stagnation enthalpy transport:

$$\frac{\partial}{\partial t}\left(\alpha^k \rho^k H^k\right) + \frac{\partial}{\partial x_j}\left(\alpha^k \rho^k u_j^k H^k\right) = \alpha^k \frac{\partial p}{\partial t} + \frac{\partial}{\partial x_j}\left[\alpha^k \left(\mu^k + \frac{\mu_t^k}{\sigma_h^k}\right)\frac{\partial h^k}{\partial x_j}\right] + \frac{\partial}{\partial x_j}\left(\tau_{ij}^k u_i^k\right) \tag{61}$$

*Turbulence Model*

NPHASE-PSU has a number of low and high Reynolds number form 1-equation turbulence models (Spalart-Allmaras), 2-equation turbulence models (k-ε, q-ω, k-ω, k-R) and a low Reynolds number 4-equation v2f model [Durbin, (1991)]. In the context of multifield flows, separate turbulence transport scalars are solved for each field. For example, the high Reynolds number k-ε model is written:

$$\frac{\partial}{\partial t}\left(\alpha^k\rho^k k^k\right)+\frac{\partial}{\partial x_j}\left(\alpha^k\rho^k u_j^k k^k\right)=\frac{\partial}{\partial x_j}\left[\alpha^k\left(\mu^k+\frac{\mu_t^k}{\sigma_k^k}\right)\frac{\partial k^k}{\partial x_j}\right]+P^k-\alpha^k\rho^k\varepsilon^k+S_k^k$$

$$\frac{\partial}{\partial t}\left(\alpha^k\rho^k \varepsilon^k\right)+\frac{\partial}{\partial x_j}\left(\alpha^k\rho^k u_j^k \varepsilon^k\right)=\frac{\partial}{\partial x_j}\left[\alpha^k\left(\mu^k+\frac{\mu_t^k}{\sigma_\varepsilon^k}\right)\frac{\partial \varepsilon^k}{\partial x_j}\right]+C_1\frac{\varepsilon^k}{k^k}P^k-C_2\frac{\varepsilon^k}{k^k}\alpha^k\rho^k\varepsilon^k+S_\varepsilon^k \quad (62)$$

In equation (62), all field indicator superscripts are eliminated if only the liquid field is solved. $S_k$ and $S_\varepsilon$ are available source/sink terms to: extract turbulence energy associated with breakup [Meng and Uhlman (1998), Kunz et al. (2003)], and modified production due to interface dynamics and mass transfer mechanisms proposed by various authors [Ferrante and Elghobashi (2004, 2005), Tryggvason and Lu (2005)].

## Numerics/Code

For single phase flow, the present algorithm follows established segregated pressure based methodology. A colocated variable arrangement is used and a lagged coefficient linearization is applied [Clift and Forsyth (1994), for example]. One of several diagonal dominance preserving, finite volume spatial discretization schemes is selected for the momentum and turbulence transport equations. Continuity is introduced through a pressure correction equation, based on the SIMPLE-C algorithm [Van Doormal and Raithby (1984)]. In constructing cell face fluxes, a momentum interpolation scheme [Rhie and Chow (1983)] is employed which introduces damping in the continuity equation. At each iteration, the discrete momentum equations are solved approximately, followed by a more exact solution of the pressure correction equation. Turbulence scalar and volume fraction equations are then solved in succession. As discussed above, several important numerical issues arise in two-fluid CFD, foremost among these, that sufficient implicit coupling between the constituents be established. In the present work this is accomplished using the Coupled Phasic Exchange (CPE) algorithm [Kunz et al. (1998)]. In NPHASE-PSU, CPE has been extended to a fully unstructured, parallel, time accurate scheme employing higher-order discretization practices. Details of the data structure, discretization, and CPE elements of the scheme are summarized in this section.

### *Data Structure*

The hierarchal data structure employed is illustrated in Figure 5. The cell-centered finite volume flow solver accepts arbitrary polyhedral elements. The data structure is face based, that is, subsequent to the assembly of geometric parameters in the front end, all inter-element connectivity is retained in face pointers to the two adjacent cells. The fundamental data structure member is the “fedge” (face edge) which points to its two vertices and faces. Each face points to its bounding elements. This data structure provides a convenient framework for assembly of all required geometric parameters.

fedges and faces are identified as either internal or boundary. A “boundary_patch” structure in the C flow solver includes as members a number of attributes for boundary faces including areas and other geometric information, scalar values at the face center, fluxes, and inter-partition boundary data storage and transfer buffers.

*Discretization*

The governing equations are discretized using a cell centered finite volume method applied to arbitrary polyhedral cell types. Inviscid and viscous fluxes are accumulated by sweeping through internal and boundary faces. For inviscid flux evaluation:

$$\int_{\underline{A}} \alpha^k \rho^k \phi^k \underline{V}^k \bullet d\underline{A} = \sum_f C_f^k \phi_f^k \qquad (63)$$

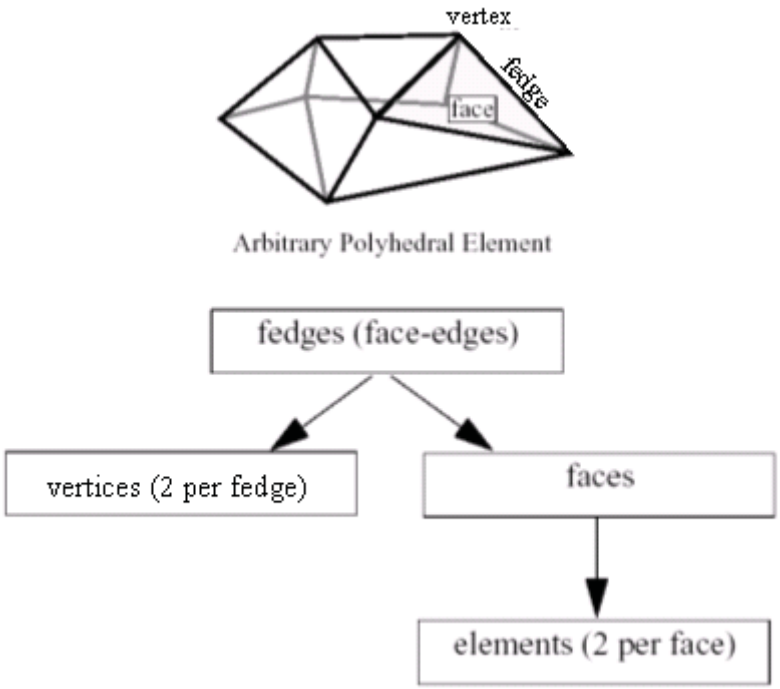



**Figure 5. Heirarchal data structure in NPHASE-PSU**

where $C_f^k$ is face mass flux for field k, and $\phi_f^k$ is the value of general transport scalar $\phi^k$ evaluated at face, f. The summation is taken over all faces bounding the element. $C_f^k$ is evaluated based on field variables available prior to the solution of the transport equation for $\phi^k$ (lagged coefficient linearization). Second order accuracy is obtained by evaluating $C_f^k$ using a central plus 4th difference pressure artificial dissipation term due to Rhie and Chow (1983) : C

$$C_f^k = \rho_f^k \alpha_f^k \overline{\underline{V}}^k \bullet \underline{A}^f + \rho_f^k \alpha_f^k \left[ \overline{B}^k \left( \overline{\nabla p} \bullet \underline{A}^f - \Delta p \left| \underline{A}^f \right|^2 \right) \right] + \rho_f^k \alpha_f^k \left[ \overline{F}^k \left( \overline{\nabla \alpha} \bullet \underline{A}^f - \Delta \alpha \left| \underline{A}^f \right|^2 \right) \right] \qquad (64)$$

and by evaluating $\phi_f^k$ from Lien (2000):

$$\phi_f^k = \phi_U^k + \left(\nabla\phi^k \bullet d\underline{r}\right)_U \qquad (65)$$

In equation (64), the overbar denotes a geometrically weighted mean at the face, i.e., referring to Figure 6. :

$$\overline{\nabla p} = (1-s)(\nabla p_1) + s(\nabla p_2)$$
$$s \equiv \delta s_1 / (\delta s_1 + \delta s_2) \qquad (66)$$

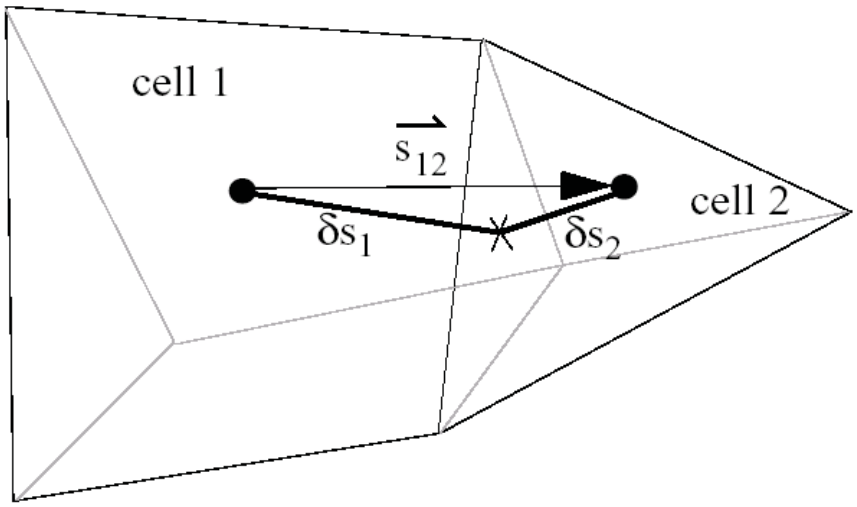



**Figure 6. Geometry nomenclature for cell face evaluations.**

and Δ designates a difference across the face (i.e., $\Delta p \equiv p_2 - p_1$). In equation (65) subscript U designates the quantity associated with the element upwind of face, f (which can vary with field), and $d\bar{r}$ is the vector from the upwind cell center to the face center. In Figure 7, results of a two-dimensional inviscid parallel stream test case are presented using a square mesh on a square domain aligned 45° skew to the flow direction and also using a triangular mesh. Inflow axial velocities are specified as =2 along the upper half inlet and =1 on the lower half. On both meshes, the significant interface smearing associated with first order upwinding is significantly reduced using the second order expression in equation (65). As detailed in Kunz et al. (1998) , dissipation parameters, $\overline{B^k}$ and $\overline{F^k}$ in equation (64) are scaled in a fashion that accommodates interfacial drag, mass transfer and dispersion forces:

$$B^k \equiv \sum_{l=1,nfield} \alpha^l \left(\underline{\underline{P}}^{-1}\right)^{kl}, F^k \equiv K \sum_{l=1,nfield} \left(\underline{\underline{P}}^{-1}\right)^{kl} \qquad (67)$$

where **P** is the NP x NP point coefficient matrix for the momentum equations defined below, which incorporates drag and mass transfer. $F^k$ is consistent with a widely used class of dispersive interfacial forces, $M_i^{kl} = K\nabla\alpha^l$ [e.g., Lopez DeBertodano (1998)].

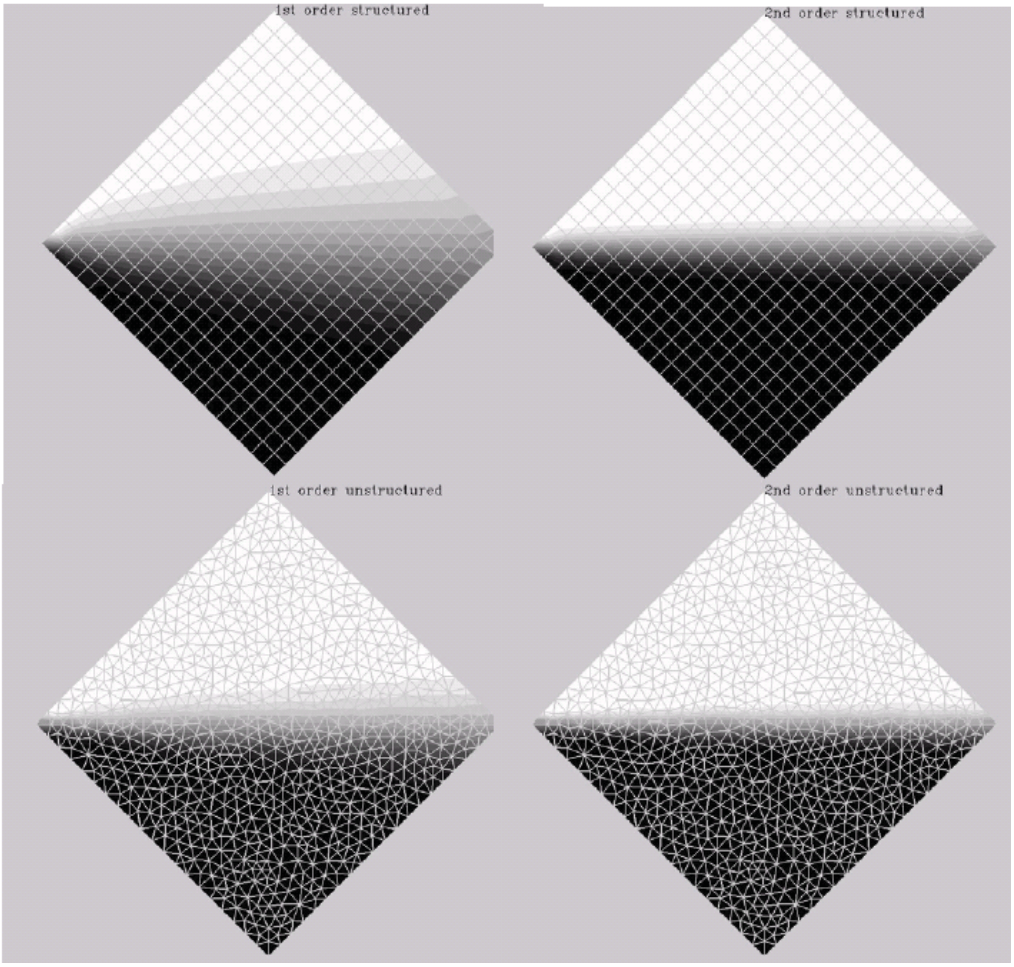



**Figure 7. Comparison of first and second order convection discretization for an inviscid "mixing" layer. Flow is left to right. Axial velocity contours, white: $\vec{V} = 2$, black: $\vec{V} = 1$.**

The evaluation of $B^k$ and $F^k$ in equation (67), requires the inversion of a rank NP matrix **P** at each grid point, at each iteration. This potentially CPU intensive procedure is circumvented by applying a simple Jacobi fixed point iterative procedure to approximately invert **P**. This procedure is rapidly convergent (2 sweeps are employed) since the **P** matrices are very well conditioned as discussed below.

Neglecting cross-diffusion and dilatation, the viscous flux in the momentum equations can be written for an element face as:

$$\int\left(\underline{\underline{\tau}}\bullet d\underline{A}\right), \underline{\underline{\tau}} = \alpha\mu\left(\nabla\underline{V}\right) \tag{68}$$

Referring to Figure 6, the gradient of a scalar, $\phi$, on the face can be written as:

$$\nabla\phi = \underbrace{\nabla\phi - \left(\nabla\phi\bullet\frac{\underline{S}_{12}}{|\underline{S}_{12}|}\right)\hat{\varepsilon}_{s12}}_{\mathbf{A}} + \underbrace{\left(\nabla\phi\bullet\frac{\underline{S}_{12}}{|\underline{S}_{12}|}\right)\hat{\varepsilon}_{s12}}_{\mathbf{B}} \tag{69}$$

The terms labelled A represent components of the gradient that are orthogonal to $\bar{s}_{12}$. These terms are generally small (for hexahedral or prismatic elements extruded from geometric surfaces, neglecting them is nearly equivalent to the thin-layer assumption). Their discrete form is treated explicitly in the solution of the momentum equations (term $S^k$ in equation (76) below). The terms labelled B represent components of the gradient that are parallel to $\bar{s}_{12}$. These are discretized as:

$$\int_f (\alpha\mu \nabla \underline{V} \bullet d\underline{A}) = \overline{(\alpha\mu)}_f \left( \nabla \underline{V} \bullet \frac{d\underline{s}}{|d\underline{s}|} \right) (\hat{\varepsilon}_s \bullet d\underline{A}) = \overline{(\alpha\mu)}_f \frac{(\underline{V}_2 - \underline{V}_1)(d\underline{s} \bullet d\underline{A})}{|d\underline{s}|^2} \quad (70)$$

and are treated implicitly (terms $A_P^k$ and $A_{nb}^k$ in equation (76) below).

Gradients that appear in the flux calculations, and elsewhere, are computed using Gauss' Law:

$$\nabla\phi = \frac{1}{\forall} \sum_f \underline{A}_f \bar{\phi}_f^k \quad (71)$$

with internal face values of $\bar{\phi}_f^k$ computed from equation (66), and the summation take over all faces bounding an element. Equation (71) is computed by sweeping all internal and boundary faces, accumulating adjacent element contributions to $V$ and $\bar{A}_f \phi_f^k$ from the face.

*Interfacial Force Evaluation*

In order to discuss interfacial force discretization issues, we consider three classes of these terms. First, when cast as in equation (10), drag can be viewed as a scalar sink term. That is to say, drag term, $D^{kl}$, which appears in the momentum equations as:

$$\sum_{k \neq l} D^{kl} \left( u_i^l - u_i^k \right) \quad (72)$$

is generally evaluated for each element, and by virtue of its relative velocity factor, incorporated implicitly in the NPx NP block diagonal, **P**, of the momentum equation coefficient matrix, as seen in equation (80) below. As discussed above, the appearance of drag in this term is accommodated consistently in the Rhie-Chow scale factors in equation (67). It is noted that numerically, mass transfer plays a very similar role to drag (though $D^{kl} = D^{lk}$ and in general $\Gamma^{kl} \neq \Gamma^{lk}$), and accordingly its treatment is consistent with that discussed for drag.

The second class of interfacial force terms are those that are linear in the gradient of volume fraction. These are generally dispersive in nature. These terms are evaluated straightforwardly using model equations such as (18), however, as is demonstrated in Kunz et al. (1998), including the Rhie-Chow-like term, $\rho_f^k \alpha_f^k \left[ \bar{F}^k \left( \overline{\nabla\alpha} \bullet \underline{A}^f - \Delta\alpha \left| \underline{A}^f \right|^2 \right) \right]$, in equation (67) is critical for obtaining convergent oscillation free solutions when such forces are present.

The third class of forces are simply those that do not conform to drag-like or dispersive-like forms. An example is lift, a particular form of which is taken here from Lahey and Drew (2000):

$$\underline{M}_{LIFT}^{c-d} = \alpha^d \rho^c C_L \underline{V}^{rel} \times \nabla \times \underline{V}^c \quad (73)$$

where superscripts c and d refer to continuous and disperse fields respectively.

A straightforward discretization of equation (73) in an element centered (or variable colocated) scheme such as presented here, would involve evaluating gradients using equation (71) and multiplying by appropriate velocity, volume fraction and density factors using element values. In Kunz and Venkateswaran (2000) it was demonstrated that such an approach can also lead to solution oscillations and attendant convergence degradation. There it was observed that staggered grid methods (i.e., those where the momentum equations are evaluated at locations staggered to the element centers) do not exhibit this behavior for this class of force. Accordingly, a staggered force discretization was proposed wherein the force in equation (73) is evaluated at each cell-face. Face values are then averaged to obtain element values. This force distribution renders staggered and colocated forms identical for linear forces and thereby removes solution oscillations. In the present unstructured framework this "distribution" of force across several nodes can be written:

$$\overline{M} = \frac{\sum_f M_f \forall_f}{\forall} \quad (74)$$

where $M_f$ represents the force averaged to the face per equation (73), $\forall$ is the element volume and $\forall_f$ is the volume formed by face f and the segments connecting the face vertices to the volume centroid. It was observed in Kunz and Venkateswaran (2000) that this approach is equivalent to the addition of a second difference artificial dissipation to the standard colocated discretization, i.e., in the present unstructured context:

$$\overline{M} = M + \nabla \bullet K \nabla M \quad (75)$$

where scaling factor, K, has dimensions of length$^2$.

*Boundary Conditions*

A palette of boundary conditions are available in the code including walls, symmetry boundaries, inlets (transport scalars specified, pressure extrapolated from the domain interior), pressure boundaries (transport scalars extrapolated from the domain interior, pressure specified), and cyclic boundaries (for turbomachinery analysis). All boundary conditions are treated implicitly in the formation of influence coefficients for the transport scalars. For scintered metal plate injection, porous wall boundary conditions are used, where an area permeability, $\lambda$, is specified. Shear force on porous boundary faces is apportioned as $F = \tau_w A_f (1 - \lambda)$, where $A_f$ is the face area and $A_f \lambda$ is the area available for injection flux.

*Implicit Solution Procedure*

Invoking a dual-time formulation, the discretized governing equations for transport scalar, $\phi^k$, can be written in $\Delta$-form as:

$$\left[A_P^k + \sum_{k\neq l} b^{kl} + \frac{\rho^k \alpha^k \forall}{\Delta\tau} + \frac{3\rho^k \alpha^k \forall}{2\Delta\tau}\right]\Delta\phi_P^k$$
$$-\sum_{k\neq l} b^{lk}\Delta\phi_P^l - \sum_{nb} A_{nb}^k \Delta\phi_{nb}^k = \left[\sum_{nb} A_{nb}^k \left(\phi_{nb}^k\right)^{n+1,m} -\right.$$
$$\left.\left(A_P^k + \sum_{k\neq l} b^{kl}\right)\left(\phi_P^k\right)^{n+1,m} - \sum_{k\neq l} b^{lk}\left(\phi_P^l\right)^{n+1,m}\right]$$
$$+S^k - \left[\frac{3\rho^k \alpha^k \forall}{2\Delta t}\left(\phi_P^k\right)^{n+1} - \frac{2\rho^k \alpha^k \forall}{\Delta t}\left(\phi_P^k\right)^{n} + \frac{\rho^k \alpha^k \forall}{2\Delta t}\left(\phi_P^k\right)^{n-1}\right] \quad (76)$$

where $\Delta\phi^k \equiv \left(\phi^k\right)^{n+1,m+1} - \left(\phi^k\right)^{n+1,m}$, and $b^{kl}$ represents the accumulated drag and mass transfer terms (i.e., for the momentum equations, $b^{kl} = D^{kl} + \Gamma^{kl}$)

In equation (76), second order backward differencing has been used for the physical time derivative ($\Delta t$) and Euler implicit differencing is employed for the pseudo-time derivative ($\Delta\tau$). A standard under-relaxation procedure is employed where an appropriate underrelaxation factor, $\omega$ is selected ($0.3 \le \omega \le 0.7$) and the pseudo-timestep is evaluated from:

$$\Delta\tau \equiv \frac{\omega}{1-\omega}\left(\frac{\rho^k \alpha^k \forall}{A_P^k + \sum_{k\neq l} b^{kl}}\right) \quad (77)$$

It has been observed in Venkateswaran et al. (1997) that such a specification is equivalent to a local timestepping procedure that accommodates CFL and VonNeuman stability. For physical transients, pseudo-timesteps correspond to sub-iterations of the SIMPLE-C algorithm.

*Phase Coupled Scalar Linear Solution Strategy*

Equation (76) represents a coupled system of NP equations for the NP unknowns $\phi^k$:

$$\underline{\underline{A}}\,\underline{\phi} = \underline{S} \quad (78)$$

where coefficient matrix **A** has the form:

$$\underline{\underline{A}} = \begin{bmatrix} \underline{\underline{P}} & & & \\ & \underline{\underline{P}} & \underline{\underline{U}} & \\ & \underline{\underline{P}} & \ddots & \\ & \ddots & \ddots & \underline{\underline{P}} \\ \underline{\underline{L}} & & \underline{\underline{P}} & \underline{\underline{P}} \end{bmatrix} \tag{79}$$

with

$$\underline{\underline{P}} = \begin{bmatrix} \dfrac{A_P^1 + \sum_{k\neq 1}^{NP} b^{1k}}{\omega} & -b^{21} & \cdots & -b^{NP1} \\ -b^{12} & \dfrac{A_P^2 + \sum_{k\neq 2}^{NP} b^{2k}}{\omega} & \cdots & -b^{NP2} \\ \vdots & \vdots & \ddots & \vdots \\ -b^{1NP} & -b^{2NP} & \cdots & \dfrac{A_P^{NP} + \sum_{k\neq NP}^{NP} b^{NPk}}{\omega} \end{bmatrix} \tag{80}$$

and upper and lower block triangular matrices **U** and **L** containing neighbor cell influence coefficients.

For the diagonal dominance preserving discretizations employed, conventional iterative schemes will have diagonally dominant iteration matrices with spectral radii less than or equal to the underrelaxation factor, ω [Kunz et al. (1998)], a direct consequence of the well conditioned nature of the main diagonal block matrix **P**. Accordingly, we consistently employ a simple point Jacobi scheme for solving equation (76) for all scalars ($u_i$, α, k, ε), as this scheme is guaranteed to provide adequate convergence within several sweeps. For the momentum equations, all three velocity components are solved for all fields simultaneously using point Jacobi iteration.

