

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/318710028>

Special Topics in CFD

Technical Report · April 2021

CITATION

1

READS

7,612

1 author:



Ideen Sadrehaghighi

CFD Open Series

51 PUBLICATIONS 62 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Airfoil and Wing Shape Optimization [View project](#)

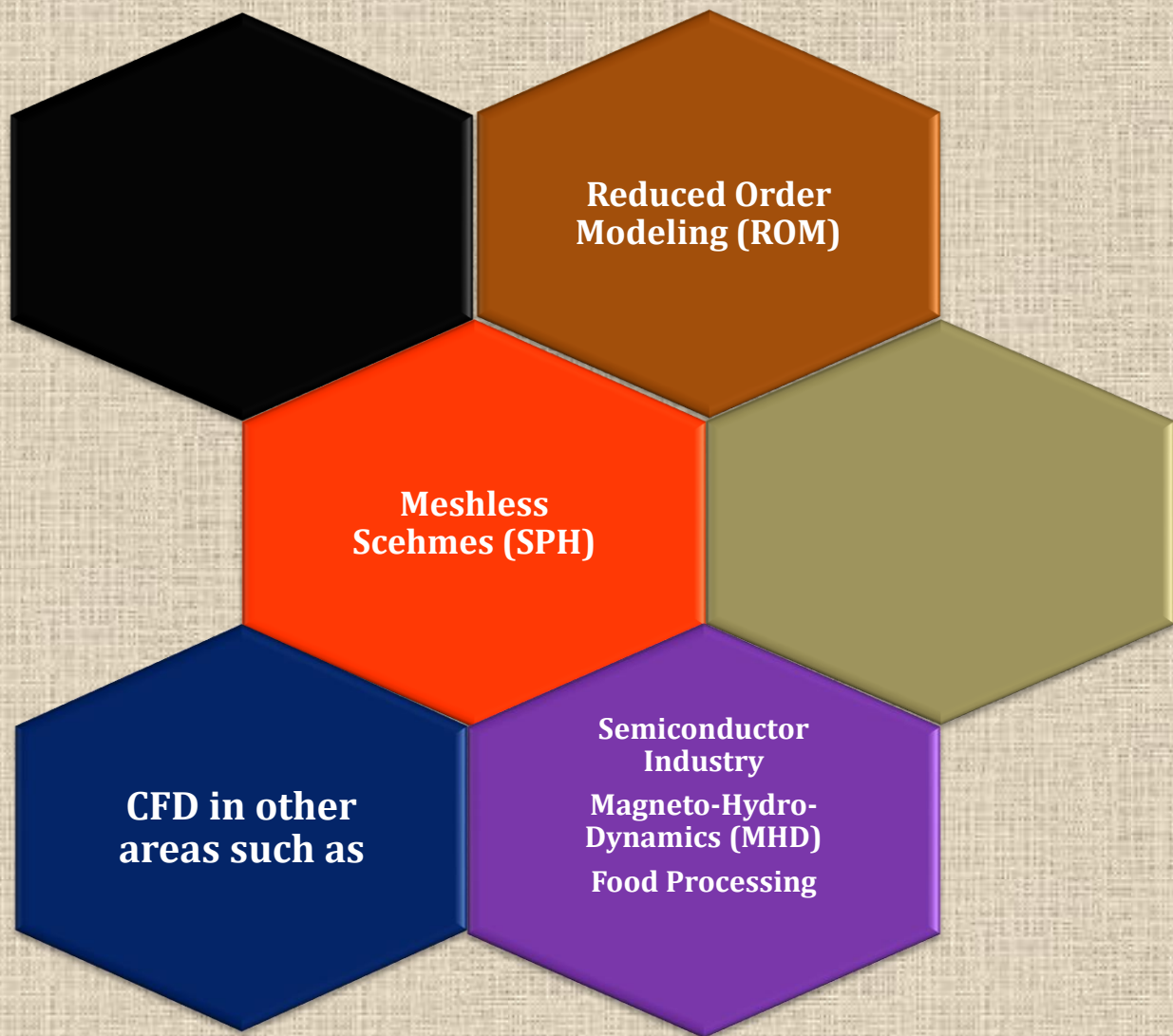
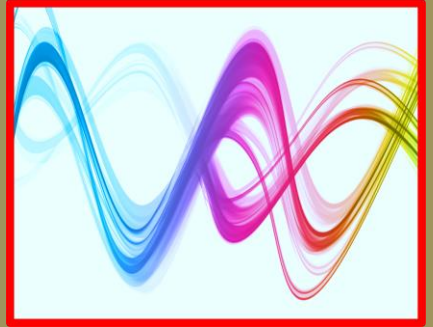


Mesh Generation [View project](#)

CFD Open Series
Patch: 2.20

Special Topics in CFD

Edited by:
Ideen Sadrehaghighi, Ph.D.