As discussed above, the well conditioned nature of **P** renders determination of dissipation parameters $B^k$ and $F^k$, in equation (67), (which scale with $\mathbf{P}^{-1}$) amenable to a simple point Jacobi iteration as well.

*Continuity Equation Linear Solution Strategy*

In the present work a mixture volume conservation equation is derived by summing individual field volume fraction equations, each normalized by field density. A SIMPLE-C [Van Doormal and Raithby, (1984)] based pressure-velocity corrector relation (which

accommodates the same level of interfield coupling as the artificial dissipation operators discussed above, Kunz et al. [1998]) is applied to develop an elliptic pressure correction equation. Transport equations for the field volume fraction equations are then solved. In this fairly standard method, under-relaxation is not employed for the pressure corrector equation in order to achieve a measure of mixture volume conservation at each pseudo-timestep. As as result, the discrete pressure corrector equation system is symmetric positive semi-definite ( $A_P = \sum_{nb} A_{nb}$ ) and thereby its linear solution is a challenging and important factor in the nonlinear convergence rate of the overall scheme. In NPHASE-PSU, the PETSC suite of solvers are employed for the solution of this system. Depending on the degree to which mass conservation needs to be satisfied at a given non-linear iteration, a GMRES solver or a more CPU intensive Algebraic Multigrid procedure are invoked from the PETSC library of solvers. Details of these solvers are provided in (PETSC [2006]).

*Parallelization*

The code is parallelized based on domain decomposition using MPI. Partitioning is carried out in the pre-processor, fump, as described in the user's Manual below. Inter-partition boundaries are input to the flow code from fump as any other boundary condition with a single additional boundary patch attribute being the neighbor partition processor number. fump writes inter-partition face pointers to the NPHASE-PSU input files (unphase.gridxxx) in the same order that these faces are encountered in fump. Accordingly no reordering is required when loading and unloading 1-D structures associated with message passing. Data is passed after each scalar is computed in the segregated procedure. For the point iterative solvers used for the scalar equations, $\Delta\phi$ is passed at every sweep of the linear solver, so that there is no degradation in convergence due to domain decomposition. For the PETSC solvers used for the pressure corrector equation, the code is parallelized at the matrix level. Accordingly, a global matrix is assembled each non-linear iteration and global mass conservation is strictly enforced each timestep as the pressure solver is converged.

Further details on the physical models, numerics and code are available in Kunz et al. (1999, 2000, 2001, 2007, 2011).

## *Preprocessing*

### Overview

Figure 8 illustrates the front end for NPHASE-PSU. The code accepts a "light" faced-based grid specification. This achieves several goals. Firstly it enables the specification of arbitrary polyhedral meshes rather than being restricted to the four simplicial element types (tetrahedral, hexahedra, prisms and pyramids). Secondly, most commercial grid generators produce the face-based COBALT files now native to NPHASE-PSU (Gridgen, Pointwise, ICEM, HARPOON) or closely related face based formats (e.g., GAMBIT).

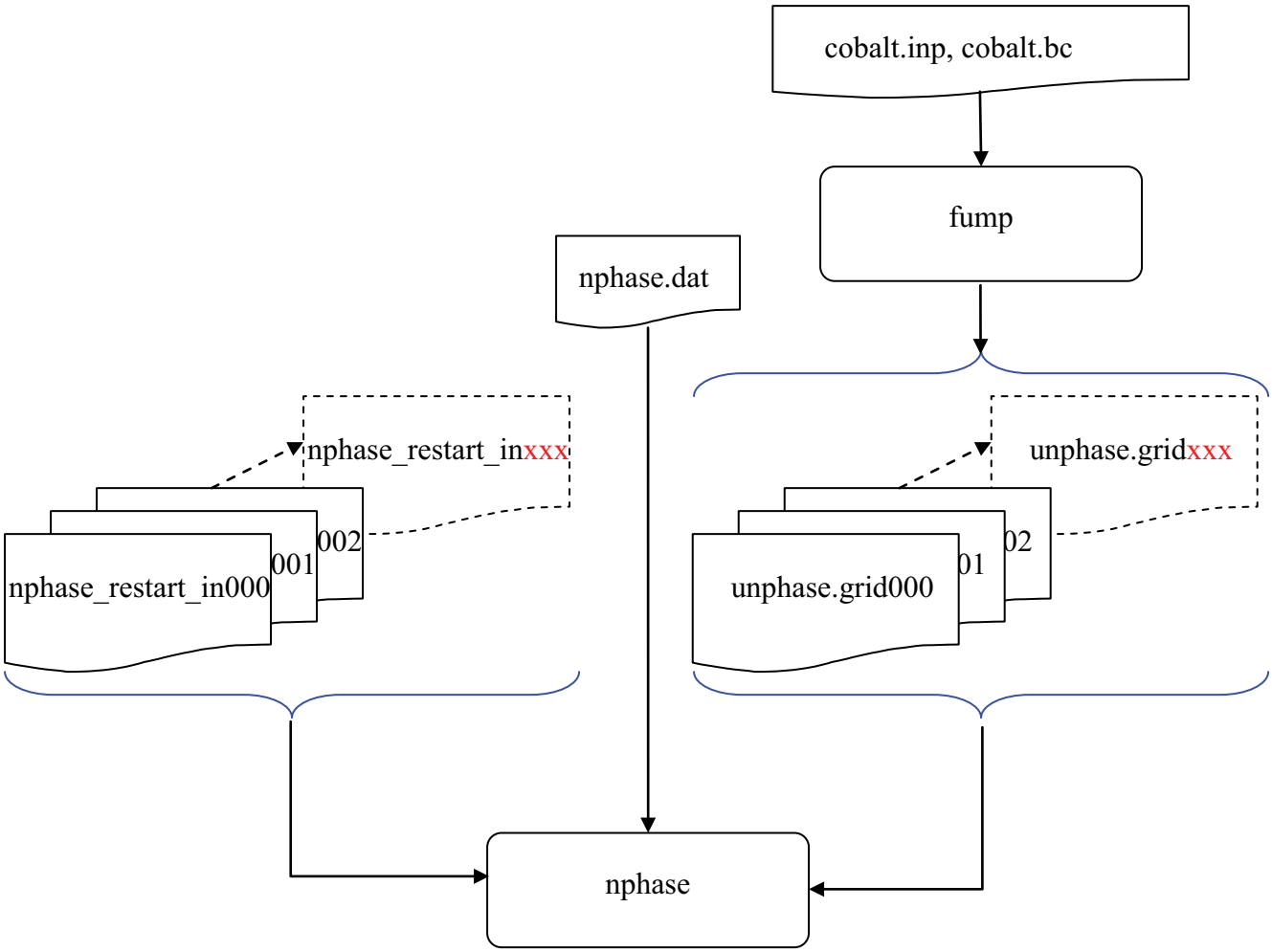



**Figure 8. Sketch of frontend for NPHASE-PSU**

## Input Files

Referring to Figure 8, there are three sets of input files to NPHASE-PSU. The first is the author file, nphase.dat, which is a simple key-word based ascii file that specifies all of the real and integer data defining execution control, boundary condition flags and attributes, fluid properties and initial conditions. A simple example nphase.dat file is included in Figure 9, and the "Control Commands" section below is devoted to a description of all of the key words available. The keywords (such as "number of fields") are case sensitive but blanks are ignored. Integer and real attributes can be written in free format. Lines are commented out by placing # in column 1.

```
#case title:
#simple nphase.dat file

iterations to perform 100

number of fields 1

time accurate simulation
temporal discretization momentum 1
physical timestep in seconds .1
number of physical timesteps 1
transient file write frequency 2

#initialize run with restart file

produce ensight output
restart file write frequency 100

dont perform wall match logic

inlet patch 1 0
1. 0. 0. 1.  1. 0.1 0.1 0. 0. 0 0

pressure patch 1 0
0. 1. 1. 0.1 0.1 0. 0.

turbulent flow high reynolds number k epsilon

constant fluid molecular viscosity 1.0e-2
constant fluid density 1.0

function entry/exit echo off

solversweepsforu 3
solversweepsforv 3
solversweepsforw 3
solversweepsfork 3
solversweepsfore 3

solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel

initialize u field   1.
initialize k field   .1
initialize e field   .1
```

**Figure 9. Sample nphase.dat file**

The second set of files are the grid files, written in COBALT unstructured format, cobalt.inp and cobalt.bc. The cobalt.inp file completely defines the input grid in faced based format. The file is written in COBALT format, as defined in Figure 10. The user need not concern himself with the content of this file although if a formatted cobalt.inp file is written by the grid generator, it will be human readable.

ndim nzones nbc
nvert nface ncell maxppf maxfpc
$x_1$ $y_1$ $z_1$
$x_2$ $y_2$ $z_2$
.
.
.
$x_{nvert}$ $y_{nvert}$ $z_{nvert}$
$nvf_1$ ($f\_v_1$(ivf),ivf=1,$nvf_1$) $f\_e1_1$ $f\_e2_1$
$nvf_2$ ($f\_v_2$(ivf),ivf=1,$nvf_2$) $f\_e1_2$ $f\_e2_2$
.
.
.
$nvf_{nface}$ ($f\_v_{nface}$(ivf),ivf=1,$nvf_{nface}$) $f\_e1_{nface}$ $f\_e2_{nface}$

**Figure 10. cobalt.inp grid file format**

In this file the parameters are defined as follows:

- ndim = # of dimensions ➔ always 3 for NPHASE-PSU, even if a 2D case is set up (see "Running Two-Dimensional Problems,' below)
- nzones = 1 (always for NPHASE-PSU)
- nbc = total number of boundary conditions that are defined in cobalt.inp and cobalt.bc
- nvert = number of vertices in model
- nface = number of vertices in model
- ncell = number of vertices in model
- maxppf = maximum number of vertices per face in any one face in domain
- maxfpc = maximum number of faces per element in any one element in domain
- $x_{ivert}$ $y_{ivert}$ $z_{ivert}$ = vertex coordinates for vertex ivert
- $nvf_{iface}$ = number of vertices on face iface
- $f\_v_{iface}$ = list of vertex numbers for face iface, listed in order around the periphery of the face such that the right-hand-rule applied to this ordering defines a direction from bounding element ) f_e1 to f_e2

- f_e1$_{iface}$ = element number on “low” side of face iface
- f_e2$_{iface}$ = element number on “high” side of face iface. If f_e2$_{iface}$ < 0 then iface is a boundary face and f_e2$_{iface}$ specifies the (negative of the) boundary identifier as defined in cobalt.bc.

The cobalt.bc file defines boundary condition names. This ASCII file is written in standard COBALT boundary condition file format, an example of which is shown in Figure 11. The user need not concern himself with the content of this file unless it is not written by the grid generator (e.g., HARPOON), in which case the user must build this file by hand to conform to the boundary condition numbering in the grid generator and thereby the cobalt.inp file.

```
##################################################
  Boundary Condition Specification File for:
  Gridgen grid exported : Thu May 18 10:51:30 2006
##################################################
11
Wall_00
Gridgen bc region: 11
Methods: User Created BC
User data supplied here - see COBALT doc!
##################################################
9
Inflow_00
Gridgen bc region: 9
Methods: User Created BC
User data supplied here - see COBALT doc!
##################################################
10
Pressure_00
Gridgen bc region: 10
Methods: User Created BC
User data supplied here - see COBALT doc!
##################################################
```

**Figure 11. cobalt.bc file format**

In this file three boundary conditions have been defined, Wall_00, Inflow_00 and Pressure_00. This file tells the pre-processor, fump, that Wall_00 faces in the cobalt.inp file will have identifier f_e2$_{iface}$ = - 11. Also, for Inflow_00, f_e2$_{iface}$ = -9, and for Pressure_00, f_e2$_{iface}$ = -10.

The third set of files is solution restart files. NPHASE-PSU always generates a restart file for each processor after execution completion (and optionally at intermediate iterations/timesteps as defined in “Control Commands” section below). These output files conform to the naming convention nphase_restart_out*xxx* where *xxx* is the 3-digit processor identifier (range from 000 to 999). Solution restarts are invoked by: 1) copying each of these output restart files to a corresponding input restart file (i.e., mv nphase_restart_out000 nphase_restart_in000) and then 2) activating the keyword “initialize run with restart file” in nphase.dat (i.e., uncomment by removing leading “#”)

**Boundary Condition Specification**

NPHASE-PSU supports no-slip wall, porous wall, pressure, inflow, symmetry, farfield and other specialized boundary conditions. The user needs to define boundary patch names that conform to what NPHASE-PSU is expecting. Specifically, NPHASE-PSU requires boundary patch names that conform to the following syntax: Boundarytype_boundarynumber, where Boundary type is either of these character strings: Wall, Porwall, Inflow, Symmetry, Farfield, Pressure. Boundarynumber is a 2-digit integer starting at 00. So for example there may be two inflows in a model with different attributes, these would be named by the user Inflow_00 and Inflow_01. These strings must appear in cobalt.bc. The grid generators GRIDGEN, Pointwise and ICEM automatically propagate these names into cobalt.bc upon grid output, provided the user names them in the grid generator. If HARPOON is used the user must define these in cobalt.bc.

Each boundary type has its own attributes, which are defined in nphase.dat as specified in the "Control Commands" section below.

**fump**

The pre-processor to NPHASE-PSU, fump, reads the cobalt.inp file (formatted or unformatted) and the cobalt.bc files as input and performs two tasks:

1) Executes domain decomposition by invoking METIS (2006). fump does this by first extracting graph information for METIS. Specifically, each element in the grid is designated as a "vertex" in the graph and each element with which it shares a face represents an "edge". The kmetis module is used to partition the graph into approximately equal sizes.
2) Builds the internal pointer connectivity (e.g., face➔element, edge➔ vertex, edge➔ face), boundary pointer connectivity and interprocessor communication information needed by NPHASE-PSU

fump is run interactively (or in a script) by simply typing the absolute path of the executable, fump, to a UNIX shell prompt. fump has four small user inputs: 1) # of processors to use for domain decomposition, 2) whether METIS is to be used or the user will be specifying a decomposition graph, 3) any scale factor the user may wish to apply to the grid, 4) whether or not a global vertex connectivity file is generated (specialized option – most applications set = 0).

The output of fump is a series of files, unphase.grid*xxx*, where where *xxx* is the 3-digit processor identifier (range from 000 to 999). Each of these files is read by the corresponding processor from the executing front end of NPHASE-PSU.

fump is written in ANSI-C and C++ and compiles (at least) under the gnu C compilers (gcc, g++). The executable is generated by invoking make from the FUMP directory delivered with the software. The METIS libraries that fump requires are delivered with the software.

### *Code Execution*

If a single processor job is required, one simply need type the name of the absolute path to the executable, nphase, into a UNIX shell prompt in the working directory.

NPHASE-PSU is instrumented with mpi for inter-processor communication. MPICH2 is used, so mpirun is executed behind the scenes by the invoked mpiexec wrapper and shown below. On most production UNIX cluster systems interactive invocation of mpirun and mpiexec are not allowed, rather, the user must build a submit script, and submit the job to a queueing system such as PBS.

A typical run script for execution of NPHASE-PSU on the gale cluster at Penn State ARL is included in the tutorial section below. The input files (nphase.dat, unphase.grid*xxx* and nphase_restart_in*xxx*) must be located in the working directory from where the executable is invoked. Output files (discussed in next section) are written to this directory as well.

### *Postprocessing*

Figure 12 illustrates the back end for NPHASE-PSU.

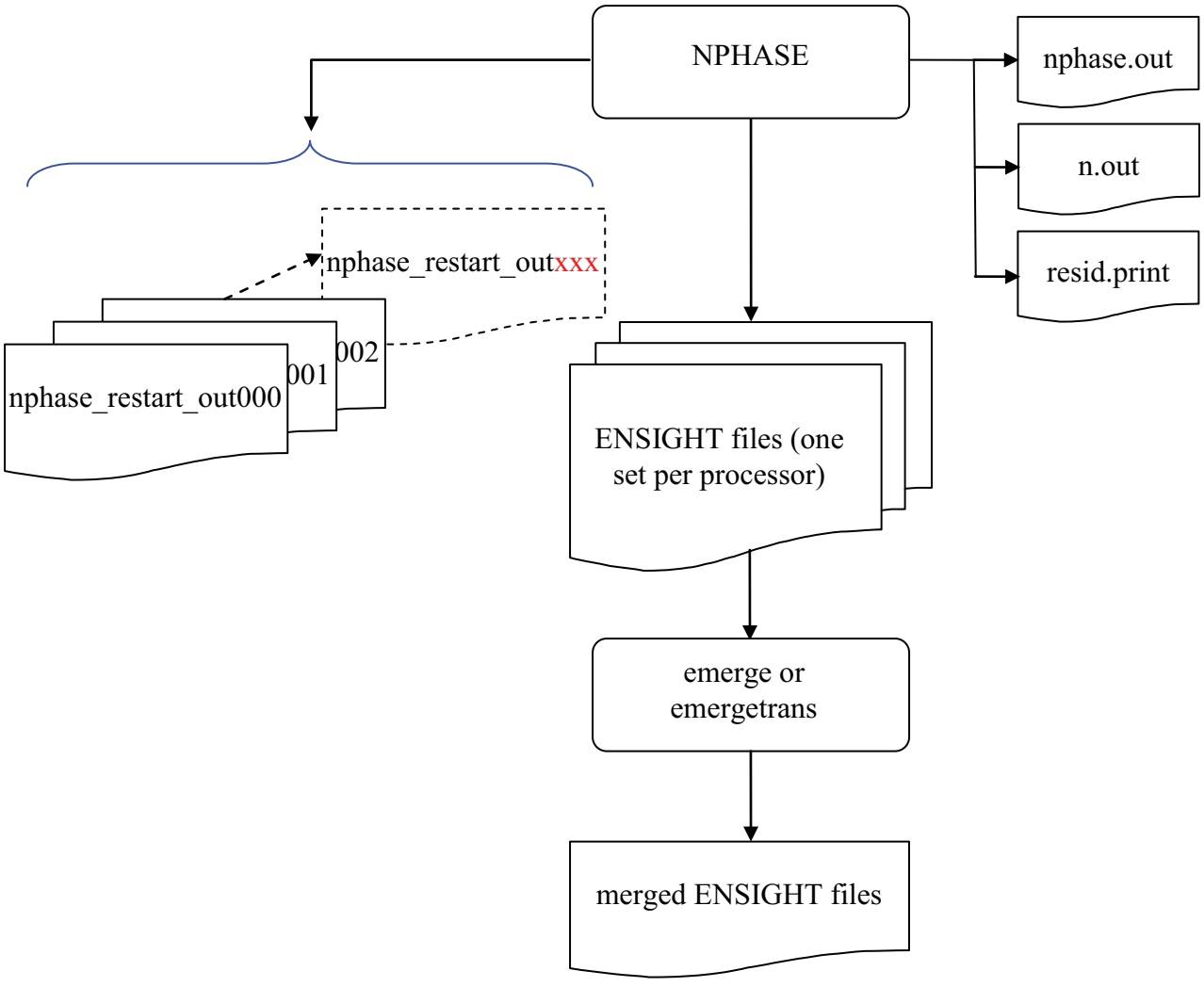



**Figure 12. Sketch of back end for NPHASE-PSU**

As with the front end, the backend now also supports arbitrary polyhedra. Specifically, ENSIGHT GOLD format is used to write the output files. ENSIGHT Versions 8.0 and later will read and display these files.

## Output Files

There are five classes of output files to NPHASE-PSU. The first is the standard ASCII printed output file, nphase.out, an example of which is included in Figure 13. This contains an echo of the input, residual history, and, if enabled in backend, printed field data for single processor jobs. (these print_nodal commands are commented out in backend.c since it is unlikely that a user would ever wish to obtain a printed output for an unstructured domain).

```
   NN      NN  PPPPPPPP   HH     HH  AAAAAAAAA  SSSSSSSSS  EEEEEEEEE
   NNN     NN  PP    PPP  HH     HH  AAAAAAAAA  SSSSSSSSS  EEEEEEEEE
   NNNN    NN  PP     PP  HH     HH  AA     AA  SS         EE
   NN NN   NN  PP    PPP  HH     HH  AA     AA  SS         EE
   NN  NN  NN  PPPPPPPP   HHHHHHHHH  AAAAAAAAA  SSSSSSSSS  EEEE
   NN   NN NN  PP         HH     HH  AA     AA         SS  EE
   NN    NNNN  PP         HH     HH  AA     AA         SS  EE
   NN     NNN  PP         HH     HH  AA     AA  SSSSSSSSS  EEEEEEEEE
   NN      NN  PP         HH     HH  AA     AA  SSSSSSSSS  EEEEEEEEE


  NPHASE: A COMPUTER PROGRAM FOR THE PREDICTION OF MULTIFIELD
          FLOWS WITH MASS, MOMENTUM AND ENERGY TRANSFER

   Developed by: Rob Kunz

 *** entered pre_author: reading nphase.dat input ***

 *** exiting pre_author: finished pass 1 on input file ***

 *** entered pre_author2: reading nphase.dat input ***

 *** exiting pre_author2: finished pass 1 on input file ***

 *** entered author: reading nphase.dat input file ***

iterations to perform 5
number of fields 1
time accurate simulation
temporal discretization momentum 1
physical timestep in seconds .1
number of physical timesteps 4
transient file write frequency 1
produce ensight output
dont perform wall match logic
inlet patch 1 0
1.0 0. 0. 1.  1. 0.1 0.1 0. 0. 0 0
pressure patch 1 0
0. 1. 1. 0.1 0.1 0. 0.
turbulent flow high reynolds number k epsilon
constant fluid molecular viscosity 1.e-3
constant fluid density 1
function entry/exit echo off
solver sweeps for u 3
solver sweeps for v 3
solver sweeps for w 3
solver sweeps for p 6
solver sweeps for k 3
solver sweeps for e 3
solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel
initialize u field   1.
initialize k field   .1
initialize e field   .1

 *** exiting author: finished 2nd pass on input file ***

  about to exit front end
     1  1 6.498e-02 0.000e+00 0.000e+00 1.515e+01 8.950e-03 5.938e-02 -1.337e+01
     2  1 1.168e-01 8.494e-02 9.137e-02 3.547e+00 2.788e-03 2.511e-02 -3.696e+00
     3  1 3.017e-02 2.215e-02 2.427e-02 1.142e+00 2.530e-03 1.011e-02 -1.147e+00
     4  1 1.281e-02 8.440e-03 9.386e-03 2.657e-01 1.725e-03 3.497e-03 -4.204e-01
     5  1 7.488e-03 3.947e-03 4.671e-03 1.383e-01 1.132e-03 1.649e-03 -2.265e-01
```

**Figure 13. Sample nphase.out file**

The second output file that is generated is standard output + standard error conventionally redirected to n.out, an example of which is included in Figure 13. This file contains a summary of the grid topology, front end progress, iteration history, user defined outputs, and summary flow rate and CPU performance information.

```
  12 processor run

 *** begin execution of nphase ***

unphase.grid files generated by fump version 2.0

 nvert=409798 nnode=388114 nfedge=4332994 nface=1102458
    inlet.nbcfedge=   1024,     inlet.nbcface=    256,     inlet.nbcfaceid=      1
partition.nbcfedge= 215690, partition.nbcface=  53192, partition.nbcfaceid=     12
 pressure.nbcfedge=   2080,  pressure.nbcface=    542,  pressure.nbcfaceid=      1
     wall.nbcfedge= 116882,      wall.nbcface=  32228,      wall.nbcfaceid=      5

finished reading grid in read_grid_unstruct

number of tets     in model                       =     20732
number of hexs     in model                       =    320105
number of prisms   in model                       =      5442
number of pyramids in model                       =     29983
number of non-simplicial volume elements in model =     11852
total mass of ifield =   0 in volume =  5.0867325450934e+04
solidflags: icount10 =         0
Number of internal solid faces = 0

  about to exit front end

************************
time accurate simulation

number of time steps to be run:     4
number of pseudotimesteps per physical timestep:     5
time step in seconds          :  1.00000000e-01
************************

about to  make subdirectory TRANSIN if it does not exist
done with make subdirectory TRANSIN if it does not exist

writing ensight output file(s) for current timestep
finished writing ensight output file(s) for current timestep

*********************************************
commencing inner iterations for t =  1.00000000e-01
                 iphystimeglobal =              1
*********************************************

  iter fld   ru        rv        rw        rp        rk        re        gmass     ws/cif
     1   1 6.498e-02 0.000e+00 0.000e+00 1.515e+01 8.950e-03 5.938e-02 -1.337e+01 3.659e-06
     2   1 1.168e-01 8.494e-02 9.137e-02 3.547e+00 2.788e-03 2.511e-02 -3.696e+00 3.435e-06
     3   1 3.017e-02 2.215e-02 2.427e-02 1.142e+00 2.530e-03 1.011e-02 -1.147e+00 3.455e-06
     4   1 1.281e-02 8.440e-03 9.386e-03 2.657e-01 1.725e-03 3.497e-03 -4.204e-01 3.453e-06
     5   1 7.488e-03 3.947e-03 4.671e-03 1.383e-01 1.132e-03 1.649e-03 -2.265e-01 3.455e-06

writing ensight output file(s) for current timestep
finished writing ensight output file(s) for current timestep
total mass of ifield =   0 in volume =  5.0867325450934e+04

*********************************************
commencing inner iterations for t =  2.00000000e-01
                 iphystimeglobal =              2
*********************************************

⇩
```

**Figure 14. Sample n.out file (part 1)**

```
⇩



*********************************************
commencing inner iterations for t =  4.00000000e-01
                iphystimeglobal =               4
*********************************************


  iter fld   ru        rv        rw        rp        rk        re        gmass     ws/cif
    16  1 6.074e-03 3.018e-03 3.006e-03 2.330e-01 3.567e-03 5.877e-03 -2.646e-01 3.485e-06
    17  1 1.721e-03 6.843e-04 6.456e-04 1.321e-01 1.260e-03 3.092e-03 -1.695e-01 3.470e-06
    18  1 1.002e-03 4.132e-04 3.987e-04 5.940e-02 6.373e-04 2.435e-03 -6.422e-02 3.438e-06
    19  1 5.080e-04 2.381e-04 2.078e-04 2.989e-02 3.783e-04 1.976e-03 -4.274e-02 3.469e-06
    20  1 3.505e-04 1.604e-04 1.606e-04 1.385e-02 2.450e-04 1.577e-03 -1.894e-02 3.429e-06
total mass of ifield =   0 in volume =  5.0867325450934e+04

writing output restart file(s) at iter = 20

finished writing output restart file(s) at iter = 20


writing ensight output file(s) for current timestep
finished writing ensight output file(s) for current timestep

total wall secs elapsed =  3.00361910e+01
total cpu  secs used by all processors =  3.29250000e+02
total elements on all processors =     388114
cpu  secs/element/iteration =  4.24166611e-05
wall secs/element/iteration =  3.86950625e-06
average wall secs/element for each iteration =  3.47251214e-06
ratio of partition faces to elements =  1.37052521e-01
  about to enter back end

writing ensight output file(s)
finished writing ensight output file(s)

Number of Inlet Boundaries = 1
Area and Average Pressure for Inlet Boundary ID 0 = 1024 m**2, 0 Pa
Mass Flow for Field 0 Through Inlet Boundary ID 0 = -1024 Kg/s
Mass Flow for Field 0 Through All Inlet Boundaries = -1024 Kg/s

Number of Pressure Boundaries = 1
Area and Average Pressure for Pressure Boundary ID 0 = 1024 m**2, 0 Pa
Mass Flow for Field 0 Through Pressure Boundary ID 0 = 1023.98 Kg/s
Mass Flow for Field 0 Through All Pressure Boundaries = 1023.98 Kg/s

Done with boundary mass flow print

ifield,(massin-massout)/massin:         0  1.8497181118102e-05


 *** ending execution of nphase ***
```

**Figure 15. Sample n.out file (continued)**

The third output file that is generated is resid.print which is text file containing only the residual history for convenient plotting. An example is included in Figure 15. Residuals in the code are not "true" residuals, rather they are the RMS of $\Delta\phi$, the change in value of the variable solved for at a given iteration. (Note that this is the value solved for by NPHASE-PSU, see equation (76) in the theory manual.). The columns in resid.print (and the residual prints in nphase.out and n.out for that matter) are column 1: iteration number; column 2: field number (there will be a residual for each field [say liquid and gas]; columns 3-6: RMS($\Delta$u), RMS($\Delta$w), RMS($\Delta$w), RMS($\Delta$p); For multi-field simulations column 7 contains the volume fraction residual for each field, RMS($\Delta\alpha$); For diabatic simulations column 8 contains the enthalpy residual, RMS($\Delta$h); For two-equation turbulence simulations columns 9 and 10 contains the k and $\varepsilon$ residuals, RMS($\Delta$k), RMS($\Delta\varepsilon$) (or q and $\omega$, k and $\omega$, for other turbulence models); column 11 contains the wall-clock seconds per (cell*iteration*field) for that iteration.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1.996e-03 | 0.000e+00 | 0.000e+00 | 1.138e-02 | 0.000e+00 | 0.000e+00 | 6.205e-03 | 2.659e-01 | -7.525e-04 |
| 2 | 1 | 1.968e-03 | 1.736e-04 | 1.733e-04 | 9.678e-03 | 0.000e+00 | 0.000e+00 | 5.294e-03 | 5.726e-02 | -1.426e-03 |
| 3 | 1 | 1.416e-03 | 1.886e-04 | 1.883e-04 | 2.501e-02 | 0.000e+00 | 0.000e+00 | 3.857e-03 | 4.036e-02 | -1.723e-03 |
| 4 | 1 | 1.095e-03 | 1.614e-04 | 1.610e-04 | 2.796e-02 | 0.000e+00 | 0.000e+00 | 3.151e-03 | 3.965e-02 | -1.837e-03 |
| 5 | 1 | 8.819e-04 | 1.336e-04 | 1.332e-04 | 5.263e-02 | 0.000e+00 | 0.000e+00 | 2.715e-03 | 3.441e-02 | -1.635e-03 |
| 6 | 1 | 7.288e-04 | 1.156e-04 | 1.153e-04 | 1.691e-02 | 0.000e+00 | 0.000e+00 | 2.403e-03 | 2.881e-02 | -1.195e-03 |
| 7 | 1 | 6.596e-04 | 1.026e-04 | 1.023e-04 | 1.255e-02 | 0.000e+00 | 0.000e+00 | 2.164e-03 | 2.407e-02 | -7.996e-04 |
| 8 | 1 | 6.206e-04 | 9.439e-05 | 9.413e-05 | 3.820e-03 | 0.000e+00 | 0.000e+00 | 1.977e-03 | 2.025e-02 | -2.927e-04 |
| 9 | 1 | 5.890e-04 | 8.817e-05 | 8.796e-05 | 1.047e-03 | 0.000e+00 | 0.000e+00 | 1.826e-03 | 1.719e-02 | 1.957e-04 |
| 10 | 1 | 5.605e-04 | 8.313e-05 | 8.297e-05 | 2.504e-03 | 0.000e+00 | 0.000e+00 | 1.703e-03 | 1.473e-02 | 6.738e-04 |
| 11 | 1 | 5.347e-04 | 7.913e-05 | 7.901e-05 | 1.099e-02 | 0.000e+00 | 0.000e+00 | 1.600e-03 | 1.275e-02 | 1.061e-03 |
| 12 | 1 | 4.976e-04 | 7.523e-05 | 7.513e-05 | 9.648e-02 | 0.000e+00 | 0.000e+00 | 1.512e-03 | 1.113e-02 | 7.315e-04 |
| 13 | 1 | 4.812e-04 | 7.881e-05 | 7.868e-05 | 6.256e-03 | 0.000e+00 | 0.000e+00 | 1.436e-03 | 9.803e-03 | 4.853e-04 |
| 14 | 1 | 4.669e-04 | 7.254e-05 | 7.225e-05 | 3.667e-03 | 0.000e+00 | 0.000e+00 | 1.370e-03 | 8.714e-03 | 2.734e-04 |
| 15 | 1 | 4.548e-04 | 6.911e-05 | 6.875e-05 | 1.714e-03 | 0.000e+00 | 0.000e+00 | 1.311e-03 | 7.812e-03 | 9.532e-05 |
| 16 | 1 | 4.404e-04 | 6.653e-05 | 6.614e-05 | 8.748e-04 | 0.000e+00 | 0.000e+00 | 1.259e-03 | 7.060e-03 | -6.927e-05 |
| 17 | 1 | 4.198e-04 | 6.422e-05 | 6.381e-05 | 3.219e-03 | 0.000e+00 | 0.000e+00 | 1.213e-03 | 6.426e-03 | -2.094e-04 |
| 18 | 1 | 3.951e-04 | 6.193e-05 | 6.151e-05 | 1.189e-02 | 0.000e+00 | 0.000e+00 | 1.170e-03 | 5.889e-03 | -2.697e-04 |

**Figure 16. Sample resid.print file**

The fourth set of output files are solution restart files. NPHASE-PSU always generates a restart file for each processor after execution completion (and optionally at intermediate iterations/timesteps as defined in "Control Commands" section below). These output files conform to the naming convention nphase_restart_out*xxx* where *xxx* is the 3-digit processor identifier (range from 000 to 999). Solution restarts are invoked by 1) copying each of these output restart files to a corresponding input restart file (i.e., mv nphase_restart_out000 nphase_restart_in000) and then 2) activating the keyword "initialize run with restart file" in nphase.dat.

The fifth set of files are ENSIGHT output files. For steady state cases, each processor generates at least a geometry/grid file: engold.geo.*xxx*, pressure and velocity scalar files: engold.Esca.p00.*xxx,* engold.Esca.u*nn.xxx*, engold.Esca.v*nn.xxx,* and engold.Esca.w*nn.xxx,* where *xxx* is the processor number (0-999) and *nn* are the fields present (00-99). (Since NPHASE-PSU employs a single pressure formulation only engold.Esca.p00.*xxx* is generated). In addition to velocity and pressure files, other ENSIGHT files are generated depending on problem type and user specification (keyword – see Control Commands section below). Specifically, volume fraction files are generated for multi-field simulations: engold.Esca.a*nn.xxx*; enthalpy and temperature files are generated for diabatic simulations: engold.Esca.h*nn.xxx,* engold.Esca.t*nn.xxx* ; turbulence quantity files are generated for turbulence simulations as appropriate for the model used: engold.Esca.k*nn.xxx* (turbulent kinetic energy), engold.Esca.e*nn.xx x*(turbulent dissipation rate, ε), engold.Esca.m*nn.xxx* (eddy viscosity), engold.Esca.uu*nn.xxx,* engold.Esca.vv*nn.xxx,* engold.Esca.ww*nn.xxx,* engold.Esca.uv*nn.xxx,* engold.Esca.vw*nn.xxx,* engold.Esca.wu*nn.xxx,* (Reynolds stresses) for full Reynolds Stress models, engold.Esca.f*nn.xxx* (v2f damping function) and engold.Esca.vv*nn.xxx* (wall normal Reynolds stress) for the v2f model, and engold.Esca.n*nn.xxx* (Spalart-Allmaras kinematic-eddy-viscosity-like variable) for the Spalart-Allmaras model. The user can select two popular flow visualization parameters for ensight file generation: engold.Esca.hel*nn.xxx* (relative helicity density) and engold.Esca.isw*nn.xxx* (intrinsic swirl). For compressible flow simulations, Mach number files are produced: engold.Esca.mach*nn.xxx.* For simulations where interfacial area density transport is invoked, A''' (see eq. (7)) files are produced: engold.Esca.ai*nn.xxx.*

**emerge and emergetrans**

Once NPHASE-PSU has completed there exist many ENSIGHT files in the working directory from which the case is executed. One runs emerge to merge the ENSIGHT files generated on the different processors into single ENSIGHT files for each variable. One runs emergetrans to merge the ENSIGHT files generated on the different processors into single ENSIGHT files for each variable at each timestep that the user has selected for outputting these files. emerge and emergetrans are delivered in software directory EMERGE. They are written in FORTRAN and some C (I/O functions) and are compiled using makefiles (makeem, makeemt) that expect pgf90 and gcc, for which these codes are guaranteed to compile and run. Other compilers are untested.

To run emerge, simply types "emerge" to the UNIX shell. The user will be prompted for the number of processors and the number of fields. Once emerge has completed, the following files are resident in the user's working directory: engold.geo, engold.Esca.p00*,* engold.Esca.u*nn*, engold.Esca.v*nn,* and engold.Esca.w*nn* (where *nn* are the fields present (00-99)). Other case specific merged files are also created (e.g., engold.Esca.a*nn*, engold.Esca.h*nn*, engold.Esca.k*nn*, etc…). Finally emerge generates velocity vector files: engold.Evec.uvw*nn*. All files generated appear in the file engold.case, an example of which appears in Figure 17. Upon execution of emerge the user needs to transfer the case file, and all of the files it points to, to the file system where he will run ENSIGHT. To run ENSIGHT, the case file is read in and each of the other files are automatically brought in.

```
# BOF:  engold.case

FORMAT

type:   ensight gold

GEOMETRY

model:  engold.geo

VARIABLE

scalar per element:       Pressure      engold.p00.Esca
scalar per element:       X-velocity_00 engold.u00.Esca
scalar per element:       Y-velocity_00 engold.v00.Esca
scalar per element:       Z-velocity_00 engold.w00.Esca
scalar per element:              TKE_00 engold.k00.Esca
scalar per element:              TDS_00 engold.e00.Esca
scalar per element:   Eddy-viscosity_00 engold.m00.Esca

vector per element:  Velocity-vector_00 engold.uvw00.Evec
# EOF:  engold.case
```

**Figure 17. Example engold.case file**

To run emergetrans, simply type "emergetrans" to the UNIX shell. The user will be prompted for the number of processors, the number of fields, whether the grid is stationary or moving, the first integer timestep counter, the last integer timestep counter and the

interger timestep increment. emergetrans reads all of the transient per-processor files from the subdirectory, TRANSIN, and writes all output files to a subdirectory, TRANSOUT. Once emergetrans has completed, the following files are resident in TRANSOUT: engold.geo (or engold.geo.*ttttt* (if grid is moving), engold.Esca.p00.*ttttt,* engold.Esca.u*nn.ttttt*, engold.Esca.v*nn.ttttt,* and engold.Esca.w*nn.ttttt* (where *nn* are the fields present (00-99), and *ttttt* is the integer timestep). Other case specific merged files are also created (e.g., engold.Esca.a*nn.ttttt*, engold.Esca.h*nn.ttttt*, engold.Esca.k*nn.ttttt*, etc…). Finally emergetrans generates velocity vector files: engold.Evec.uvw*nn.ttttt*. All files generated appear in the file TRANSOUT/engold_transient.case, an example of which appears in Figure 18. Upon execution of emerge the user needs to transfer the case file and all of the files it points to the file system where he will run ENSIGHT. To run ENSIGHT, the case file is read in and each of the other files are automatically brought in. Upon execution of emergetrans the user needs to transfer the case file, and all of the files it points to, to the file system where he will run ENSIGHT. To run ENSIGHT, the case file is read in and each of the other files are automatically brought in.

```
# BOF:  engold_transient.case

FORMAT

type:   ensight

GEOMETRY

model:  engold.geo

VARIABLE

scalar per element: Pressure      engold.p00.Esca.******
scalar per element: X-velocity_00 engold.u00.Esca.******
scalar per element: Y-velocity_00 engold.v00.Esca.******
scalar per element: Z-velocity_00 engold.w00.Esca.******
vector per element: Velocity_00  0 gold.uvw000 vec.******
scalar per element: TKE_00        engold0k00.Esca.******
scalar per element: TDS_00        engold0e00.Esca.******
scalar per element: Mut_00        engold0m00.Esca.******

TIME
time set: 1
number of steps:    5
filename numbers:
          0          1          2          3          4
time values:
  0.000E+00  0.100E+01  0.200E+01  0.300E+01  0.400E+01
# EOF:  engold_transient.case
```

**Figure 18. Example engold_transient.case file**

## *Tutorials*

Four tutorials are provided as of Ver. 3.1, Rev. 1.12. The first is a general familiarization case, the second and third are specific to the DARPA FDR program which co-funded this documentation. The fourth is a highly compressible case. Each is delivered

### *Tutorials*

Four tutorials are provided as of Ver. 3.1, Rev. 1.12. The first is a general familiarization case, the second and third are specific to the DARPA FDR program which co-funded this documentation. The fourth is a highly compressible case. Each is delivered with the software and they unpack to ./NPHASE-PSU/TUTORIAL_1, ./NPHASE-PSU/TUTORIAL_HIPLATE, ./NPHASE-PSU/TUTORIAL_5415, ./NPHASE-PSU/TUTORIAL_MACH_BUMP. Specifically, the necessary input files are provided for each tutorial: nphase.dat, cobalt.inp, cobalt.bc, and a few other utility files. The user can follow the tutorials below successively applying the NPHASE-PSU pre-processing, execution and postprocessing steps, starting with these files.

**Tutorial Case 1: Turbulent, unsteady, arbitrary polyhedra two-body model**

This purely notional case is for flow around a deformed sphere in the vicinity of a solid wall in a channel. The grid was generated by HARPOON and is hex dominant, but contains "hanging nodes" which gives rise to elements that NPHASE-PSU interprets as arbitrary n-sided-polyhera (i.e., not tetrahedral, hexehedra, prisms or pyramids). The run is turbulent and unsteady.

To start the tutorial the user needs to go to the TUTORIAL_1 directory. There are six files there when the software is unpacked: nphase.dat, run.nphase, cobalt.inp, cobalt.bc attributes.inp and mrf. The first step is to execute fump as illustrated in Figure 18. Here we have chosen 8 processors and no scaling of the geometry. This step generates the 8 files unphase.grid000, unphase.grid001, …., unphase.grid007. Figure 19 shows the file nphase.dat. This is an unsteady run, with four timesteps. 5 inner iterations per timestep and an ENSIGHT output after every timestep.

```
fump
 *** begin execution of fump ***
enter number of processors for domain decomposition:
12
model will be decomposed into 12 partitions
enter scale factor:
1.0
grid will be scaled by 1
enter :
0) metis decomposition :
1) decomposition specified in graph.dat:
0
decomposition will be performed using metis
in problem_size
cobalt.inp is ascii

*file_type        0
ibc =   0, is a Wall_00 boundary
ibc =   1, is a Inlet_00 boundary
ibc =   2, is a Pressure_00 boundary
ndim,nzones,nbc        3        1        3
*nvert,*nface_cobalt,*ncell,maxppf,maxfpc =     40670    277884    125058        4        6
*nface  =    267148, pressure.nbcface  =      1420, pressure.nbcfedge =      4260
inlet.nbcface     =       400, inlet.nbcfedge    =      1600
farfield.nbcface  =         0, farfield.nbcfedge =         0
. . . .
```

**Figure 19. fump run stream for TUTORIAL_1. Bold-black=user supplied, blue=code generated.**

```
#case title:
#2-CELL HANGING NODES

iterations to perform 5

number of fields 1

time accurate simulation
temporal discretization momentum 1
physical timestep in seconds .1
number of physical timesteps 4
transient file write frequency 1

#initialize run with restart file

produce ensight output

dont perform wall match logic

inlet patch 1 0
1.0 0. 0. 1.  1. 0.1 0.1 0. 0. 0 0

pressure patch 1 0
0. 1. 1. 0.1 0.1 0. 0.

turbulent flow high reynolds number k epsilon

constant fluid molecular viscosity 1.e-3
constant fluid density 1

function entry/exit echo off

solver sweeps for u 3
solver sweeps for v 3
solver sweeps for w 3
solver sweeps for p 6
solver sweeps for k 3
solver sweeps for e 3

solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel

initialize u field   1.
initialize k field   .1
initialize e field   .1
```

**Figure 20. nphase.dat file for TUTORIAL_1.**

The file run.nphase is the job submit script that needs to be edited by the user to conform to the system he is running on. The delivered version appears in Figure 20, and is that used to execute on Penn State ARL clusters UWE, STU and SADIE. The user must edit the script to point to mpiexec and the executable (highlighted in blue) on their system, as well as the nodes=x:ppn=x line to conform to the number of nodes and processors for the current run. Standard PBS attributes are included that presumably are machine independent.

The job is submitted to PBS using the qsub command: qsub run.nphase (unless there are other submit scripts available on the user's system). This 388,114 element 12 processor job runs in about 90 seconds on a modern LINUX cluster. The code generates many files, most of them ENSIGHT files, as can be verified by doing an "ls".

```
#!/bin/bash
#PBS -l nodes=1:ppn=12
#PBS -l walltime=1:00:00
#PBS -j oe
#PBS -q batch

cd $PBS_O_WORKDIR
export PATH=/home/local/mpich2-1.0.5/intel/bin:$PATH

# Calculate the number of processors
NPROCS=`wc -l < $PBS_NODEFILE`
echo $NPROCS

# Create nodefile (with duplicates removed) for use by mpdboot
NODEFILE_UNIQ=/tmp/`basename ${PBS_NODEFILE}`.uniq
NODEFILE_MPDS=/tmp/`basename ${PBS_NODEFILE}`.rmpd
cat $PBS_NODEFILE | uniq > $NODEFILE_UNIQ
N_MPDS=`wc -l <  $NODEFILE_UNIQ`
tail -"$(($N_MPDS-1))" $NODEFILE_UNIQ > $NODEFILE_MPDS

echo "all nodes:"
cat $NODEFILE_UNIQ
echo "remote nodes:"
cat $NODEFILE_MPDS

# Boot the MPI2 engine.
/home/local/mpich2-1.0.5/intel/bin/mpdboot --totalnum=$N_MPDS --verbose --rsh=ssh --file=${NODEFILE_MPDS}

#/home/local/mpich2-1.0.5/intel/bin/mpdtrace

############################
# Run your executable
############################

/home/local/mpich2-1.0.5/intel/bin/mpiexec -np $NPROCS ./nphase > n.out
/home/local/mpich2-1.0.5/intel/bin/mpdallexit
```

**Figure 21. run.nphase script for TUTORIAL_1.**

The user can view nphase.out, n.out (which is the standard output and standard error redirect of the NPHASE-PSU run,) and resid.print. The next postprocessing step is to execute emerge which produces processor merged ENSIGHT output files of the final solution. The emerge run stream for this case is included in Figure 21. Once emerge is completed the user can migrate the merged data files and case files to the location he plans on postprocessing using ENSIGHT. These files are: engold.case, engold.geo, engold.w00.Esca, engold.v00.Esca, engold.uvw00.Evec, engold.u00.Esca, engold.p00.Esca, engold.m00.Esca, engold.k00.Esca, engold.e00.Esca. The user can also execute emergetrans for this case which produces processor merged ENSIGHT output files at every timestep. The emergetrans run stream for this case is included in Figure 22. Once emergetrans is completed the user can migrate the merged data files and case files to the location he plans on postprocessing using ENSIGHT. These files are the complete contents of the subdirectory TRANSOUT.

```
emerge

      ******************
      egoldmerge assembly
      ******************

 enter number of partitions/processors
12
 enter number of fields
1
engold.geo.000
          1       49860
          1 numcells_tets(iproc)   =       2555
          1 numcells_hexs(iproc)   =      39823
          1 numcells_prisms(iproc) =        650
.
.
.
no WU file
 reading F-V2F files
no F_V2F file
 reading IBLANK files
no IBLANK file
 reading Temperature files
no Temperature file
 reading Aint files
no Aint
 re-reading velocity files
FORTRAN STOP
```

**Figure 22. emerge run stream for TUTORIAL_1. Bold-black=user supplied, blue=code generated.**

```
emergetrans


      *********************************
      egoldmerge assembly for transients
      *********************************


 enter number of partitions/processors
12
 enter number of fields
1
 enter:
 1) stationary mesh
 2) moving mesh
1
 enter the integer number of the first timestep
0
 enter the integer number of the  last timestep
4
 enter the integer timestep increment
1
enter:
 1) use nphase.out to decide which variables to process
 2) use efiles.dat to decide which variables to process
1

 reading from nphase.out to determine which ensight
 transient files were written by run being postprocessed
 Ensight transient variable files to be merged
 Pressure
 X-velocity
.
.

.
no VW file
 reading WU files
no WU file
 reading F-V2F files
no F file
 reading IBLANK files
no I file
 reading Temperature files
no Temperature file
 reading Aint files
no Aint file
 re-reading velocity files
FORTRAN STOP
```

**Figure 23. emergetrans run stream for TUTORIAL_1. Bold-black=user supplied, blue=code generated.**

The second part of this tutorial involves modifying nphase.dat to run a steady state problem, and then running this problem to convergence, including doing a mid-run restart. To do this copy nphase.dat to nphase.dat.sav, so it can be recovered later. Next change the number of iterations to 250 and comment out the unsteady run keywords as shown (in blue) in Figure 23. Submit and run the job to completion of 250 iterations. To execute a restart comment in the "initialize run with restart file" keyword (as shown in green in Figure 23), and copy all of the nphase_restart_out*xxx* files to nphase_restart_in*xxx*. A convenient way of doing this is to build a small script, mrf, as included in the TUTORIAL_1 directory and as appears in Figure 24. Resubmit, running the code out to 500 iterations, and then run emerge again.

Figure 26 and 27 show the convergence history (TECPLOT used with resid.print as input) and a representative ENSIGHT visualization of the output of this steady state run

```
case title:
2-CELL HANGING NODES

iterations to perform 250

number of fields 1

#time accurate simulation
#temporal discretization momentum 1
#physical timestep in seconds .1
#number of physical timesteps 4
#transient file write frequency 1

#initialize run with restart file

produce ensight output

.
.
.
```

**Figure 24. modified nphase.dat file for TUTORIAL_1.**

```
mv nphase_restart_out000 nphase_restart_in000
mv nphase_restart_out001 nphase_restart_in001
mv nphase_restart_out002 nphase_restart_in002
mv nphase_restart_out003 nphase_restart_in003
mv nphase_restart_out004 nphase_restart_in004
mv nphase_restart_out005 nphase_restart_in005
mv nphase_restart_out006 nphase_restart_in006
mv nphase_restart_out007 nphase_restart_in007
```

**Figure 25. mrf script**

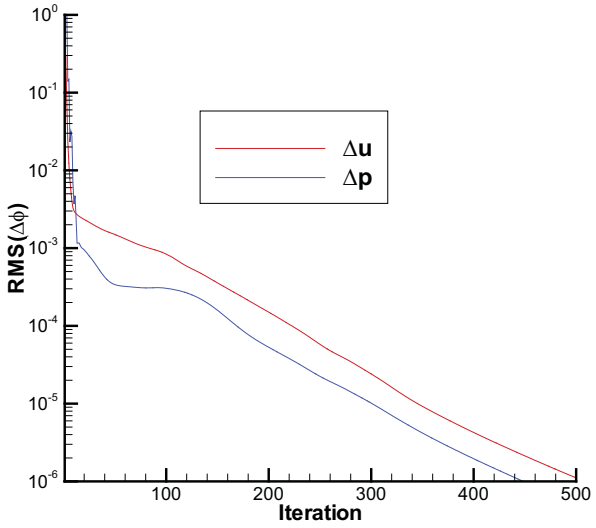



**Figure 26. TECPLOT plot of residual history of steady TUTORIAL_1 simulation.**

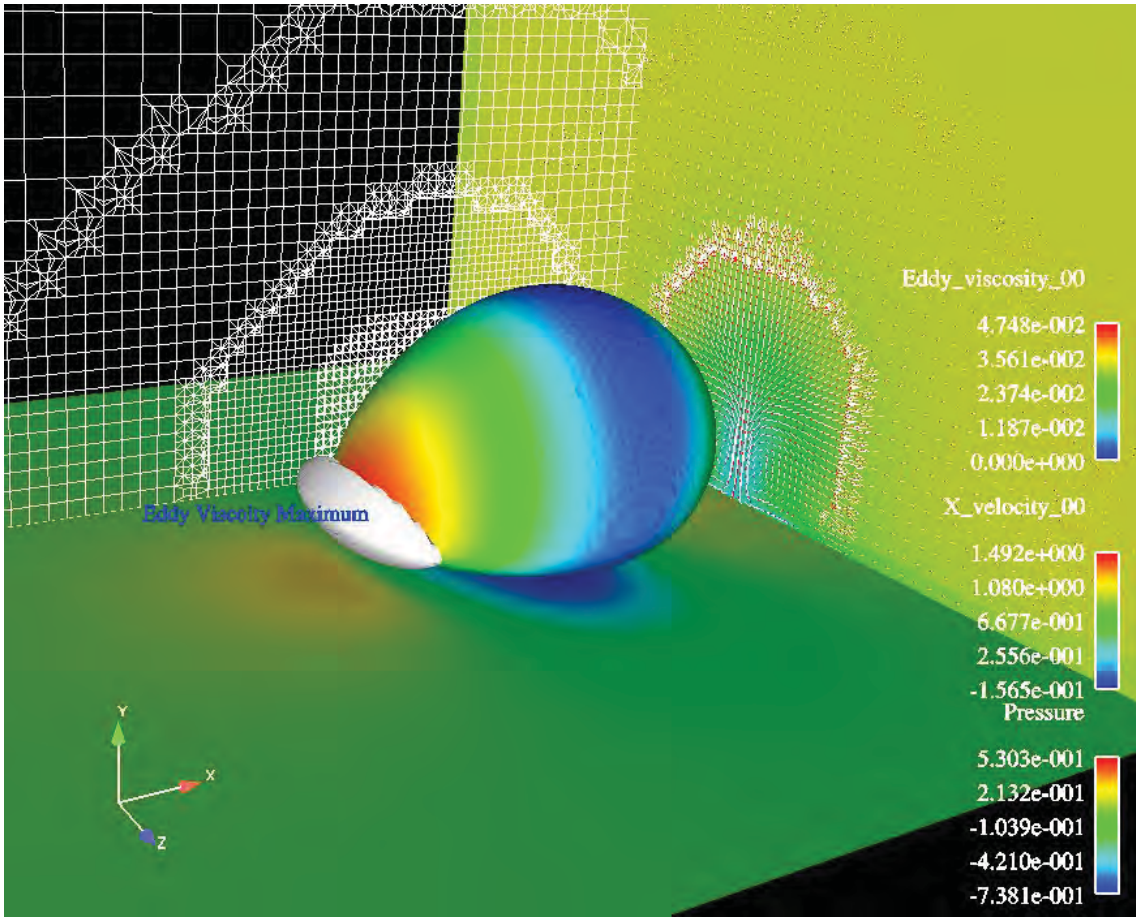



**Figure 27. ENSIGHT visualization of steady TUTORIAL_1 simulation.**

**Tutorial Case 2: Multiphase HIPLATE Simulation**

This case demonstrates the application of NPHASE-PSU to the HIPLATE configuration. The HIPLATE experimental programs were carried out by Ceccio and his colleagues at the Universty of Michigan (Sanders et al., 2006) under the DARPA Friction Drag Reduction program. Comparison of NPHASE-PSU to and calibration of NPHASE-PSU against these data sets were elements of the program (Kunz, et al., 2006, Kunz et al., 2007). HIPLATE is a very high Reynolds number plate configuration tested in the US Navy's Large Cavitation Channel, using microbubble and polymer drag reduction schemes.

Although NPHASE-PSU was validated against and/or applied to the entire spectrum of HIPLATE runs (five gas injection rates, three tunnel speeds, various injector port gas flow splits, with vs. without surfactant, two injector geometries, and two water tunnel test programs [HIPLATE I, II]), we here demonstrate a coarse grid (fast running) application corresponding to the January 2005 program deliverable (validation of code against HIPLATE I data). Specifically, a 12 m/s, $Q_{gas}$=800 scfm case is set up and run. The physical model set used here is very close to that used for the January 2006 deliverable to DARPA which was a validation study comparing NPHASE-PSU simulations to all HIPLATE I data.

To start the tutorial the user needs to go to the TUTORIAL_HIPLATE directory. There are four files there when the software is unpacked: nphase.dat, run.nphase, cobalt.inp, and cobalt.bc. The first step is to execute fump (as described in the *Preprocessing* section and *Tutorial_1* section above), using (here) 1 domain and a scaling factor of 1.0. This step generates a single grid+topology file, unphase.grid000. Figure 28 shows the file nphase.dat file used for this tutorial. The keywords "employ hiplate modeling" are included to instruct nphase to execute the function hiplate_output.c each iteration which prints skin friction data (and, if commented in, other data) to standard output (redirected to n.out in run.nphase). (See the section entitled *Building User Specific/Case Specific Postprocessing* for details on how to modify or adapt this kind of output for different output or simulation case.)

First the code has to be run in single phase mode to generate the comparison flat plate skin friction values at the 6 axial locations on the HIPLATE where shear stress measurements were made. To do this, the user needs to modify the green highlighted sections in Figure 28 as follows: 1) change number of iterations to 500, 2) change porous wall injection velocity from 9.0476 to 0.0000 and the porous wall permeability from 0.7 to 0.0., 3) comment out the relaxation factor lines (allow for defaults). So the code is being run in full 2-phase mode but there is no gas present or injected to it returns a single phase convergence and results.

This single phase job can be run interactively (since its only one processor) or submitted to PBS using the qsub command: qsub run.nphase (unless there are other submit scripts available on the user's system). This single phase job processor job executes 500 iterations in about 55 wall seconds on a modern LINUX cluster. Figure 29 shows a section of the n.out file for this single phase run that includes the computed $C_f$ distribution at the six axial HIPLATE stations. To run the microbubble drag reduction case, the nphase.dat

```
iterations to perform 20000
number of fields 2

produce ensight output

restart file write frequency 1000
dont perform wall match logic

employ hiplate modeling

gravity vector 0. 9.81 0.
employ modified pressure
reference density for modified pressure 1

overwrite inlet patch boundary conditions on restart
overwrite porous wall patch boundary conditions on restart
overwrite pressure patch boundary conditions on restart
overwrite_gas_molecular_viscosities 200 1.5e+3

inlet patch 2 0
12.0 0. 0. 1000. 0.999999 1.33e-7 1.76e-6 0. 1.
12.0 0. 0. 1.    0.000001 0.      0.      0. 1.

porous wall patch 2 0
9.0476 0. 0. 1000. 0.000001 1.33e-7 1.76e-6 0. 0.7  1.
9.0476 0. 0. 1.    0.999999 0.      0.      0. 0.7  1.

pressure patch 2 0
0.0  1000.  0.999999 1.33e-7 1.76e-6  0.
0.0  1.     0.000001 0.      0.       0.

constant fluid density 1000. 1.
constant fluid molecular viscosity 1.e-3 1.5e-5
constant fluid surface tension .072 0.

Turbulence model for each field 1 0

function entry/exit echo off
adiabatic flow

solver sweeps for u 3
solver sweeps for v 3
solver sweeps for a 3
solver sweeps for p 10
solver sweeps for tke 3
solver sweeps for tds 3

solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel
employ rhie chow for dispersion terms

relaxation factor for u .05 .05
relaxation factor for v .05 .05
relaxation factor for a .05 .05

initialize u field 12. 12.
initialize v field 0. 0.
initialize w field 0. 0.
initialize a field .999999 0.000001
initialize tke field 1.33e-7 0.
initialize tds field 1.76e-6 0.
```

```
interfacial area transport model for each field 0 1
interfacial area coalescence model for each field 0 3
initialize interfacial area using volume fraction and characteristic length
interfacial area feeds back into bubble diameter frequency 100
constant field characteristic diameter 99. .000400
interfield drag models 1
1 2 6 .8
bubble cluster drag modification
interfield nondrag models 5
1 2 2 5 50.
1 2 2 7 1. 2. -2.0 0.5 2.
1 2 3 3 .1
1 2 1 6 .25
1 2 5 3 1.
```

**Figure 28. nphase.dat file for HIPLATE tutorial**

```
  iter fld   ru        rv        rw        rp        ra        raint     rh        rk     ...
     1  1 3.580e-03 4.631e-01 0.000e+00 5.767e+03 2.210e-08 0.000e+00 0.000e+00 1.104e-05 ...
     1  2 1.005e+00 1.672e-14 0.000e+00 5.767e+03 2.139e-08 3.208e-04 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  2.57927225e-01  2.57927225e-01  2.57927225e-01  2.57927225e-01   ...
clipcount0         64
     2  1 7.490e-02 9.948e-02 0.000e+00 1.157e+03 3.543e-08 0.000e+00 0.000e+00 1.324e-03 ...
     2  2 5.775e-01 9.039e-02 0.000e+00 1.157e+03 3.467e-08 5.207e-04 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  4.48393267e+00  5.93114814e+00  6.51944986e+00  6.64137946e+00   ...



   499  1 8.279e-06 1.624e-07 0.000e+00 3.534e-02 6.881e-06 0.000e+00 0.000e+00 2.490e-06 ...
   499  2 5.797e-01 4.853e-01 0.000e+00 3.534e-02 4.581e-06 6.947e-02 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  1.59246792e+02  1.49083851e+02  1.39654688e+02  1.36211604e+02   ...
   500  1 8.102e-06 1.540e-07 0.000e+00 3.885e-02 3.922e-06 0.000e+00 0.000e+00 2.440e-06 ...
   500  2 1.103e-01 1.703e-01 0.000e+00 3.885e-02 3.541e-06 5.406e-02 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  1.59246794e+02  1.49083857e+02  1.39654745e+02  1.36211706e+02   ...
final cf1,cf2,cf3,cf4,cf5,cf6  1.59246794e+02  1.49083857e+02  1.39654745e+02             ...
avgcf cf1,cf2,cf3,cf4,cf5,cf6  1.56126868e+02  1.41946021e+02  1.39721464e+02             ...
stdv  cf1,cf2,cf3,cf4,cf5,cf6  1.36051236e+01  1.79146591e+01  1.66518929e+01             ...
```

**Figure 29. Snippets from standard output (n.out) for single phase TUTORIAL_HIPLATE simulation, illustrating case specific skin friction coefficient output.**

```
  iter fld   ru        rv        rw        rp        ra        raint     rh        rk     ...
     1  1 2.007e-04 1.944e-02 0.000e+00 7.098e+05 2.068e-03 0.000e+00 0.000e+00 1.102e-05 ...
1  2 7.272e-02 1.021e-02 0.000e+00 7.098e+05 2.067e-03 9.201e+03 0.000e+00 0.000e+00      ...
cf1,cf2,cf3,cf4,cf5,cf6  2.57927225e-01  2.57927225e-01  2.57927225e-01  2.57927225e-01   ...
clipcount0         85
     2  1 4.648e-01 2.057e-01 0.000e+00 6.805e+05 1.970e-03 0.000e+00 0.000e+00 1.373e-03 ...
     2  2 7.036e-02 3.113e-03 0.000e+00 6.805e+05 1.969e-03 7.497e+03 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  4.53209563e+00  5.96370783e+00  6.54783080e+00  6.66829586e+00   ...
```

```
 19999  1 1.463e-05 4.050e-06 0.000e+00 8.241e-01 3.974e-04 0.000e+00 0.000e+00 1.243e-05 ...
 19999  2 1.963e-03 1.345e-04 0.000e+00 8.241e-01 3.906e-04 8.455e+01 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  7.61674767e+01  9.95633653e+01  1.12126232e+02  1.20645034e+02   ...
 20000  1 1.447e-05 4.117e-06 0.000e+00 8.499e-01 3.973e-04 0.000e+00 0.000e+00 1.226e-05 ...
 20000  2 2.261e-03 1.369e-04 0.000e+00 8.499e-01 3.905e-04 8.445e+01 0.000e+00 0.000e+00 ...
cf1,cf2,cf3,cf4,cf5,cf6  7.61674747e+01  9.95633580e+01  1.12101175e+02  1.20644965e+02   ...
final cf1,cf2,cf3,cf4,cf5,cf6  7.61674747e+01  9.95633580e+01  1.12101175e+02             ...
avgcf cf1,cf2,cf3,cf4,cf5,cf6  7.61676731e+01  9.95634673e+01  1.12181731e+02             ...
stdv  cf1,cf2,cf3,cf4,cf5,cf6  1.64306879e-04  2.11569909e-04  5.42004104e-02             ...
```

**Figure 30. Snippet from standard output (n.out) for two-phase TUTORIAL_HIPLATE simulation, illustrating case specific skin friction coefficient output.**

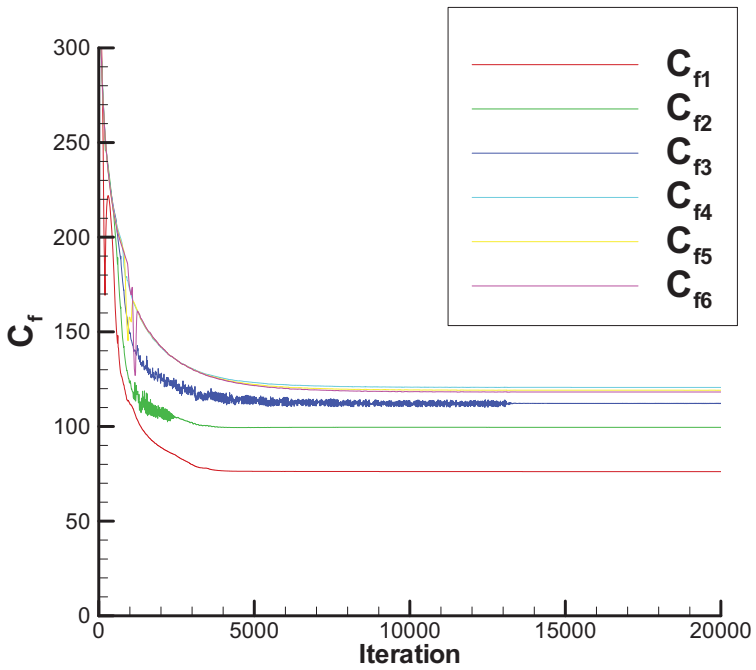



**Figure 31. Predicted skin friction coefficient vs. iteration at the six HIPLATE stations for the two-phase TUTORIAL_HIPLATE simulation.**

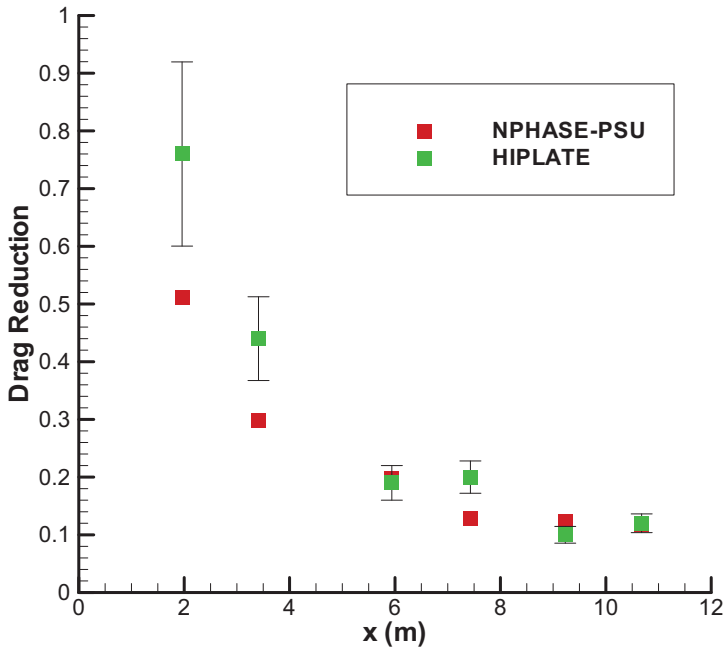



**Figure 32. Predicted skin friction coefficient vs. x at the six HIPLATE stations for the two-phase TUTORIAL HIPLATE simulation.**

Figure 30 shows a section of the n.out file for the two-phase microbubble drag reduction run that includes the computed $C_f$ distribution at the six axial HIPLATE stations.

Figure 31 shows the computed 2-phase skin friction iteration history. This solution converged to the final $C_f$ values in about 5000 iterations. Figure 32 shows the predicted drag reduction vs. x for this case. These results are nearly identical to the January 2006 submission (grid is coarser here). Figure 33 shows contours of predicted gas volume fraction with the vertical axis scaled by a factor of 100 in order that the thin microbubble layer can be viewed.

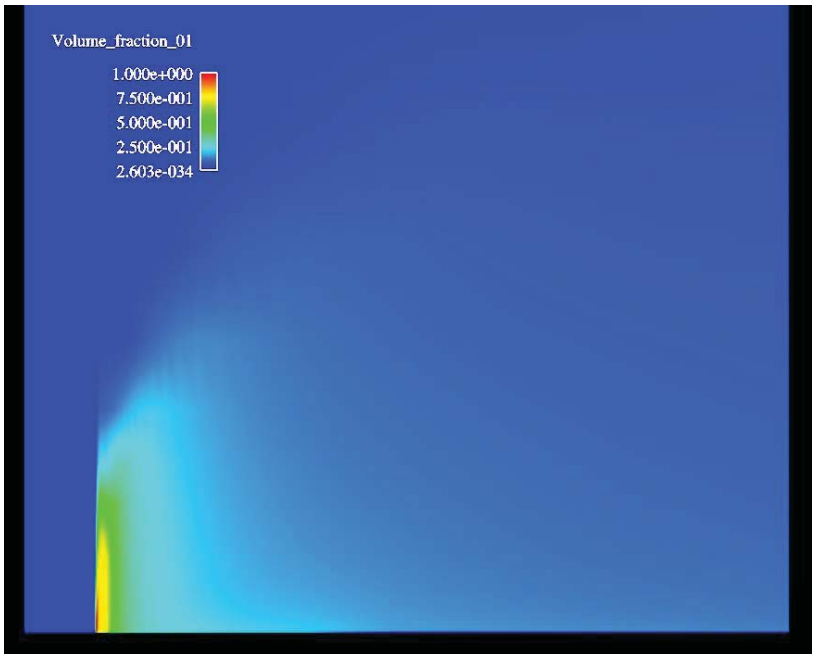



**Figure 33. Predicted contours of gas volume fraction for the two-phase TUTORIAL_HIPLATE simulation. Vertical axis scaled by a factor of 100.**

**Tutorial Case 3: Multiphase 5415 Simulation**

This case demonstrates the application of NPHASE-PSU to a Navy relevant configuration in 1-phase and 2-phase modes. The geometry is the 5415 model, a hull form representative of the Arleigh-Burke class destroyer. This model has been extensively tested at the Naval Surface Warfare Center Carderock Division (http://www.dt.navy.mil/hyd/surshi-mod/index.html, for example.) The grid was generated using ICEM-CFD and is a standard tetra+prism mesh. Specifically, the mesh contains 1438852 elements of which 463837 are prisms extruded in (nominally) five layers from the triangulated hull surface. This comparatively coarse mesh is suitable for wall function turbulence modelling. The hull surface incorporates a standard solid wall patch and a porous wall patch, aft of the bow dome, from which gas is injected. Figure 34 shows two views of the mesh used.

To start the tutorial the user needs to go to the TUTORIAL_5415 directory. There are five files there when the software is unpacked: nphase.dat, run.nphase, cobalt.inp, cobalt.bc and mrf. The first step is to execute fump (as described in the *Preprocessing* section and *Tutorial_1* section above), using (here) 32 domains and a scaling factor of .001 (to convert the grid from mm to m.) This step generates the 32 files unphase.grid000, unphase.grid001, …., unphase.grid0031. Figure 35 shows the file nphase.dat. The keywords "employ model 5415 modeling" are included to instruct nphase to execute the function model5415_output.c each iteration which generates a wetted area and a net vehicle drag coefficient printout to standard output (redirected to n.out in run.nphase). (See the section entitled *Building User Specific/Case Specific Postprocessing* for details on how to modify or adapt this kind of output for different output or simulation case.) A freestream velocity of 2.2134 m/s is specified (model scale) and the k-ε model is selected.

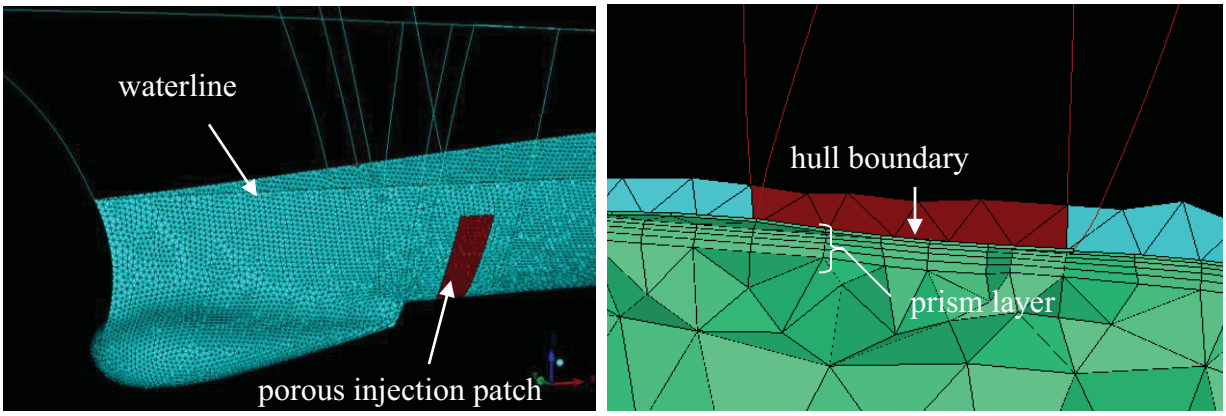



**Figure 34. Views of mesh employed for 5415 tutorial**

First the user should run the code to generate a single phase solution for comparison to multiphase runs to be made later. To do this, the user needs to modify the green highlighted sections in Figure 35 as follows: 1) change number of iterations to 2500, 2) change porous wall injection velocity from 0.3000 to 0.0000 and the porous wall permeability from 0.7 to 0.0. So the code is being run in full 2-phase mode but there is no gas present. The job is submitted to PBS using the qsub command: qsub run.nphase (unless there are other submit scripts available on the user's system). This 32 processor job

executes 2500 iterations in about 20000 wall seconds on a modern LINUX cluster (banyan.dt.navy.mil). As before, the user can view nphase.out, n.out, and resid.print. Then emerge is run and the user can migrate the merged data files and case files to the location he plans on postprocessing using ENSIGHT. These files are: engold.case, engold.geo, engold.w00.Esca, engold.v00.Esca, engold.uvw00.Evec, engold.u00.Esca, engold.p00.Esca, engold.m00.Esca, engold.k00.Esca, engold.e00.Esca.

```
#case title:
#5415

iterations to perform 10000

employ model 5415 modeling

number of fields 2
#initialize run with restart file
produce ensight output
restart file write frequency 100

dont perform wall match logic
read wall proximity from file

gravity vector 0. 0. -9.81
employ modified pressure
reference density for modified pressure 1000.
#simple hydrostatic pressure treatment
overwrite inlet patch boundary conditions on restart
overwrite porous wall patch boundary conditions on restart
overwrite pressure patch boundary conditions on restart
#overwrite_gas_molecular_viscosities 200 1.5e+3

inlet patch 2 0
2.2134 0. 0. 1000. .99999999999 .002939 .02618 0. 0.
2.2134 0. 0. 1.    .00000000001 .0      .0     0. 0.

pressure patch 2 0
0.0  1000.  .99999999999 .002939 .02618  0. 0.
0.0  1.     .00000000001 .0      .0      0. 0.

porous wall patch 2 0
0.3000 0. 0. 1000. .00000000001 .002939 .02618 0. 0.7  1.
0.3000 0. 0. 1.    .99999999999 .0      .0     0. 0.7  1.

turbulence model for each field 1 0

constant fluid molecular viscosity 1.e-3 1.5e-5
constant fluid density 1000. 1.
constant fluid surface tension .072 0.

spatial discretization momentum 2

function entry/exit echo off
adiabatic flow

solversweepsforu 3 3
solversweepsforv 3 3
solversweepsforw 3 3
solversweepsfora 3 3
solversweepsfork 3 3
solversweepsfore 3 3
```

```
solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel
petscprintnorm

employ rhie chow for dispersion terms

#relaxation factor for u .2 .2
#relaxation factor for v .2 .2
#relaxation factor for w .2 .2
#relaxation factor for a .2 .2
#relaxation factor for k .2 .2
#relaxation factor for e .2 .2
#
initialize u field   2.2134 2.2134
initialize v field   0. 0.
initialize w field   0. 0.
initialize a field .99999999999 .00000000001
initialize k field   .002939 .0
initialize e field   .02618 .0

clips_on_velocity -5. 10. -5. 5. -5. 5.
#interfacial area transport model for each field 0 1
#interfacial area coalescence model for each field 0 3
#initialize interfacial area using volume fraction and characteristic length
#interfacial area feeds back into bubble diameter frequency 100
constant field characteristic diameter 99. .000400
interfield drag models 1
1 2 6 0.8
bubble cluster drag modification
interfield nondrag models 5
1 2 2 5 50.
1 2 2 7 1. 2. -2.0 0.5 2.
1 2 3 3 .1
1 2 1 6 .25
1 2 5 3 1.
```

**Figure 35. nphase.dat file for TUTORIAL_5415.**

Figure 35 shows the convergence history for this single phase case. The code converges well and then eventually flatlines due to weak unsteadiness in the solution.

```
2500 1 4.207e-06 2.098e-06 8.752e-07 1.029e-03 2.231e-11 0.000e+00 1.127e-06 2.070e-04
       ...
  2500  2 1.480e-01 2.008e-01 6.633e-02 1.029e-03 2.395e-11 0.000e+00 0.000e+00 0.000e+00
       ...
Awetted dot i:  2.2368721880023e-05 m^2
Viscous  drag force:  3.3070651056143e+01
Pressure drag force:  1.0161932291489e+01
Total    drag force:  4.3232583347632e+01
```

Figure 36 shows a section of the n.out file which illustrates the computed frontal wetted area and drag force computation.

The second part of this tutorial involves running a notional microbubble drag reduction case, for comparison to the single phase drag just computed. To execute this part of the tutorial, the user should employ the nphase.dat file as it occurs in the software deliverable directory, shown in Figure 34. Numerous multifield extensions to a single phase run are apparent. Specifically:

1) a wall proximity computation (for all cells not just wall adjacent cells) is automatically carried out to support some of the interfacial force models (wall-lift). This wall proximity computation can be quite time consuming the first run of this case (several

minutes). So the user should not be concerned. After a few minutes this process will be complete and the files wall_proximity.dat*xxx* (x*xx* = processor number, 000-031 here) will appear in the working directory and the iterations will begin. For all subsequent runs using this grid the user can forego the time consuming wall proximity calculation by reading these files on input, by commenting in the keyword: "read wall proximity from file".

2) Gravity of course needs to be specified. Here a conventional modified pressure approach is used with a reference density set equal to the liquid density. This zeroes the magnitude of the buoyancy term in the liquid momentum equation which has been found to improve convergence. A simple hydrostatic head initialization and exit pressure boundary specification is invoked.

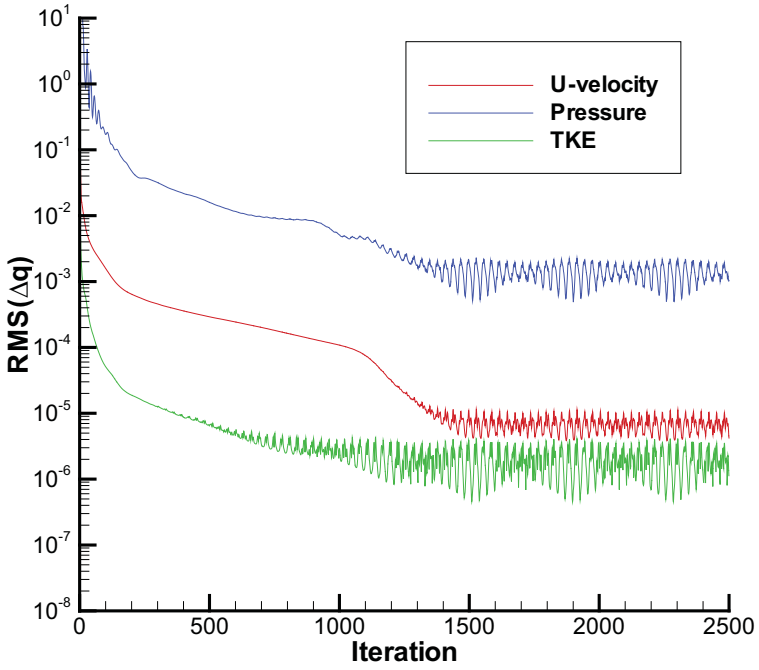



**Figure 36. TECPLOT plot of residual history of single phase TUTORIAL 5145 simulation.**

```
  iter fld   ru        rv        rw        rp        ra        rh        rk        re     ...
.
.
.
.
  2500  1 4.207e-06 2.098e-06 8.752e-07 1.029e-03 2.231e-11 0.000e+00 1.127e-06 2.070e-04 ...
  2500  2 1.480e-01 2.008e-01 6.633e-02 1.029e-03 2.395e-11 0.000e+00 0.000e+00 0.000e+00 ...
Awetted dot i:  2.2368721880023e-05 m^2
Viscous  drag force:  3.3070651056143e+01
Pressure drag force:  1.0161932291489e+01
Total    drag force:  4.3232583347632e+01
```

**Figure 37. Snippets from standard output (n.out) for single phase TUTORIAL_5145 simulation, illustrating case specific drag output.**

3) Inlet, pressure and porous boundary conditions have attributes specified for both fields. Note here that we specify liquid and gas volume fractions of $1x10^{-6}$ within 0 and 1, again arising for robustness (matrix singularity) reasons. If a porous wall normal velocity of 0.38 m/s were set, this would yield a nondimensional gas injection rate, $Q_a/(Q_a+Q_w) \cong 0.44$ which corresponds to the HIPLATE 12m/s, 800 scfm case analyzed in TUTORIAL_HIPLATE (based on a representative $Q_w \cong Ub(\delta-\delta^*)$, where $\delta$ and $\delta^*$ are computed based on standard flat plate momentum integral relations at the axial injector location). Accordingly a range of injection velocities ranging from 0.0 to 0.40 m/s were run, thereby spanning the $Q_a/(Q_a+Q_w)$ range of all bubbly flow HIPLATE runs. The particular case detailed here is for $U_{inj}$=0.3 m/s.

4) Turbulence scalars are solved only for the continuous field here.

5) Density and viscosity are set for both fields

6) Solver sweeps are set for both fields for all scalars (except pressure). Relaxation factors are set for both fields for all scalars (except pressure).

7) Initial values are set for both fields for all scalars. Initially the domain is filled with water ($\alpha^1$=1, $\alpha^2$=0.)

8) The remainder of the keywords specify various interfacial dynamics, mass transfer and interfacial area density transport models.

The grid supplied in this tutorial is fairly coarse for a high Reynolds number boundary layer flow. Only five prism layers are employed and wall function resolution is specified. Accordingly the first order numerics associated with the volume fraction in this case gives rise to some numerical smearing of the gas layer.

The model set chosen for this simulation includes all relevant interfacial dynamics (drag and non-drag) forces (see HIPLATE tutorial above and Kunz et al., (2006)). The interfacial force model set is identical to that presented for the HIPLATE tutorial.

Typically, for microbubble drag reduction applications one runs the code until it "flatlines". At that point one can look at the $C_D$ history with iteration and its standard deviation. If the standard deviation is small compared to the magnitude of the predicted drag, then one accepts the mean $C_D$ value as the prediction.

In this $U_{inj}$=0.3 simulation, the code was run for 10000 iterations. Figure 37 shows the first 2000 iterations of the convergence history for this two-phase case. The code flatlines due to gas velocity cutoffs that are specified (see Figure 34) which keep the gas velocities reasonable in regions where the gas fraction is vanishingly small ($O(10^{-6})$). Figure 38 shows the drag history for this case illustrating that for MBDR applications, 2500 iterations is adequate for injection scheme assessment (and this is what was used for the other five 2-phase cases run as discussed below). Figure 39 shows a section of the n.out file which illustrates the computed frontal wetted area and drag force computation. Figure 40 shows a view of the hull surface pressure and a gas volume fraction isosurface, $\alpha^{gas}$=0.5, illustrating the predicted topology of the bubble layer.

The last set of results included in this tutorial parameterize the gas flow rate as might be done in a design assessment. Specifically, the code was run an additional five times, spanning $Q_a/(Q_a+Q_w)$ values from 0.0 through nearly 0.5. The results of these computations are compiled in Figure 41 and Figure 42.

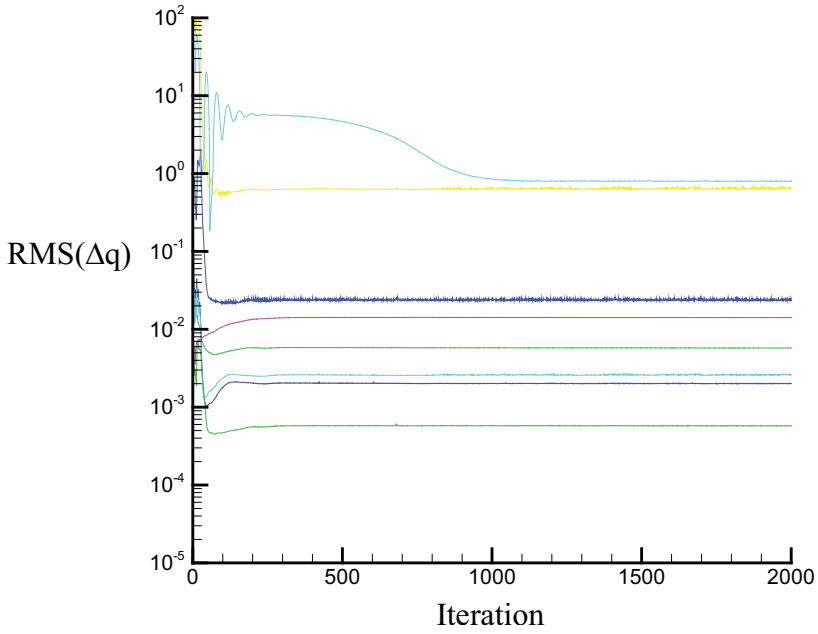



**Figure 38. TECPLOT plot of residual history for two-phase TUTORIAL_5145 simulation.**

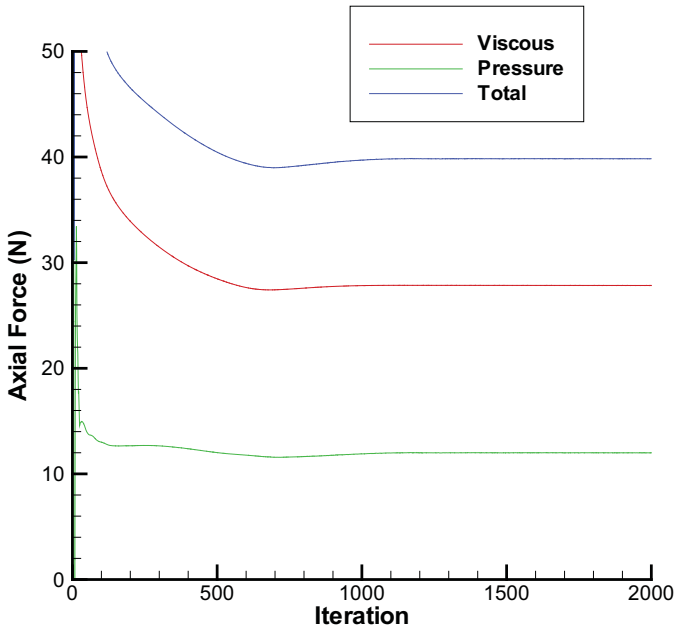



**Figure 39. TECPLOT plot of drag force history for two-phase TUTORIAL_5145 simulation.**

```
  iter fld   ru        rv        rw        rp        ra        rh        rk        re     ...
.
.
.
.
10000  1 5.809e-03 2.045e-03 2.639e-03 6.509e-01 1.482e-02 0.000e+00 6.091e-04 2.577e-02  ...
10000  2 3.713e-01 4.032e-01 1.596e-01 6.509e-01 1.099e-02 0.000e+00 0.000e+00 0.000e+00  ...
Awetted dot i:  2.2368721880023e-05 m^2
Viscous  drag force:  2.7868256856116e+01
Pressure drag force:  1.2050646663068e+01
Total    drag force:  3.9918903519184e+01
```

**Figure 40. Snippets from standard output (n.out) for two phase TUTORIAL_5145 simulation, illustrating case specific drag output.**

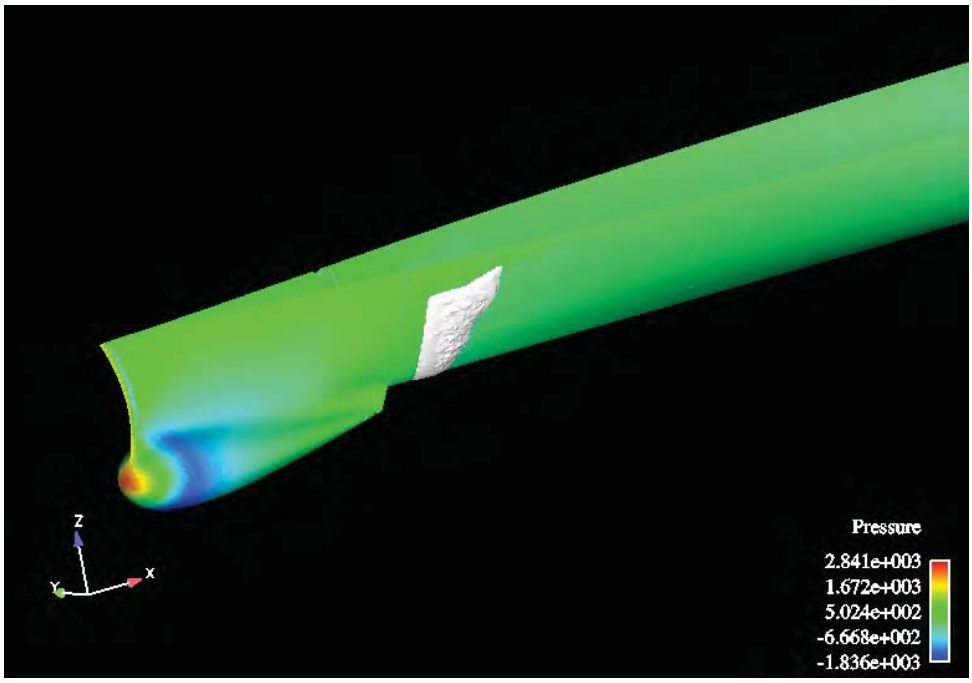



**Figure 41. Surface pressure contours and $\alpha^{gas}$=0.5 isosurface for $U_{inj}$=0.3 m/s TUTORIAL_5145 simulation.**

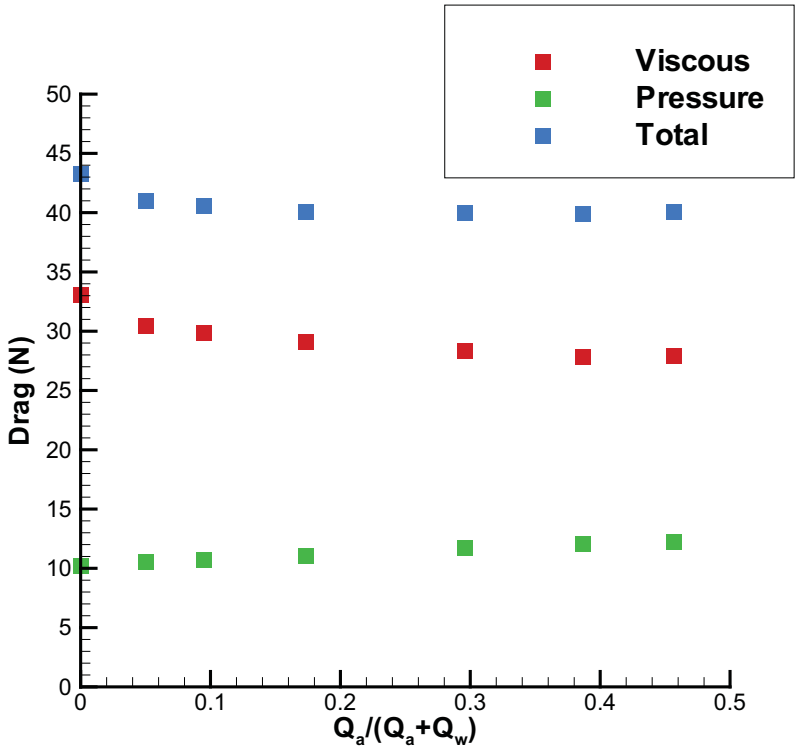



**Figure 42. Predicted Total Hull Drag vs. Non-Dimensional Gas Injection for 5415 Simulations**

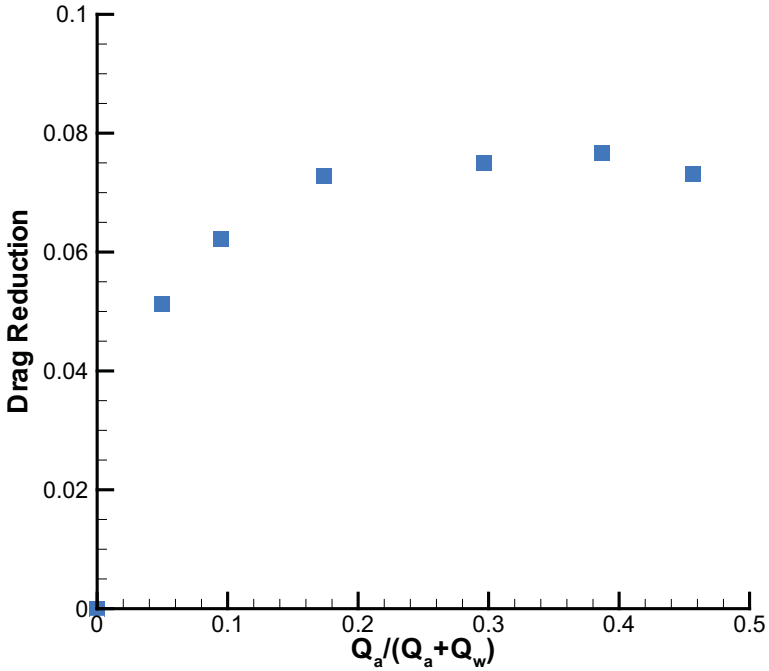



**Figure 43. Predicted Total Drag Reduction vs. Non-Dimensional Gas Injection for 5415 Simulations**

**Tutorial Case 4: Compressible Flow over a Bump**

This case demonstrates the application of NPHASE-PSU to a highly compressible flow field. Specifically, a calorically perfect air flow at an inlet Mach number of 0.7 flows inviscidly over a circular arc bump in a channel. Inviscid runs are accomplished by specifying Symmetry boundaries on walls and setting the viscosity to a tiny number.

To start the tutorial the user needs to go to the TUTORIAL_MACH_BUMP directory. There are three files there when the software is unpacked: nphase.dat, cobalt.inp, and cobalt.bc. The first step is to execute fump (as described in the *Preprocessing* section and *Tutorial_1* section above), using 12 domains and a scaling factor of 1.0. This step generates grid+topology files, unphase.grid000 – unphase.grid011.

To define initial conditions the following procedure was invoked:

1) The keywords `"use stagnation enthalpy as primitive energy transport variable"` means just that, so stagnation enthalpy must be set at the inlet and as an initial guess for enthalpy. This is the standard approach used for highly compressible flow (but not necessarily for weakly compressible flow where static enthalpy can be more convenient [e.g., thermal driven convection]).
2) Assuming an inlet static temperature and pressure of 293.15K (20$^{o}$C)and 101325 Pa, we compute an inlet density from the perfect gas law and the defined specific heat ratio and gas constant for air ($\gamma$=1.4, R=287 J/kg*K) ➔ $\rho$=1.204328093 kg/m$^3$ which appears in the inlet and pressure patch attributes.
3) The inlet and initial velocity is computed taking an inlet Mach number of 0.7 and computing the sound speed from the inlet static temperature for calorically perfect air (a=sqrt($\gamma$RT)).
4) The stagnation enthalpy is computed by determining the static enthalpy for calorically perfect air from h=$C_P$T, with $C_P$=$\gamma$R/($\gamma$-1), and h0=h+u$^2$/2.

The code is run on 12 processors for 2500 iterations by submitting an appropriately modified run.nphase script to the queue. The run converges fairly slowly due to the fine grid, 2$^{nd}$ order convection numerics, the use of simple (i.e. non-characteristic) inflow/outflow boundary conditions, and the sharp shock that arises.

The user can then run emerge (12 processor, 1 field) and migrate the merged data files (engold.geo, engold.u00.Esca, engold.v00.Esca, engold.w00.Esca, engold.uvw00.Evec, engold.r00.Esca, engold.p00.Esca, engold.mach00.Esca, engold.h00.Esca, engold.t00.Esca, and case file (engold.case) to the location he plans on postprocessing using ENSIGHT.

Figure 44 shows a contour plot of the predicted Mach number contours.

```
#case title:
#INVISCID BUMP

iterations to perform 2500

number of fields 1

#initialize run with restart file

produce ensight output
restart file iteration write frequency 1000

dont perform wall match logic

#pamb=101325 Pa, Tamb=293.15 K
inlet patch 1 0
240.24145833  0. 0. 1.204328093 1. 0. 0. 0. 323327.154 0 0 #stagnation enthalpy

pressure patch 1 0
101325. 1.204328093 1. 0. 0. 323327.154 0. #stagnation enthalpy

constant fluid molecular viscosity 1.0e-15

use stagnation enthalpy as primitive energy transport variable 0

strongly coupled compressibility
perfect gas compressibility parameters 0
287. 1.4
single phase heating on
continuity error treatment for enthalpy
print umin and umax
print tmin and tmax
#clips on velocity -500. 500. -500. 500. -500. 500.

spatialdiscretizationmomentum 2
spatialdiscretizationenthalpy 2

function entry/exit echo off

#relaxation factor for p .3
relaxation factor for u 0.7
relaxation factor for v 0.7

solver choice for velocity components jacobiuvw
solver choice for pressure petsc
#solver choice for pressure jacobi
parallel strategy for pressure corrector: matrixlevel

initialize u field 240.24145833
initialize p field   101325.
initialize h field   323327.154 #stagnation enthalpy
```

**Figure 44. nphase.dat file for TUTORIAL_MACH_BUMP.**

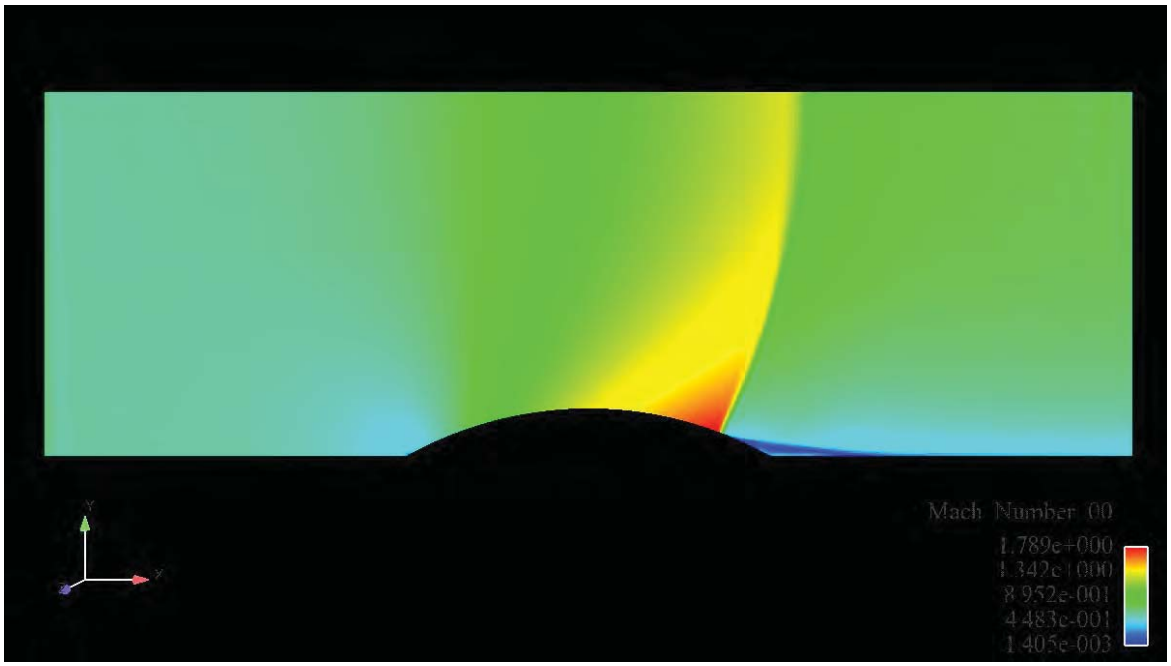



**Figure 45. Predicted Mach number contours for TUTORIAL_MACH_BUMP.**

## Tutorial Case 5: Flow around a Shrouded Gear

To demonstrate application of NPHASE-PSU to gear windage simulation, we have provided a tutorial case of a 72-tooth shrouded spur gear investigated by Diab et al. (2004). Considering the symmetry of the problem, only one side of a single tooth sector is simulated, decreasing simulation size substantially. The simulation is performed in a reference frame rotating with the gear, so that a stationary mesh can be used with a moving wall boundary condition applied on the shroud. Figure 46 shows the nphase.dat file used for this case. Because the simulated flow is incompressible and there are no inlets or outlets, pressure must be set to a reference value (0) somewhere within the domain for the equations of motion to remain well-posed. We have chosen to anchor the pressure value implicitly (appearing in the pressure equation matrix). This is specified through inclusion of the line, `anchor pressure implicitly`.

Attributes specific to gear windage simulation are specified in the line `employ diab gear modeling 1000.0 2 72 0`. The first argument is the gear rotation speed in radians per second. The second argument is an integer denoting the strategy used for the rotation. 0 is a stationary grid with a stationary reference frame, and implicit wall motion, but is applicable only to cylinders. 1 is for a stationary grid in the relative reference frame with absolute velocities. 2 is a stationary grid in the relative frame with relative velocities. 3 is a rotating grid in the absolute reference frame with absolute velocities. Here, we have chosen to use option 2, for a stationary grid in the relative frame with relative velocities. The third argument is the number of teeth. The fourth argument is a Boolean toggle denoting whether the output should be made relative to the first time-step. As this is a steady state simulation, this parameter is irrelevant and is set to zero. Due to the lack of inlets and outlets, the cyclic boundary conditions, and the manner in which flow is driven, this simulation takes many iterations to converge. To maintain numerical stability, it is best to run it with first order discretization for the first 50,000 iterations, then uncomment the second order discretization commands in nphase.dat, and run it again, re-initializing from restart files.

To run this case, NPHASE requires that the high and low cyclic boundary faces be sorted. First, sort_cyclic is run, which generates cobalt.inp.sort. Next, this file must be re-named cobalt.inp. From here, fump is run. The current case is decomposed for 12 processors using METIS and a scale of 1. The sector angle for cyclic simulation is 5 degrees for a 72-tooth gear, so this is input when prompted. After running fump to generate unphase.grid000 – unphase.grid011, the run script can be appropriately modified for the batch scheduling system and mpiexec used, and then submitted to the queue. After running with first order accuracy, nphase.dat should be modified for second order accuracy and initializing from restart files. To move the nphase_restart_out### files to nphase_restart_in###, run mrf. The run script can then be re-submitted to the queue to achieve a second order accurate solution. Figure 47 shows streamlines colored by pressure for this simulation. The streamlines were created in EnSight by applying computational symmetry to the solution as a post-processing step.

```
# Case title
# Diab gear windage tutorial

iterations to perform 50000
restart file iteration write frequency 5000
#initialize run with restart file
number of fields 1

produce ensight output

employ diab gear modeling 1000.0 2 72 0

#spatial discretization momentum 2
#spatial discretization turbulence 2

dont perform wall match logic

anchor pressure implicitly

pressure gradient at walls extrapolation

constant fluid molecular viscosity 1.e-5
constant fluid density 1.

turbulence model for each field 1
turbulence model reynolds number regime for each field 0

production destruction ratio clip 500

solver choice for velocity components jacobiuvw
solver choice for pressure petsc
parallel strategy for pressure corrector: matrixlevel

#relaxation factor for u .7
relaxation factor for k 0.7
relaxation factor for e 0.7

initialize u field 0.
initialize p field 0.
initialize k field 0.015
initialize e field 5.0
```

**Figure 46. nphase.dat file for TUTORIAL_GEAR.**

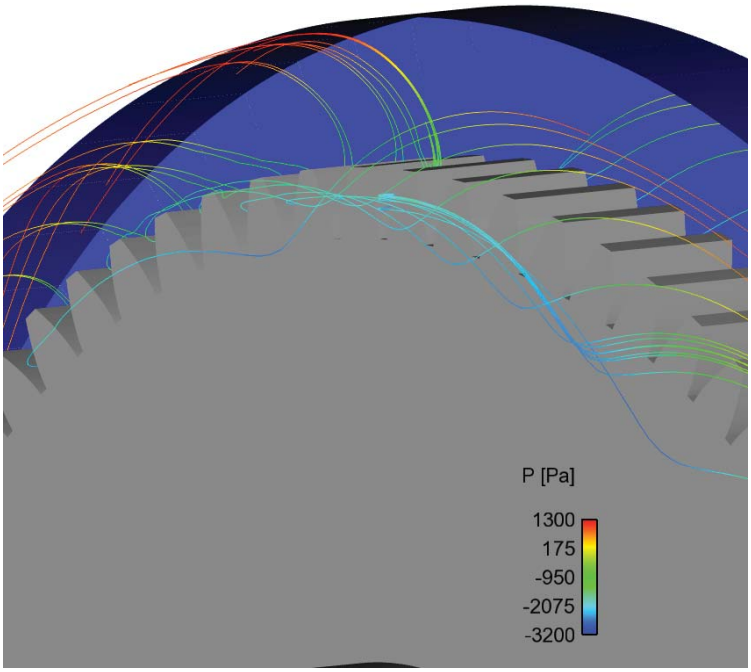



**Figure 47. Streamlines colored by pressure from shrouded gear tutorial.**

### *Other Items of Interest*

**Running Two-Dimensional Problems**

NPHASE-PSU can be run in two-dimensional mode quite easily by specifying a one-element thick grid and defining symmetry boundaries on the front and back faces. It does not matter how thick the element is. So, for example, a pure 2D triangular mesh will extrude to a one-element thick prism mesh.

**Building User Specific/Case Specific Postprocessing**

Often an NPHASE-PSU user is interested in case specific output, or even case specific coding that modifies some element of execution (say grid motion, boundary condition, hard coded initialization, etc…). For these situations, this section documents the procedures to incorporate such coding in a fashion that is accessed from the front end, i.e., does not affect the execution of the code for cases where user supplied keywords are not supplied. The process is illustrated with an example – defining a user specific output for the 5415 simulation carried out in the preceding tutorial.

The first step is to define a new keyword and attendant integer pointer flag. In this example we add the following lines to the main NPHASE-PSU data C-structure in NPHASE-PSU, *struct data*, which is defined in nphase_struct.h:

```
//   5415 specific stuff:
int *imodel5415_modeling ;
```

Next, storage is allocated for this pointer in constmemory.c:

```
// 5415 specific:
var.imodel5415_modeling= (int *) malloc(sizeof(int)) ;
```

The value of this integer is initialized to 0 (i.e., not set) in set_parameters.c:

```
int *imodel5415_modeling=var.imodel5415_modeling ;  //type definition
.
.
.
// 5415 specific
*imodel5415_modeling=0;
```

The value of this integer pointer is broadcast to all processors in broadcast.c:

```
int *imodel5415_modeling=var.imodel5415_modeling ;  //type definition
.
.
.
MPI_Bcast(imodel5415_modeling,1,MPI_INT,0,MPI_COMM_WORLD);
```

User access to setting this parameter is accomplished by defining a keyword in author_****.c, where *** is either “abcd”, “efgh”, “ijkl”, “mnop”, “qrstu”, or “vwxyz” depending on the fist letter of the keyword. Here we define the keyword “employ model 5145 modeling” in author_efgh.c:

```
int *imodel5415_modeling=var.imodel5415_modeling ;  //type definition
.
.
.
/*-------------------------------------------------------------*/
  else if(strncmp(keyword,"employmodel5415modeling"
                    ,strlen(keyword)) == 0) {
    *imodel5415_modeling = 1;
    *idid = 1;
    fprintf(fo,"%s",line) ;
  }
```

Note that keyword is defined with no spaces or special characters.

Now that all of the data structure and front end hooks are in place, the user can build or modify a function to execute what they wish. In this case, a new function is defined, model5415_output.c, the source of which appears in Figure 43. This function is called, in this case, at the end of every iteration, from nphase.c:

```
int *imodel5415_modeling=var.imodel5415_modeling ;  //type definition
.
.
.
if(*imodel5415_modeling==1)model5415_output();
```

Once these coding modifications have been made, the object model5415_output.o is add to the file Obj_nphase, and the entire code is recompiled using make –f makefile nphase *(options)*. The 5415 tutorial presented above includes the use of this coding as shown in the nphase.dat files included in Figure 34. The output generated by this coding is shown in Figure 36 and Figure 39.

```
#include "nphase_struct.h"
#include "mpi.h"
#include <math.h>
#include <stdio.h>
extern struct boundary_patch wall;
extern struct boundary_patch porwall;
extern struct data var;
extern int myid;
       int  model5415_output()
{
/*-------------------------------------------------
       called from:
       nphase
       V2.0 baseline code
       initials       comment
       rfk
-------------------------------------------------*/
       int *nfield=var.nfield;
       int *nnode=var.nnode;
       int ibf,nstride,inode,fstride,ifield;
        real sumforce,sumforce0,sumflux,sumflux0;
        real awetted,awetted0;
        real rinf,uinf,cd;
        real *a=var.a;
        real *u=var.u;

       entered("model5415_output") ;

    uinf=2.2134;
    rinf=1000;

// wetted area

    awetted=0.;
    for(ibf=0;ibf<=wall.nbcface-1;++ibf){
    inode=wall.bcface_n[ibf];
    awetted+=wall.bamag[ibf];
    }
    for(ibf=0;ibf<=porwall.nbcface-1;++ibf){
    inode=porwall.bcface_n[ibf];
    awetted+=porwall.bamag[ibf];
    }
    MPI_Reduce(&awetted,&awetted0,1,mpireal,MPI_SUM,0,MPI_COMM_WORLD);
    if(myid==0)printf("Awetted: %20.13e m^2\n",awetted0);

// drag force on boat

    sumforce=0.;
    for(ifield=0;ifield<=*nfield-1;++ifield){
    fstride=ifield*wall.nbcface ;
    nstride=ifield* *nnode;
      for(ibf=0;ibf<=wall.nbcface-1;++ibf){
      inode=wall.bcface_n[ibf] ;
      sumforce+=wall.tmlt[ibf+fstride]* *(a+inode+nstride) * wall.bamag[ibf]* *(u+inode+nstride);
      }
      for(ibf=0;ibf<=porwall.nbcface-1;++ibf){
      inode=porwall.bcface_n[ibf] ;
      sumforce+=porwall.tmlt[ibf+fstride]* *(a+inode+nstride) * porwall.bamag[ibf]* *(u+inode+nstride);
      }
    }
    MPI_Reduce(&sumforce,&sumforce0,1,mpireal,MPI_SUM,0,MPI_COMM_WORLD);
    if(myid==0)printf("Drag force: %20.13e\n",sumforce0);
    if(myid==0)cd=sumforce0/(.5*rinf*uinf*uinf*awetted0);
    if(myid==0)printf("CD=: %20.13e\n",cd);

       exiting("model5415_output") ;

       return 0 ;
}
```

**Figure 48. model5415_output.c**

**Turbomachinery**

*Documentation not yet written.*

**Homogeneous Multiphase Flow**

*Documentation not yet written.*

### *Control Commands*

This section includes a description of most of the keyword commands available in NPHASE. These appear in nphase.dat in free format. They need not appear in any particular sequence. It is not exhaustive and many commands are not likely to be used by most users.

# 1.1. Comment Card

Any line in the "nphase.dat" input file that starts with a "#" is considered a comment card. The remainder of the line is ignored.

# 1.2. Job Control

## 1.2.1. Case Title

Specifies a title for the output of the job. Optional. A maximum of 132 characters can be used in the title. The title itself must appear on following line

```
case title:
$Title
```

## 1.2.2. Iterations to Perform

Specifies the number of iterations to perform in job. For a restart job, the specified number of iterations will be performed after the restart file is read. For a time accurate simulation, this is the number of inner iterations per physical timestep.

```
iterations to perform  $NITER
```

## 1.2.3. Over Write Boundary Conditions on Restart

During a restart from a previous run (See "initialize run with restart file" command) the boundary conditions are specified by the information in the restart file. If the user wants the boundary conditions specified by the "nphase.dat" input file, the over write boundary condition on restart command must be used. The command can be used on various boundary conditions as specified below.

```
over write inlet patch boundary conditions on restart
over write far field patch boundary conditions on restart
over write pressure patch boundary conditions on restart
over write wall patch boundary conditions on restart
over write porous patch boundary conditions on restart
```

## 1.2.4. Number of Physical Time Steps

The number of time steps for the case can be specified with this command.

```
number of physical time steps  $NTSTEPS
```

## 1.2.5. Physical Time Step in Seconds

This command is used to specify a constant time step size for a transient analysis.

```
physical time step in seconds  $DT
```

## 1.2.6. Time Accurate Simulation

Specifies that a transient analysis is to be run. The number of time steps and time step size can be specified with the "number of physical time steps" and "physical time step size in seconds" command

```
time accurate simulation
```

### 1.2.7. Transient File Write Frequency

Specifies the number of time steps between saving a time step numbered restart file. This command can be used to save restart files for postprocessing animation.

```
restart file write frequency  $NRST_FREQ
```

### 1.2.8. Volume Fraction Normalization

Specifies normalization of volume fraction equation. Normalization will ensure the sum of the volume fraction of all fields equals one.

```
carver volume fraction normalization
```

Specifies that volume fraction normalization is not used.

```
do not employ carver volume fraction normalization
```

### 1.2.9. Continuity Error Treatment for Momentum

Adds terms to the LHS and RHS of the discrete momentum equations that ensures that if the linear solver is brought in then no spurious sources of momentum will arise due to mass imbalances

```
continuity error treatment for momentum
```

### 1.2.10. Continuity Error Treatment for Enthalpy

Adds terms to the LHS and RHS of the discrete enthalpy equations that ensures that if the linear solver is brought in then no spurious sources of enthalpy will arise due to mass imbalances

```
continuity error treatment for enthalpy
```

### 1.2.11. Momentum Cross Diffusion

This command specifies that the nonorthogonal components of the viscous shear term are included in the momentum equation. For a general grid, this command should be included in the input deck. See equation **Error! Reference source not found.** in the theory manual.

```
cross diffusion included in momentum equations
```

### 1.2.12. Number of PISO Correction Steps

The number of PISO corrections steps is required when the PISO algroithm is employed (see piso employed command). The PISO algorithm is a more recent varient of the SIMPLE algroithm.

```
number of piso corrections  $NPISO_STEPS
```

### 1.2.13. Parallel Strategy for Pressure Correction

Specifies whether the pressure correction equation is solved implicitly across all inter-processor boundaries.

```
parallel strategy for pressure correction: matrix level
```

```
parallel strategy for pressure correction: explicit partition boundary update
```

### 1.2.14. PISO Algorithm Employed

The PISO algorithm is a more recent variant of the SIMPLE algorithm. The PISO algorithm can be specified using this command. The number of PISO correction steps can be specified using the “number of piso corrections” command.

```
piso employed
```

### 1.2.15. Relaxation Factor

The relaxation factor command is used to improve numerical convergence by the addition of numerical damping to eliminate oscillations and improve stability in the solution. The form of the relaxation factor command is:

```
relaxation factor for u $RFU
```

```
relaxation factor for v $RFV
relaxation factor for w $RFW
relaxation factor for a $RFA
relaxation factor for k $RFK
relaxation factor for e $RFE
relaxation factor for h $RFH
```

Values for the false time step command are specified as:
$RFU - User input value for false time step in “u” momentum equation
$RFV - User input value for false time step in “v” momentum equation
$RFW - User input value for false time step in “w” momentum equation
$RFA - User input value for false time step in phasic volume fraction equation
$RFK - User input value for false time step in turbulent kinetic energy equation
$RFE - User input value for false time step in turbulent dissipation rate equation

### 1.2.16. SIMPLEC Factor

The use of this command can modify the numerical method to use the original SIMPLE algorithm ($SIMPLEC=0.) or a varient call the SIMPLEC algorithm ($SIMPLEC=1.). The SIMPLEC algorithm is the default value.

```
simplec factor  $SIMPLEC
```

### 1.2.17. Solver Choice for All Scalars Jacobi

This command specifies the use of the Jacobi algorithm for solving the volume fraction, k-e turbulence and energy equations. This is the default solver for the volume fraction, k-e turbulence and energy equations.

```
solver choice for all scalars jacobi
```

### 1.2.18. Solver Choice for Enthalpy Jacobi

This command specifies the use of the Jacobi algorithm for solving the energy equation. This is the default solver for the energy equation.

```
solver choice for all scalars jacobi
```

### 1.2.19. Solver Choice for Pressure Jacobi

This command specifies the use of the Jacobi algorithm for solving the pressure correction equations in the SIMPLE algorithm. This is the default solver for the pressure correction equations

```
solver choice for pressure jacobi
```

### 1.2.20. Solver Choice for Turbulence Scalars Jacobi

This command specifies the use of the Jacobi algorithm for solving the k-e turbulence equations. This is the default solver for the k-e turbulence equations

```
solver choice for turbulence scalars jacobi
```

### 1.2.21. Solver Choice for Velocity Components Jacobi

This command specifies the use of the Jacobi algorithm for solving the momentum equations in the SIMPLE algorithm. This is the default solver for the momentum equations

```
solver choice for velocity components jacobi
```

### 1.2.22. Solver Sweeps

This command is used to specify the number of linear solver sweeps for each linear equation solver. Command are available to specify the number of solver sweeps separately for each equation or in combination..

```
solver sweeps for u  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for v  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for w  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
```

```
solver sweeps for p  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for a  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for h  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for tke  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
solver sweeps for tds  $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN
```

### 1.2.23. Spatial Discretization

This command is used to specify the spatial discretization used for the convective term in the governing equations. The default discretization is the hybrid scheme.

```
spatial discretization momentum $ISPATIAL
spatial discretization turbulence $ISPATIAL
spatial discretization volume fraction $ISPATIAL
spatial discretization turbulence $ISPATIAL
```

Values for the spatial discretization command are specified as:
$ISPATIAL = 0 ( Use 1st order hybrid scheme)
1(Use 1st order upwind scheme)
2(Use 2nd order upwind scheme)

### 1.2.24. Temporal Discretization

This command is used to specify the spatial discretization used to transfrom the convective term in the governing equations. The default discretization is the hybrid scheme.

```
temporal discretization momentum $ITEMPOR
temporal discretization turbulence $ITEMPOR
temporal discretization volume fraction $ITEMPOR
temporal discretization turbulence $ITEMPOR
```

Values for the temporal discretization command are specified as:
$ITEMPOR = 1 (Use 1st order [Euler Implicit] scheme - default)
$ITEMPOR = 2 (Use 2nd order backward difference in time)

### 1.2.25. Thin Layer Approximation Employed/ Not Employed

Specifies whether or not a thin layer approximation is employed in the construction of viscous terms.

```
thin layer approximation employed
thin layer approximation not employed
```

## 1.3. Geometry Control

### 1.3.1. Cylindrical Coordinates

Specifies use of cylindrical coordinates (R-Z). Where axial flow (Z) is in the (X) coordinate direction, and radial (R) flow is in the (Y) coordinate direction. Requires (Y=0) to align with (R=0).

```
Cylindrical Coordinates
```

## 1.4. Boundary Conditions

### 1.4.1. Cyclic Boundary

All cyclic boundaries are specified in the “unphase.grid” file.

### 1.4.2. Far Field Boundary

### 1.4.3. Far Field Patch

All far field boundaries are specified in the “unphase.grid” file.

```
farfield patch  $NCARD, $FACEID
```

| |
|---|
| $U, $V, $W, $RHO, $VF, $TKE, $TDS, $H,$P // for field one, repeat NCARD times for other fields |

Specified far field boundary conditions for all farfield boundary faces with FACEID. Far field values are specified as:
$NCARD - Number of input lines imediately following. Usually equals NFIELD.
$FACEID - Identification of particular far field patch. (e.g., FACEID = 3 for Farfield_03)
$U - Cartesian "u" velocity at far field boundary (m/s)
$V - Cartesian "v" velocity at far field boundary (m/s)
$W - Cartesian "w" velocity at far field boundary (m/s)
$RHO - Fluid Density at inlet (kg/$m^3$), used for backflow conditions only
$VF - volume fraction at far field boundary
$TKE - Turbulent Kinetic Energy at far field boundary ($m^2/s^2$)
$TDS - Turbulence Dissipation at far field boundary
$H - Enthalpy at far field boundary (deg-K)
$P - Pressure at far field boundary (Pa)

## 1.4.4. Inlet Boundary

### 1.4.4.1.Inlet Patch

The Inlet Patch command is used to specify uniform inlet boundary conditions. All inlet boundaries are specified in the "unphase.grid" file.

| |
|---|
| Inlet Patch $NCARD, $FACEID |
| $U, $V, $W, $RHO, $VF, $TKE, $TDS, $H, $P, $T1, $T2 // for field one, repeat NCARD times for other fields |

Specified inlet flow boundary conditions for all inlet boundary faces with FACEID. Inflow values are specified as:
$NCARD - Number of input lines imediately following. Usually equals NFIELD.
$FACEID - Identification of Faces to apply inlet patch. Face ID is specifed in "unphase.grid" file.
$U - Cartesian "u" velocity at inlet (m/s)
$V - Cartesian "v" velocity at inlet (m/s)
$W - Cartesian "w" velocity at inlet (m/s)
$RHO - Fluid Density at inlet (kg/$m^3$), used for backflow conditions only
$VF - volume fraction at inlet
$TKE - Turbulent Kinetic Energy at inlet ($m^2/s^2$)
$TDS - Turbulence Dissipation at inlet
$H - Enthalpy at inlet (deg-K). Specified either stagnation or static depending on ienergy_variable
$P – Stagnation pressure at inlet
$T1 = 0,1 for subsonic, supersonic treatment
$T2 = 0,1 for incompressible, compressible

### 1.4.4.2.Inlet Velocities Specified in Absolute Frame.

| |
|---|
| inlet velocities specified in absolute frame |

### 1.4.4.3.Inlet Velocity Specified in Cylindrical Coordinates.

| |
|---|
| inlet velocities specified in cylindrical coordinates |

### 1.4.4.4.Inlet Velocities Specified Transiently.

| |
|---|
| inlet velocities specified transiently |

### 1.4.4.5.Inlet Profile Specified in inflow.pro

| |
|---|
| inlet profile specified in inflow.pro |

### 1.4.4.6.Inlet Profiles 2D specified in inflow.pro

| |
|---|
| inlet profile 2d specified in inflow.pro |

## 1.4.5. Porous Boundary

### 1.4.5.1.Porous Wall Patch

The Porous Wall Patch command is used to specify uniform boundary conditions through a porous surface. All porous boundaries are specified in the "unphase.grid" file.

| |
|---|
| porous wall patch $NCARD, $FACEID |
| $U, $V, $W, $RHO, $VF, $TKE, $TDS, $T, $PERMA // for field one, repeat NCARD times for other fields |

Specified inlet flow boundary conditions for all inlet boundary faces with FACEID. Inflow values are specified as:
$NCARD - Number of input lines imediately following. Usually equals NFIELD.
$FACEID - Identification of Faces to apply inlet patch. Face ID is specifed in "unphase.grid" file.
$U - Cartesian "u" velocity through porous wall (m/s)
$V - Cartesian "v" velocity through porous wall(m/s)
$W - Cartesian "w" velocity through porous wall (m/s)
$RHO - Fluid Density through porous wall ($kg/m^3$)
$VF - volume fraction at through porous wall
$TKE - Turbulent Kinetic Energy through porous wall($m^2/s^2$)
$TDS - Turbulence Dissipation through porous wall
$T - Temperature through porous wall (deg-K)
$PERMA - permeability of porous wall

## 1.4.6. Pressure Boundary

### 1.4.6.1.Pressure Patch

The Pressure Patch command is used to specify uniform pressure boundary conditions. All pressure boundaries are specified in the "unphase.grid" file.

| |
|---|
| Pressure Patch $NCARD, $FACEID |
| $P, $RHO, $VF, $TKE, $TDS, $H, $P0, $TYPE // for field one, repeat NCARD times for other fields |

Specified inlet flow boundary conditions for a patch of faces. Inflow values are specified as:
$NCARD - Number of input lines imediately following. Usually equals NFIELD.
$FACEID - Identification of Faces to apply inlet patch. Face ID is specifed in "unphase.grid" file.
$P - Pressure at pressure boundary (Pa)
$RHO - Fluid Density at pressure boundary ($kg/m^3$), used for backflow conditions only
$VF - volume fraction at pressure boundary, used for backflow conditions only
$TKE - Turbulent Kinetic Energy at pressure boundary ($m^2/s^2$), used for backflow conditions only
$TDS - Turbulence Dissipation at pressure boundary, used for backflow conditions only
$H - Enthalpy at pressure boundary (J/kgK), used for backflow conditions only
$P0 – Stagnation pressure at pressure boundary (Pa), currently unused
$TYPE – Included for future support. Just use 0 for now

## 1.4.7. Symmetry Boundary

All symmetry boundaries are specified in the "unphase.grid" file.

## 1.4.8. Wall Boundary

### 1.4.8.1.Wall Patch

The Wall Patch command is used to modify wall boundary conditions. All wall boundaries are specified in the "unphase.grid" file.

| |
|---|
| Wall Patch $NCARD, $FACEID |
| $XTW, $XMW, $XFW, $XFW0, $XFW1, $XFW2 //heat transfer card<br>$XTW_STRUCT, $XFW_TENSION //wall structural parameters card<br>$XTW_VEL, $XFW_XVEL, $XFW_VVEL, $XFW_ZVEL //wall motion card |

| |
|---|
| //for field one, repeat these 3 lines NCARD times for other fields |

NOTE: one does not need to specify any wall attributes if the wall is adiabatic, rigid and statinoary. If any of these attributes is non-default, all must be specified
Specified wall boundary conditions for all wall boundary faces with FACEID. Wall attributes are specified as:
$NCARD - Number of input lines imediately following. Usually equals NFIELD.
$FACEID - Identification of Faces to apply wall patch. Face ID is specifed in "unphase.grid" file.
$XTW – Type of wall boundary:
iwt=0: adiabatic
iwt=1: specify temperature
iwt=2: specify heat flux (xfw= 0. = adiabatic)
iwt=3: specify heat transfer coeficient (not implemented yet)
iwt=4: conjugate heat transfer across thin matching walls (continuous flux on either side of wall)
iwt=5: conjugate 1D heat transfer across thickness of assumed material adjacent to wall face with temp specified on outside face of that material
iwt=6: conjugate 1D heat transfer across thickness of assumed material adjacent to wall face with htc and tfilm specified on outside face of that material
$XMW – If this is a matching wall this integer is the the negative of the faceid of the matching wall. If this is not a matching wall it is ignored.
$XFW - Real # parameter:
if iwt=1, xfw=specified temperature in $^{o}$K
if iwt=2, xfw=specified heat flux in W/m$^2$
if iwt=3, xfw=specified heat transfer coefficient (not implemented)
if iwt=4, xfw is unused since heat flux is continuous across matching internal walls
if iwt=5, xfw=specified temperature on outside of material assumed adjacent to wall patch
if iwt=6, xfw=specified film temperature (K) on outside of material assumed adjacent to wall patch
$XFW0 - unused for iwt=0->4
if iwt=5, xfw0=shell thermal conductivity (J/m*s*K) of material assumed adjacent to wall patch
if iwt=6, xfw0= heat transfer coefficient (W/m$^2$*K) on outside of material assumed adjacent to wall patch
$XFW1 - unused for iwt=0->4
if iwt=5, xfw1=shell thermal conductivity (J/m*s*K) of material assumed adjacent to wall patch
if iwt=6, xfw1= shell thickness (m) of material assumed adjacent to wall patch
$XFW2 - unused for iwt=0->5
if iwt=6, xfw2=shell thermal conductivity (J/m*s*K) of material assumed adjacent to wall patch

# 1.5. Initial Conditions

## 1.5.1. Initialize with Restart

The restart file "nphase_restart_in" will be used to specify the boundary and interior node values.

| |
|---|
| initialize run with restart file |

## 1.5.2. Initial u Velocity

This command initializes the u velocity to a constant value specified by the user:.

| |
|---|
| initialize u velocity $u_field0, $u_field1,...,$u_fieldN |

## 1.5.3. Initial v Velocity

This command initializes the v velocity to a constant value specified by the user:.

| |
|---|
| initialize u velocity $v_field0, $v_field1,...,$v_fieldN |

## 1.5.4. Initial w Velocity

This command initializes the w velocity to a constant value specified by the user:.

| |
|---|
| initialize u velocity $w_field0, $w_field1,...,$w_fieldN |

### 1.5.5. Initial Pressure Field

This command initializes the pressure field to a constant value specified by the user:.

initialize p field $w_field0, $w_field1,...,$w_fieldN

.

### 1.5.6. Initial Volume Fraction Field

This command initializes the volume fraction field to a constant value specified by the user:.

initialize volume fraction field $VF_field0, $VF_field1,...,$VF_fieldN

### 1.5.7. Initial Interfacial Area Density

This command initializes the volume fraction field to a constant value specified by the user:.

initialize ai field $AI_field0, $AI_field1,...,$AI_fieldN

### 1.5.8. Initial Turbulent Kinetic Energy (tke) Field

This command initializes the tke field to a constant value specified by the user:.

initialize tke field $TKE_field0, $TKE_field1,...,$TKE_fieldN

### 1.5.9. Initial Turbulent Dissipation Rate (tds) Field

This command initializes the turbulent dissipation rate field to a constant value specified by the user:.

initialize tds field $TDS_field0, $TDS_field1,...,$TDS_fieldN

### 1.5.10. Initial Enthalpy Field

This command initializes the enthalpy field to a constant value specified by the user:.

initialize h field $H_field0, $H_field1,...,$H_fieldN

### 1.5.11. Hard Coded Initialization

hard coded initialization

## 1.6. Physical Models

### 1.6.1. Environmental Properties

#### 1.6.1.1.Laminar Flow

This command specifies that the flow is to be treated as laminar flow..

laminar flow

#### 1.6.1.2.Gravity Vector

This command specify the direction and magnitude of gravity.

gravity vector $gx, $gy, $gz

Where,
$gx == “x” direction gravity magnitude,
$gy == “y” direction gravity magnitude,
$gz == “z” direction gravity magnitude

#### 1.6.1.3. Employ Modified Pressure

This command specifies that a conventional gravity modified pressure treatment will be used.

employ modified pressure

#### 1.6.1.4. Reference Density for Modified Pressure

This command specifies the reference density for the modified pressure treatment.

reference density for modified pressure $rhoref

#### 1.6.1.5. Employ Simple Hydrostatic Pressure Treatment

This command specifies that a simple single phase hydrostatic initialization of the pressure will be used based on local coordinate ($\nabla p = \rho g x$) and modified pressure reference density.

```
employ simple hydrostatic presure treatment
```

#### 1.6.1.6. System Rotation About X Axis in Radians Per Second

Specifies the system rotation about the x-axis in radians per second.

```
system rotation about x axias in radians per second  $XRADS
```

### 1.6.2. Fluid and Solid Properties

#### 1.6.2.1.Constant Fliud Molecular Viscosity

The constant fluid molecular viscosity is used to specify the molecular viscosity for each field.

```
constant fluid molecular viscosity  $MU_field1, $MU_field1,...,$MU_fieldN
```

#### 1.6.2.2.Constant Fluid Density

The constant fluid density command is used to specify the fluid density for each field..

```
constant fluid density  $RHO_field1, $RHO_field1,...,$RHO_fieldN
```

#### 1.6.2.3.Constant Fluid Surface Tension

The constant fluid surface tension command is used to specify the fluid surface tension for each field..

```
constant fluid surface tension  $SIGMA_field1, $SIGMA_field1,...,$SIGMA_fieldN
```

#### 1.6.2.4.Constant Fluid Specific Heat at Constant Pressure

The constant fluid specific heat at constant pressure command is used to specify the fluid specific heat (Cp) for each field..

```
constant fluid specific heat at constant pressure  $CP_field1, $CP_field1,...,$CP_fieldN
```

#### 1.6.2.5.Isothermal Compressibility Parameters

This command is used to specify compressibility parameters.

```
isothermal compressibility parameters  $NCARD
$RHO_ref, $PRESS_ref, $1/C^2  // for field one, repeat  NCARD times for other fields
```

#### 1.6.2.6.Solid Region Density

The density of solid regions can be specified by this command.

```
solid region density  $RHO_solids
```

#### 1.6.2.7.Solid Region Thermal Conductivity

This command specifies the thermal conductive for solid regions.

```
solid region thermal conductivity  $K_solids
```

#### 1.6.2.8.Solid Region Specific Heat

This command specifies the specific heat for solid region.

```
solid region specific heat  $CP_solids
```

### 1.6.3. Turbulence Models

#### 1.6.3.1.Enforce Production Equals Dissipation

```
enforce production equals dissipation
```

#### 1.6.3.2.Turbulent Flow High Reynolds Number k-ε Turbulence model

This command specified the use of the industry standard k-ε turbulence model.

```
turbulent flow high reynolds number k epsilon
```

#### 1.6.3.3.Turbulent Flow Low Reynolds Number k-ε Turbulence model

This command specified the use of a low Reynolds number version of the k-ε turbulence model (Chien).

```
turbulent flow low reynolds number k epsilon
```

#### 1.6.3.4.Turbulent Flow High Reynolds Number q-ω Turbulence model

This command specified the use of the q- ω turbulence model.

```
turbulent flow high reynolds number q omega
```

#### 1.6.3.5.Turbulent Flow Low Reynolds Number q- ω Turbulence model

This command specified the use of a low Reynolds number version of the q-ω turbulence model (Coakley).

```
turbulent flow low reynolds number q omega
```

### 1.6.4. Multiphase Models

#### 1.6.4.1. Constant Field Characteristic Diameter

The constant field characteristic diameter command is used to specify a single effective diameter for each field. This diameter is used in multiphase models such as drag, wall force and other models. It is ignored except for initialization when coalescence or breakup are implemented or if interfacial area transport is employed

```
constant field characteristic diameter  $D_field1, $D_field1,...,$D_fieldN
```

NOTE: some fields (i.e., continuous liquid field) may not use the characteristic diameter, but an input is required.

#### 1.6.4.2.Number of Fields

This command is used to specify the total number of fields. In a two-fluid ensembled averaged model, the fields can either represent different phases, of various forms of the same phase (i.e., small bubbles, large bubbles, taylor bubbles or continuous vapor).

```
number of fields  $NFIELDS
```

#### 1.6.4.3.Interfacial Drag Force

The interfacial drag force command is used to specify either a user defined or built-in model between any number of fields. At least one interfacial drag model is typically used for each field, however, none are requird and more than one is allowed. The form of the interfacial drag command is:

```
interfacial drag model  $NCARD
$FIELDA,$FIELDB,$MODELID,$USERMULTIPLIER // for first drag model, repeat  NCARD times for other drag models
```

The input for the drag force command is defined as:
$NCARDS - Number of input lines immediately following command line. Each line will specify a drag relation between fields.
$FIELDA - First Field for drag model
$FIELDB -Second field for drag model
$MODELID - Specifies Drag Model
0 - User Defined Drag Model (Use "user_drag.c" routine to define drag modeling)
1 - standard bubbly flow drag, $C_D$=0.44
2 – standard bubbly flow drag, $C_D$=0.44 with virtual mass force intergrated
3 – particle drag (solid spehere)
4 – seawater bubble drag
5 – fresh water bubble drag
6 – contaminated fresh water bubble drag
$USERMULTIPLIER - User specified multiplier (To modify drag coefficient, default=1.0)

#### 1.6.4.4. Interfacial Non Drag Force

The interfacial nondrag force command is used to specify either a user defined or built-in model between any number of fields. The form of the interfacial nondrag command is:

| |
|---|
| interfacial nondrag model $NCARD |
| $FIELDA,$FIELDB,$MODELID,$MODELSUBID,$USERMULTIPLIER // for first nondrag model, repeat NCARD times for other nondrag models |

The input for the nondrag force command is defined as:
$NCARDS - Number of input lines immediately following command line. Each line will specify a drag relation between fields.
$FIELDA - First Field for drag model
$FIELDB -Second field for drag model
$MODELID - Specifies Non Drag Force Model
0 - user specified model
1 - Lift Force Non-drag Model
2 - Volume Fraction Dispersion Non-drag Model
3 - Wall Force Non-drag Model
4 - Turbulence Non-drag Model
5 - Virtual Mass Force Non-drag Model
$MODELSUBID - Specifies subid for Non Drag Force Model
$USERMULTIPLIER - User specified multiplier (To modify non-drag coefficient, default=1.0)

#### 1.6.4.5. Wall Shear Apportionment Model

The wall shear apportionment command is used to specify the fraction of wall shear associated with each field. Since, in many multiphase applications, the wall adjacent nodes may contain multiple fields, the fields in contact with the wall should be assigned the wall shear. The default for the code is the wall shear is apportioned by the local volume fraction in the wall adjacent nodes. In many applications (i.e., bubble flows, annular flows, ...) the user may assign all or a fraction of the wall shear to a particular field (or fields) with this command.

| |
|---|
| wall shear apportionment model $FRACT_field1, $FRACT_field2,...,$FRACT_fieldN |

Where, $FRACT_field is the fraction of wall shear associated with each field. The sum of all $FRACT_field must add to one inorder to account for the wall shear.

#### 1.6.4.6. Bubble Cluster Drag Modification

Specifies that the drag coefficient is modified according to equation **Error! Reference source not found.**.

| |
|---|
| Bubble cluster drag modification |

#### 1.6.4.7.Interfacial Area Transport Model for Each Field

Specifies the interfacial area transport model for each field.

| |
|---|
| interfacial area transport model for each field $iaint_field0, $iaint_field1, …, $iaint_fieldN |

#### 1.6.4.8. Interfacial Area Coalescence Model for Each Field

Specifies the interfacial area coalescence model for each field.

| |
|---|
| interfacial area coalescence model for each field $coalescence_aint_model_field0, $coalescence_aint_model_field1,…, $coalescence_aint_model_fieldN |

#### 1.6.4.9. Interfacial Area Breakup Model for Each Field

Specifies the interfacial area breakup model for each field.

| |
|---|
| interfacial area breakup model for each field $breakup_aint_model_field0, $breakup_aint_model_field1,…, $breakup_aint_model_fieldN |

#### 1.6.4.10. Initialize Interfacial Area Using Volume Fraction and Characteristic Length

Specifies that the initial interfacial area density is defined using the local field volume fraction and characteristic length, not by “initialize ai field”

| |
|---|
| initialize interfacial area using volume fraction and characteristic length |

#### 1.6.4.11. Interfacial Area Feeds Back Into Bubble Diameter Frequency

Specifies the iteration frequency with which field bubble diameters are recomputed based on interfacial area density.

```
interfacial area feeds back into bubble diameter frequency $aint_db_feedback
```

#### 1.6.4.12. Employ Rhie Chow for Dispersion Terms

Specifies that the Rhie-Chow artificial dissipation parameter **Error! Reference source not found.** includes the turbulence dispersion model.

```
employ rhie chow for dispersion terms
```

#### 1.6.4.13.

### Heat Transfer Models

#### 1.6.4.14. Single Phase Heating

The use of this command will include the solution of the energy equation in the run. This command will allow only single phase heating (i.e., no mass transfer) in the analysis.

```
single phase heating
```

#### 1.6.4.15. Adiabatic Flow

The adiabatic flow command is used to specify mass and momentum conservation are only needed for the NPHASE code.

```
adiabatic flow
```

#### 1.6.4.16. Constant Fluid Reference Enthalpy

The constant fluid reference enthalpy command is used to specify a reference enthalpy for use in heat transfer models..

```
constant fluid reference enthalpy  $HREF_field1, $HREF_field1,...,$HREF_fieldN
```

#### 1.6.4.17. Constant Fluid Reference Temperature

The constant fluid reference temperature command is used to specify a reference temperature for use in heat transfer models.

```
constant fluid reference temperature  $TREF_field1, $TREF_field1,...,$TREF_fieldN
```

### 1.6.5. Mass Transfer Models

#### 1.6.5.1.Breakup sink for carrier field turbulence

```
breakup sink for carrier field turbulence
```

#### 1.6.5.2.InterField Mass Transfer Models

This command can be used to specify a mass transfer model. The form of the interfacial drag command is:

```
interfield mass transfer models $NCARD
$FIELDA,$FIELDB,$CARRIER,$MODELID,$USERMULTIPLIER  // for first  model, repeat NCARD times for other  models
```

The input for the drag force command is defined as:
$NCARDS - Number of input lines immediately following command line. Each line will specify a drag relation between fields.
$FIELDA - First Field for mass transfer model
$FIELDB - Second field for mass transfer model
$CARRIER - Carrier field for mass transfer
$MODELID - Specifies Drag Model
1 - Standard Mass Transfer Model
$USERMULTIPLIER - User specified multiplier (To modify mass transfer coefficient, default=1.0)

#### 1.6.5.3.Mass Diffusion Coefficient Laminar

mass diffusion coefficient laminar $MCOEF_field1, $MCOEF_field1,...,$MCOEF_fieldN

#### 1.6.5.4.Mass Diffusion coefficient Turbulent.

mass diffusion coefficient turbulent $MCOEF_field1, $MCOEF_field1,...,$MCOEF_fieldN

## 1.7. Additional Commands

allow deforming walls

anchor pressure

ausm artificial dissipation

boussinesq heating beta rho0t0 $BETA_BOUSS, $RHO_REF_BOUSS, $T_REF_BOUSS

build patran neutral file

clip species fraction stolie between zero and one inclusive

do not clip species fraction stolie between zero and one inclusive

clip volume fraction stolie between zero and one inclusive

do no tclip volume fraction stolie between zero and one inclusive

compute wall proximities

dont compute wall proximities

bubbly drag model $IP1, $IP2

constant fluid shear modulus $SHRMOD_field1, $SHRMOD_field1,...,$SHRMOD_fieldN

constant fluid reference density $RHOREF_field1, $RHOREF_field1,...,$RHOREF_fieldN

dimensionality offield view output $FV_OUT_DIM

continuous field0 turbulence drives all other fields

des modifications

dont perform wall match logic

drag and mass transfer in pseudo time step specification

drag and mass transfer not in pseudo time step specification

do not employ rhie chow for dispersion terms

continuity error treatment for turbulence

continuity error treatment for volume fraction

continuity error treatment for species

deforming region specification mid density modulus poisson $IMID, $L_REG_DEFORM, $R_REG_DEFORM, $E_REG_DEFORM, $N_REG_DEFORM

deforming wall specification face id mid modulus poisson thickness $IMID, $L_WALL_DEFORM, $R_WALL_DEFORM, $E_WALL_DEFORM, $N_WALL_DEFORM

deforming grid iteration frequency $ITER_FREQ

des model control fsst flag $FSST_FLAG

des model control c des $CDES

des model control t scale des $TSCALE_DES

des model control sigma x des $SIGMAX_DES

des model control ch1 des $CH1_DES

des model control ch2 des $CH2_DES

des model control ch3 des $CH3_DES

breakup model mu $BREAKUP_MULT

coalescence model mu $C_MULT

| | |
|---|---|
| coalescence model surfactant | $C_SURFACTANT |
| coalescence model initial film thi | $TFILM_0 |
| coalescence model final film thi | $TFILM_FINAL |
| cfl number mear | $CFL |
| cfl number tutrb | $CFLT |
| enforce zero radial and tangential velocities for post processing | |
| execute front and back ends only | |
| employ froude damping | |
| employ dirt lib over set technology | |
| employ grid adaption | |
| exclude continuous field from carver volume fraction normalization | |
| exclude continuous field from carver | |
| farfield grid smoothing | |
| farfield patch species mass fractions | $VAR |
| farfield patch species volume fractions | $VAR |
| false time step for u | $FTU_field1, $FTU_field1,...,$FTU_fieldN |
| false time step for v | $FTV_field1, $FTV_field1,...,$FTV_fieldN |
| false time step for w | $FTW_field1, $FTW_field1,...,$FTW_fieldN |
| false time step for a | $FTA_field1, $FTA_field1,...,$FTA_fieldN |
| false time step for h | $FTH_field1, $FTH_field1,...,$FTH_fieldN |

| | |
|---|---|
| false time step for k | $FTK_field1, $FTK_field1,...,$FTK_fieldN |
| false time step for uu | $FTUU_field1, $FTUU_field1,...,$FTUU_fieldN |
| false time step for vv | $FTVV_field1, $FTVV_field1,...,$FTVV_fieldN |
| false time step for ww | $FTWW_field1, $FTWW_field1,...,$FTWW_fieldN |
| false time step for uv | $FTUV_field1, $FTUV_field1,...,$FTUV_fieldN |
| false time step for vw | $FTVW_field1, $FTVW_field1,...,$FTVW_fieldN |
| false time step for wu | $FTWU_field1, $FTWU_field1,...,$FTWU_fieldN |
| false time step for f | $FTF_field1, $FTF_field1,...,$FTF_fieldN |
| false time step for e | $FTE_field1, $FTE_field1,...,$FTE_fieldN |
| fluid rheology | $FRHE_field1, $FRHE_field1,...,$FRHE_fieldN |
| employ mixture mass for pressure corrector | |
| employ mixture volume for pressure corrector | |
| employ cpe continuity coupling | |
| employ symmetric form of modified pressure | |
| employ thermodynamic data fits | $THFIT_field1, $THFIT_field1,...,$THFIT_fieldN |
| function entry/exit echo on | |
| function entry/exit echo off | |
| employ leonard exact diffusion term | |
| employ gibson modeling | |
| employ crusader modeli | $CRU_ELECTRHX_EFF, $CRU_TRANSOILHX_EFF, $CRU_ENGINEHX_EFF, $CRU_ELECTRHX_LOSS, |

| |
|---|
| CRU_TRANSOILHX,LOSS, $CRU_ENGINEHX_LOSS |
| employ model 5415 modeling |
| employ merkle deutsch flate plate modeling |
| employ hiplate modeling |
| employ darpa 12 inch modeling |
| employ large flat plate modeling |
| employ sub off modeling |
| employ asds modeling |
| employ uwx cavitator modeling |
| employ mruuv modeling |
| employ nrc case modeling |
| employ elgho bashi modeling |
| employ bubble rise modeling |
| employ meghan modeling $IMEGHAN |
| employ bistline coding $UCUR, $VCUT, $WCUR, $OXYCUR, $OYZCUR, $OZXCUR, $XCENTCUR, $YCENTCUR, $ZCENTCUR |
| employ haworth lungmodeling |
| employ visitor center modeling |
| employ cfd ship nphase bubble transport procedure |
| gibson modeling starting timestep |

helicity filter $HELICITY_FILTER

helicity smoothing $HELICITY_SMOOTH

ensight gid not written

flow velocity initialization using stringer

hard coded pressure distributions

hard coded inlet

hard coded inlet swirl for cyclic verification

field1 constant molecular viscosity $VISMC

field1 constant density $RHOC

field1 constant surface tension $SIGMAC

free surface model $NFREESURF_surface1,
NFREESURF_surface2,…,NFREESURF_surfaceN

homogeneous gas mixture fields

homogeneous gas mixture field parameters $JANNAFOPT

homogeneous incompressible mixture fields

homogeneous incompressible mixture field parameters $VAR1, $VAR2

initialize u
field $u_field0, $u_field1,...,$u_fieldN

initialize v field $v_field0, $v_field1,...,$v_fieldN

initialize w
field $w_field0, $w_field1,...,$w_fieldN

initialize a
field $a_field0, $a_field1,...,$wa_fieldN

```
initialize ai field     $ai_field0, $ai_field1,...,$ai_fieldN

initialize species mass fractions $VAR1, $VAR2

initialize species volume
fractions                         $VAR1

initialize
k field    $K_field0, $K_field1,...,$K_fieldN

initialize e field  $E_field0, $E_field1,...,$E_fieldN

initialize uu field $UU_field0, $UU_field1,...,$UU_fieldN

initialize vv field $VV_field0, $VV_field1,...,$VV_fieldN

initialize ww
field               $WW_field0, $WW_field1,...,$WW_fieldN

initialize uv field $UV_field0, $UV_field1,...,$UV_fieldN

initialize vw
field               $VW_field0, $VW_field1,...,$VW_fieldN

initialize wu
field               $WU_field0, $WU_field1,...,$WU_fieldN

initialize f
field           $F_field0, $F_field1,...,$F_fieldN

initialize pressure using s req

initialize with user initialize field
routine                           $VAR1

inlet grid smoothing

inlet patch species mass
fractions                         $MFRAC_N

inlet patch species volume
fractions                         $VFRAC_N
```

| | |
|---|---|
| interior wall patch | $WPATCH_field0, $WPATCH_field1,...,$WPATCH_fieldN |

| | |
|---|---|
| intrinsic swirl filter | $ISWRL_FILTER |

| | |
|---|---|
| intrinsic swirl smoothing | $ISWRL_SMOOTH |

| |
|---|
| ignore z direction indes delta computation |

| |
|---|
| initial velocities specified in cylindrical coordinates |

| |
|---|
| inlet profiles two dimensional specified in inflow.pro |

| |
|---|
| inlet species profiles specified in inflow.pro |

| |
|---|
| inlet species profiles 2d specified in inflow.pro |

| |
|---|
| interfacial area does not feedback into bubble diameter |

| | |
|---|---|
| interfacial area coalescence model for each field | $coales_aint_model_field0, $coales_aint_model_field1,…, $coales_aint_model_fieldN |

| | |
|---|---|
| interfield drag models | $drag_fielda, $drag_fieldb, $drag_modelid, $drag_usermultiplier |

| | |
|---|---|
| interfield non drag models | $nondrag_fielda, $nondrag_fieldb, $nondrag_modelid, $nondrag_usermultiplier, $nondrag_coeff_id2, |

| | |
|---|---|
| interpolation scheme for face values | $FACEINTERP |

| | |
|---|---|
| incorporate interior wall interface forces | $INTERFACE_FORCES |

| |
|---|
| interpret stringer coordinates as cylindrical |

| |
|---|
| impose wall bubble diameter kinematic constraint |

| |
|---|
| immersed boundaries |

| | |
|---|---|
| kumar bin partitioning | $KUMAR_MOM1, $KUMAR_MOM2 |

| | |
|---|---|
| kumar bin characteristic diameter | $KUMAR_DIA |

kumar bin characteristic volume $KUMAR_VOL

kumar breakup model $KUMAR_BREAKUP_MODELID, $KUMAR_BMULT

kumar coalescence model $KUMAR_COAL_MODELID, $KUMAR_CMULT

lake evaporation model $EVAP_MODELID, $EVAP_FACEID

limit dissipation in production term $EPS_PROD_MAX

limit dissipation in kumar breakup model $EPS_BREAKUP_MAX,
$LIMIT_EPS_BREAKUP

limit er form ean flow $LIMIT

limit er for turbulence $LIMIT2EQ

inviscid flux scheme $IFLX

pet sc specify solver iterations $IPETSC_ITER

pet sc specify solver tolerance $IPETSC_TOL

pet sc specify solver $IPETSC_SOLVR

pet sc specify preconditioner $IPETSC_PRECON

pet sc print norm $IPETSC_PRINTN

print t min and t max

modified production kato launder

moving grid

moving grid read frequency $GMR_FREQ

moving grid read grids

moving grid compute grids

moving grid compute grids prescribed motion

multiple frames of reference

pad vertex coordinates

neglect off diagonal terms in pressure poisson equation

parallel strategy for pressure corrector: matrix level

parallel strategy for pressure corrector: explicit partition boundary update

pressures at pressure boundaries specified transiently

perform agglomeration operation

print linear solver residual severys weep

print linear solver residuals after final sweep

overwrite density by mid $RHOC_MID

overwrite molecular viscosity by mid $VISMC_MID

overwrite gas molecular viscosities $VISGAS_field1, $VISGAS_field2,…,
$VISGAS_fieldN

produce viewable perprocessor ensight output files

produce ensight files with partition boundaries

produce ensight files without partition boundaries

produce ensight files with i blanked cells

produce ensight files without j blanked cells

produce ensight output

produce ensight scalar files for ai

produce ensight scalar files for a

produce ensight scalar files for temperature

produce ensight scalar files for h

produce ensigh tscalar files for k

produce ensight scalar files for e

produce ensight scalar files for v2f parameters

produce ensight scalar files for reynolds stresses

produce ensight scalar files for eddy viscosity

produce ensight scalar files for density

produce ensight scalar files for helicity

produce ensight scalar files for intrinsic swirl

produce ensight scalar files for debug var0

produce ensight scalar files for debug var1

produce ensight scalar files for debug var2

produce ensight scalar files for debug var3

produce ensight scalar files for debug var4

produce ensight scalar files for debug var5

produce ensight scalar files for debug var6

produce ensight scalar files for debug var7

produce ensight scalar files for debug var8

produce ensight scalar files for debug var9

produce ensight scalar files for species mass fractions

produce ensight scalar files for i blank

produce fieldview output

produce data explorer output

produce wall boundary layer output

pressure gradients req stream sheet correction

production destruction ratio clip $PRODCLIP

perfect gas compressibility parameters $RGAS_field1, $RGAS_field2,…$RGAS_fieldN
$SPRATIO_field1, SPRATIO_field2,…, SPRATIO_fieldN

calorically perfect gas compressibility parameters $RGAS_field1, $RGAS_field2,
…$RGAS_fieldN
$SPRATIO_field1, SPRATIO_field2,…, SPRATIO_fieldN

print pressure boundary reverse flow information

print outlet boundary reverse flow information

print cylindrical coordinate patchfiles

print fieldview in cylindrical coordinates

print pressure patch profile for inflow profile

pressure boundary use ambient velocities for inflow

pressure boundary use extrapolated velocities for inflow

prandtl number laminar $PRDTLL_field1, $PRDTLL_field2,…, $PRDTLL_fieldN

prandtl number turbulent $PRDTLT_field1, $PRDTLT_field2,…, $PRDTLT_fieldN

mass diffusion coefficient species laminar $LMASSDIFF_field1,
$LMASSDIFF_field2,…, $LMASSDIFF_fieldN

mass diffusion coefficient species turbulent $TMASSDIFF_field1,
$TMASSDIFF_field2,…, $TMASSDIFF_fieldN

porous wall patch species mass fractions

porous wall patch species volume fractions

pressure grid smoothing

pressure patch species mass fractions

pressure patch species volume fractions

pressure profile patch

pid attributes

number of additional species $NPHI

order of accuracy of inviscid terms mean flow

order of accuracy of inviscid terms turbulence

mass transfer model cprod rprod c destr dest $CPROD, $RPROD, $CDEST, $RDEST

mass transfer model flag $MTMODEL

mass transfer model linearization flag $ILMTRANS

natural cavitatinonumber $CAVNUM

pre conditioner flag $IPRECON

pre conditioning parameter beta $BETA

minimum number of sgs sweeps $ISGSMIN

maximum number of sgs sweeps $ISGSMAX

number of sgs sweeps $ISWEEP

solver choice for velocity components jacobi

solver choice for velocity components jacobi uvw

solver choice for velocity components petsc

solver choice for pressure amggs $SCPRESS_field1, $SCPRESS_field2,..., $SCPRESS_fieldN

solver choice for pressure amgilu $SCPRESS_field1, $SCPRESS_field2,..., $SCPRESS_fieldN

use directional coarsening $ISOCOARSE_field1, $ISOCOARSE_field2,...,

$ISOCOARSE_fieldN

use direct solve on coarsest grid $DIRSOLVE_field1, $DIRSOLVE_field2,..., $DIRSOLVE_fieldN

solver choice for pressure block correction $NIN, $NJN, $NKN, $PCORR_field1, $PCORR_field2,..., $PCORR_fieldN

solver choice for pressure petsc

solve mixture momentum mass centered mixture velocity

solve mixture momentum volume centered mixture velocity

read wall temperature data from file

read wall match data from file

read wall proximity from file

restart with field0 frozen

restart with pressure frozen

solver monitor iteration $ITER_MON

solver choice for turbulence scalars petsc

solver choice for all scalars ilu

solver choice for enthalpy jacobi

solver choice for enthalpy petsc

relaxation factor for a $RFA

relaxation factor for ai $RFAI

relaxation factor for s $RFS

relaxation factor for uu $RFUU

relaxation factor for vv $RFVV

relaxation factor for ww $RFWW

relaxation factor for uv $RFUV

relaxation factor for vw $RFVW

relaxation factor for wu $RFWU

relaxation factor for f $RFF

relaxation factor for p $RFP

rhie chow multiplier $RHEICHOW_MULT_field1, $RHEICHOW_MULT_field2,..., $RHEICHOW_MULT_fieldN

surface roughness height $SURF_HEIGHT

residual print file not written

restart files not written

transient file write frequency $RESTART_FREQ

solver sweeps for k $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN

solver sweeps for e $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN

solver sweeps for species $NSWEEP_field1, $NSWEEP_field1,...,$NSWEEP_fieldN

rectilinear grid

second order viscous bcs for hexs and prisms

solve energy equation only

solve energy equation with frozen flow field

solve species equation only

solve species equation with frozen flow field

relaxation factor applied to linear solver only

thin layer approximation employed for momentum

thin layer approximation not employed for momentum

thin layer approximation employed for scalars

thin layer approximation not employed for scalars

spatial discretization interfacial area density $ISPATIAL

spatial discretization species equation $ISPATIAL

spatial discretization density $ISPATIAL

spatial discretization enthalpy $ISPATIAL

temporal discretization interfacial area density $ITEMPOR

temporal discretization species equation $ITEMPOR

temporal discretization enthalpy $ITEMPOR

temporal discretization grid $ITEMPOR

temporal discretization mass $ITEMPOR

temporal discretization $ITEMPOR

use absolute velocities as dependent variables

set pressure correction to zero on partition boundaries

set pressure correction gradient to zero on partition boundaries

use block correction to update pressure correction on partitions

turbulent flow menter k omega

turbulent flow menter k omega sst

turbulent flow goldber gkr

turbulent flow v2f

turbulent flow frsm

turbulent flow wilcox komega

turbulence model for each field

turbulence model reynolds number regime for each field

strongly coupled compressibility

update enthalpy during pressure corrector

update species during pressure corrector

use ficks law form for mass diffusion

use scalar relaxation for field coupling

use block relaxation for field coupling

use scalar relaxation for pc and rc coefficients

use block relaxation for pc and rc coefficients

symmetry grid smoothing

symmetry patch

smooth volume fraction in backend $VSMOOTH

solid region specification mid density conductivity cp $RSOLID, $KSOLID, $CSOLID

sgs parameters is chkgs tolsor fact $ISCHK, $GSTOL, $SORFACT

z test, no input yet

weakly coupled compressibility

wall grid smoothing

### *Software Delivery and Installation Summary*

The gzipped tar file with which the NPHASE-PSU software is delivered unpacks into the following UNIX directory structure:

**tar –xzvf nphase-*****.tar.gz**

setup (*bash script to set up UNIX environment for NPHASE-PSU*)

petsc-*xxxxx (where xxxxx is the latest version of PETSC delivered with the code, tar install file)*

**Directories:**

srcbase

FUMP

EMERGE

METIS

SORT_CYCLIC

SUGGAR_DIRT

TUTORIAL_1

TUTORIAL_HIPLATE (not included in the NGRC distribution)

TUTORIAL_5415 (not included in the NGRC distribition)

TUTORIAL_MACH_BUMP

TUTORIAL_GEAR

### *Running Tutorials on a System where NPHASE is Already Installed*

In this case delete the all of the directories except the tutorial directories, copy the nphase executable, **nphase**, into each of the tutorial directories, and set the following aliases:

**alias fump '${NPHASE-PSU_PATH} /FUMP/fump'**
**alias emergetrans '${NPHASE-PSU_PATH} /EMERGE/emergetrans'**
**alias emerge '${NPHASE-PSU_PATH}/EMERGE/emerge'**
**alias sort_cyclic '${NPHASE-PSU_PATH}/SORT_CYCLIC/sort_cyclic'**

with the path, ${NPHASE-PSU_PATH} set accordingly. These utilities each need to be recompiled as follows:

1) In directory METIS: tar -zxvf metis-4.0.3.tar.gz; cd metis-4.0.3; make
2) In directory FUMP: make
3) In directory EMERGE: make –f makeem ; make –f makeemt
4) In directory SORT_CYCLIC: make

### *Installing NPHASE-PSU software and Running Tutorials*

The delivered NPHASE-PSU source code resides in the directory srcbase. The code unpacks with numerous source files (*.c), include files (*.h), files used for installation (makefile.linux, Obj_nphase) in that directory.

To compile NPHASE-PSU, edit the file "setup", to point the installation to where the user wishes NPHASE-PSU to reside on their system. On line 2, modify the environment variable $NPHASE_OBJ set to ${NPHASE-PSU_PATH}/build. The same modifications need to be carried out for PETSC_DIR, DIRTLIB_HOME and P3DLIB_HOME and MPI_HOME. The variable PETSC_ARCH should be set to reflect the compiler architecture used. If using Intel compilers and MPI, the four source /software/intel/*** lines should be changed to reflect the location of the users' Intel compiler installation, and the same should be done for LD_LIBRARY_PATH variables. If not using Intel compilers and MPI, the four source statements are unnecessary, and can be removed.

The "setup" script should be called at login whenever NPHASE-PSU is to be used to ensure environment variables are set appropriately.

Compilation of NPHASE-PSU depends upon DiRTlib and P3Dlib. To compile these, after making appropriate modification to and calling the setup script, perform the following: cd ${P3DLIB_HOME}; ../configure --prefix=${P3DLIB_HOME}; make; make install. Then: cd ${DIRTLIB_HOME}; ../configure --prefix=${DIRTLIB_HOME} --with-p3d=${P3DLIB_HOME} --with-mpi=${MPI_HOME}; make; make install. Compilation of DiRTlib with a specified ${MPI_HOME} directory requires that it contain ${MPI_HOME}/bin, lib, include. If using Intel MPI in 64 bit mode, the directory ${I_MPI_ROOT} will contain bin64, lib64, include64 subdirectories. To get DiRTlib to compile correctly, simply create a dummy directory containing symbolic links bin, lib, include that point to these, and point ${MPI_HOME} to this dummy directory within the setup script. One caveat to this is that PETSc should be compiled by directing it to the actual installation of Intel MPI, as opposed to the dummy one. The procedure for accomplishing this is available on the PETSc website.

Upon appropriate modification to the setup script, and successful installation of PETSc, DiRTlib, and P3Dlib, the user can now compile NPHASE-PSU by:

cd ${NPHASE-PSU_PATH}/srcbase

**make –f makefile.linux –j n**

(where n is the number of processors on the compile node to be used for compilation [typically the number of processors on the head node of a cluster]).

The makefile automatically links to appropriately compiled PETSC libraries (PETSC is the open source linear solver library used by NPHASE) and "DIRTLib" libraries (DIRTLib is the open source overset meshing library used by NPHASE), compiles all of the objects and generates the final executable: nphase. To run the tutorials, copy the nphase executable into each of the tutorial directories.

## References


1. Batchelor, The Theory of Homogeneous Turbulence, Cambridge University Press, 1956.
2. Carrica, P. M., D. Drew, F. Bonetto, R. T. Lahey, Jr., “A Polydisperse Model for Bubbly Two-phase Flow around a Surface Ship,” *International Journal of Multiphase Flow*, Vol. 25, 1999, pp. 257-305.
3. Clift, S. S., Forsyth, P. A., “Linear and Non-linear and Non-linear Methods for the Incompressible Navier-Stokes Equations,” International Journal for Numerical Methods in Fluids, Vol. 18, 1994, pp. 229-256.
4. Coulaloglou, C. A., Tavlarides, L.L, “Description of Interaction Processes in Agitated Liquid-Liquid Dispersions,” *Chemical Engineering Science*, Vol. 32, 1977, p. 1289.
5. Detsch, R.M., “Small Air Bubbles in Reagent Grade Water and Seawater: 1. Rise Velocities of 20-1000-µm-Diameter Bubbles”, *Journal of Geophyscial Research - Oceans*, Vol. 96, No. 5, 1991, p. 8901.
6. Diab, Y., Ville, F., Velex, P., and Changenet, C., 2004, “Windage Losses in High Speed Gears—Preliminary Experimental and Theoretical Results,” *ASME Journal of Mechanical Design*, 126(5), pp. 903–908.
7. Durbin, P.A., “Near-Wall Turbulence Closure Modeling Without Damping Functions,” *Theoretical and Computational Fluid Dynamics*, Vol. 3, 1991, pp. 1-13.
8. ENSIGHT online documentation, http://www.ensight.com/ensight.html, 2006.
9. Ferrante, A., Elghobashi, S., “On the Physical Mechanisms of Drag Reduction in a Spatially Developing Turbulent Boundary Layer Laden with Microbubbles,” *Journal of Fluid Mechanics*, Vol. 503, 2004, pp. 345-355.
10. Ferrante, A., Elghobashi, S., "Reynolds Number Effects on Drag Reduction in a Bubble-Laden Spatially Developing Turbulent Boundary Layer Over a Flat Plate,” Proceedings of the 2nd International Symposium on Seawater Drag Reduction, Busan, Korea, May, 2005.
11. HARPOON online documentation, http://www.ensight.com/harpoon.html, 2006
12. Hibiki, T., T. Takamasa, M. Ishii: “Interfacial Area Transport of Bubbly Flow in a Small Diameter Pipe,” *Journal of Nuclear Science and Technology*, Vol. 38, No. 8, 2001, pp. 614-620.
13. Hill, M.J., Kunz, R.F., Noack, R.W., Long, L.N., Morris, P.J., Handschuh, R.F., ”CFD Technology for Rotorcraft Gearbox Windage Aerodynamics Simulations,” Gear Technology, 48-55, August, 2009.
14. Hill, M.J., Kunz, R.F., Medvitz, R.B., Handschuh, R.F., Long, L.N., Noack, R.W., Morris, P.J. “CFD Analysis of Gear Windage Losses: Validation and Parametric

Aerodynamic Studies," *ASME Journal of Fluids Engineering*, 133:031103-1-10, March 2011.

15. Hinze, J. O., Turbulence, 2nd edition, McGraw-Hill, 1975.
16. ICEM-CFD online documentation, http://www.ansys.com/products/icemcfd.asp, 2006.
17. Johansen, S.T., Boysan, F., "Fluid Dynamics in Bubble Stirred Ladles: Part II. Mathematical Modeling," *Metallurgical Transactions B*, Vol. 19b, 1988, pp. 755-764.
18. Kawamura, T., Yoshiba, H., "Numerical Modelng of Bubble Distributions in Horizontal Bubbly Channel Flow," Proceedings of the 5th International Conference on Multiphase Flow, Yokohama, Japan, 2004, Paper No. 354.
19. Kumar, S., Ramkrishna, D. "On the Solution of Population Balance Equations by Discretization – I. A Fixed Pivot Technique," Chemical Engineering Science, Vol. 51, No. 2, 1996, pp. 1311-1332.
20. Kunz, R.F., Siebert, B.W., Cope, W.K., Foster, N.F., Antal, S.P., Ettorre, S.M. "A Coupled Phasic Exchange Algorithm for Multi-Dimensional Four-Field Analysis of Heated Flows with Mass Transfer," Computers and Fluids, Vol. 27, No. 7, 1998.
21. Kunz, R.F., Venkateswaran, S., "On the Roles of Implicitness, Realizability, Boundary Conditions and Artificial Dissipation in Multidimensional Two-Fluid Simulations with Interfacial Forces," AMIF-ESF Workshop on Computing Methods for Two-Phase Flow, Centre Paul Langevin, Aussois, France, 12-14 January, 2000.
22. Kunz, R.F., Yu, W.S., Antal, S.P., Ettorre, S.M., "An Unstructured Two-fluid Method Based on the Coupled Phasic Exchange Algorithm," AIAA Paper 2001-2672, 2001.
23. Kunz, R.F., Deutsch, S., Lindau, J.W., "Two Fluid Modeling Of Microbubble Turbulent Drag Reduction," ASME Paper FED2003-45640, Proceedings of FEDSM'03: 4TH ASME–JSME Joint Fluids Engineering Conference, Honolulu, Hawaii, USA, July 6–11, 2003.
24. Kunz, R.F., Gibeling, H.J., Maxey, M.R., Tryggvason, G., Fontaine, A.A., Petrie, H.L., Ceccio, S.L., "Validation of Two-Fluid Eulerian CFD Modeling for Microbubble Drag Reduction Across a Wide Range of Reynolds Numbers," Proceedings of the 2nd International Symposium on Seawater Drag Reduction, Busan, Korea, May, 2005, to appear *ASME Journal of Fluids Engineering*, January, 2007.
25. Kunz, R.F., Fontaine, A.A., Maxey, M.A., "Multiscale Physical Modeling for Microbubble Drag Reduction at High Reynolds Numbers," FINAL REPORT to Defense Advanced Research Projects Agency Friction Drag Reduction Program, September 2006.
26. Kunz, Hill, M.J., Kunz, R.F., Medvitz, R.B., Handschuh, R.F., Long, L.N., Noack, R.W., Morris, P.J. "CFD Analysis of Gear Windage Losses: Validation and

Parametric Aerodynamic Studies," *ASME Journal of Fluids Engineering*, 133:031103-1-10, March 2011.

27. Lahey, R.T., Drew, D.A., "An Analysis of Two-Phase Flow and Heat Transfer Using a Multidimensional, Multi-field, Two-Fluid Computational Fluid Dynamics (CFD) Model," Japan/US Seminar on Two-Phase Flow Dynamics, Santa Barbara, California, 2000.
28. Lehr, F., Mewes, D., "A Transport Equation for the Interfacial Area Density Applied to Bubble Columns," *Chemical Engineering Science*, Vol. 56, 2001, pp. 1159-1166.
29. Levich, V. G.: Physicochemical Hydrodynamics, Prentice-Hall, Englewood Cliffs, NJ, 1962.
30. Lien, F.S. "A Pressure-Based Unstructured Grid Method for All-Speed Flows," *International Journal for Numerical Methods in Fluids*, Vol. 33, 2000.
31. Lopez de Bertodano, M.: "Two-fluid Model for Two-phase Turbulent Jets," *Nuclear Engineering and Design*, Vol. 179, 1998, pp. 65-74.
32. Loth, E., "Numerical Approaches for Motion of Dispersed Particles, Droplets and Bubbles," *Progress in Energy and Combustion Science*, Vol. 26, 2000, pp. 161-223.
33. Martinez-Bazan, C., Montanes, J.L., Lasheras, J.C., "On the Breakup of an Air Bubble Injected into a Fully Developed Turbulent Flow. Part 1. Breakup Frequency," *Journal of Fluid Mechanics*, Vol. 401, 1999a, pp. 157-182.
34. Martinez-Bazan, C., Montanes, J.L., Lasheras, J.C., "On the Breakup of an Air Bubble Injected into a Fully Developed Turbulent Flow. Part 2. Size PDF of the Resulting Daughter Bubbles," *Journal of Fluid Mechanics*, Vol. 401, 1999b, pp. 183-207.
35. Maxey, M.R., Dong, S., Xu, J., Karniadakis, G.E., "Simulations for Microbubble Drag Reduction (MBDR) at High Reynolds Numbers", Proceedings of the HPCMP Challenge Project, Users Group Conference 2005, Nashville TN, June 27-30, 2005. (Published by IEEE Computer Society).
36. Meng, J.C.S., Uhlman, J.S., "Microbubble Formation and Splitting in a Turbulent Boundary Layer for Turbulence Reduction," Proceedings of the International Symposium on Seawater Drag Reduction, Newport, RI, July, 1998, pp. 341-355.
37. METIS on-line documentation, http://glaros.dtc.umn.edu/gkhome/views/metis, 2006.
38. Michaelides, E. E., Particles, Bubbles & Drops: Their Motion, Heat And Mass Transfer, World Scientific Publishing Company, April 2006.
39. MPICH on-line documentation, http://www-unix.mcs.anl.gov/mpi/mpich , 2006.
40. Noack, R. W., Boger, D. A., Kunz, R. F., and Carrica, P. M., "Suggar++: An Improved General Overset Grid Assembly Capability," AIAA Paper 2009-3992, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX, 22-25 June 2009.
41. PETSC documentation: http://www-unix.mcs.anl.gov/petsc

42. Prince, M.J., Blanch, H.W., “Bubble Coalescence and Breakup in Air-Sparged Bubble Columns,” *AIChE Journal*, Vol. 36, No. 10, October, 1990, pp. 1485-1499.

43. Rhie, C. M., Chow, W. L. (1983) “Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation,” *AIAA Journal*, Vol. 21, 1983, p. 1527.

44. Richardson, J. F., Zaki, W. N., “Sedimentation and Fluidization: Part I,” Transactions of the Institute of Chemical Engineering, Vol. 32, 1954, p. 35.

45. Sanders, W.C., Dowling, D.R., Perlin, M., Ceccio, S.L.,“Bubble Friction Drag Reduction in a High Reynolds Number Flat Plate Turbulent Boundary Layer,” *Journal of Fluid Mechanics*, Vol. 552, 2006, pp. 353-380

46. TECPLOT on-line documentation, http://www.tecplot.com , 2006.

47. Tryggvason, G., Lu, J. “DNS of Drag Reduction due to Bubble Injection into Turbulent Flow,” Proceedings of the 2nd International Symposium on Seawater Drag Reduction, Busan, Korea, May, 2005.

48. Van Doormal, J. P., Raithby, G. D., “Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows,” *Numerical Heat Transfer*, Vol. 7, 1984, pp. 147-163.

49. Venkateswaran, S., Tamamidis, P., Merkle, C.L., “Interpretation of Pressure-Based Methods as Time Marching Schemes,” Proceedings 15th International Conference on Numerical Methods in Fluid Dynamics, Springer Lecture Notes in Physics, 1997.

50. Williams, M.M.R., Loyalka, S.K., Aerosol Science Theory and Practice, Pergamon Press, Oxford,1991.