ANNAPOLIS, MD

Contents

1	Introduction.....	6
1.1	Preliminaries.....	6
1.2	Computational Predictively plus Verification & Validation.....	6
1.3	Multiscale/Multiphysics.....	7
1.4	Mesh Free Methods for CFD.....	7
1.5	Integrated Simulations of Complex Systems	8
1.5.1	Case Study - CAD Embedded CFD	8
2	Reduced Order Modeling (ROM)	10
2.1	Introduction	10
2.2	Various Techniques.....	11
2.3	Common Features Shared by Reduced Order Methods (ROM)	11
2.4	Reduced Basis Methods.....	12
2.4.1	Lagrange	12
2.4.2	Hermit.....	12
2.4.3	Taylor	12
2.4.4	Snapshot Sets Method.....	13
2.5	Proper Orthogonal Decomposition (POD) Spaces.....	13
2.5.1	Case Study 1 - 2D Laminar Separated Flow Over a Flat-Plate Wing.....	14
2.5.2	Case Study 2 - Cylinder Wakes	15
2.6	Galerkin Projection into POD Space	16
2.6.1	Case Study - Vortex Shedding Around a Circular Cylinder using a POD-Galerkin Method.....	17
2.6.1.1	Governing Equations	17
2.6.1.2	Details of the Full Order Simulation	18
2.6.1.3	Details of the ROM Simulation	18
2.6.1.4	Analysis of the Results.....	18
2.7	Addressing Challenges in Reduced-Order Modeling.....	20
2.8	Reduced Order CFD Simulation.....	21
2.8.1	Case Study 1 - Designing Parameters of Test Stage Axial Turbine	22
2.8.1.1	Blade Reverse Engineering as applied to Geometry Definition	23
2.8.1.2	3D Aerodynamic Computation	24
2.8.2	Case Study 2 - Cooling Air Flow Rate.....	24
2.9	Reduced Order Model Using Empirical Relationship.....	24
3	Mesh Free Methods for CFD.....	26
3.1	Smooth Particle Hydrodynamics (SPH).....	26
3.1.1	Mesh free Local Petrov-Galerkin	27
3.1.2	Mesh free Methods Based on Radial Basis Functions	28
3.1.3	Finite Point Methods	28
3.1.4	Meshless Boundary Schemes	29
3.2	Solution Procedure for Mesh free Methods	29
3.2.1	Domain representation	29
3.2.2	Function Approximation.....	29
3.2.3	Formation of System Equations	30
3.2.4	Solving the Global Equations	30

3.3	Method of Smooth Particle Hydrodynamics (SPH).....	30
3.3.1	Formulation	30
3.3.2	Smoothing Kernels.....	31
3.3.3	Updating of Smoothing Length h	32
3.3.3.1	Constant	33
3.3.3.2	Variable.....	33
3.3.4	Boundary Treatment	33
3.3.5	Virtual Particles.....	33
3.3.6	Ghost Particles.....	33
3.3.7	Summary and Recap	34
3.3.8	Case Study 1 - Lid Driven Cavity Problem	34
3.3.9	Case Study 2 - Two-dimensional Convection–Diffusion Problem	34
3.4	RKPM Method	35
3.5	Lagrangian Description of Fluid Dynamics Using SPH	35
3.5.1	Default Kernel.....	36
3.5.2	Numerical Time Integration.....	37
3.5.2.1	The Implicit Euler Scheme	37
3.5.2.2	The Verlet Scheme	38
3.5.2.3	The Leap-Frog Scheme	38
3.5.3	Collision Handling	38
3.5.4	Case Study 1 – Comparison of Weakly Compressible and Incompressible SPH.....	39
3.5.4.1	Formulation of Problem	39
3.5.4.2	Results	40
3.5.5	Case Study 2 - Dam Break Water Flow using Lagrangian Description.....	40
3.5.6	Case Study 3 - Dam Break using MLPG-RBF and Shallow Water Equations.....	41
3.5.7	Case Study 4 - SPH Method for Evaporating Multiphase Flows	42
3.5.7.1	Basic Formulations of the SPH Method.....	42
3.5.7.2	Evaporation of a Static Drop	43
3.5.7.3	Evaporation of a Dynamic Drop Impacting on a Hot Surface.....	43
3.5.7.4	Concluding Remarks.....	44
4	CFD Applications in Other Areas	45
4.1	Food Processing.....	45
4.1.1	Drying.....	45
4.1.2	Sterilization.....	46
4.1.3	Mixing	47
4.1.4	Refrigeration.....	48
4.1.5	Crystallization	48
4.1.6	Pasteurization.....	48
4.2	CFD in Semiconductor Industry.....	49
4.2.1	Brief Description of Semiconductor Devices	49
4.2.2	Thermal Management in Semiconductors	50
4.2.3	Can You Really Fry an Egg on a CPU?.....	50
4.3	Magneto-Hydro-Dynamics (MHD).....	51
4.3.1	MHD Equations.....	52
4.3.2	Case Study - Dynamics of a Q2D Wake Behind a Cylinder in Presence of MHD Environment	53
4.3.2.1	Numerical Method and Geometry	53

4.3.2.2	Result and Discussion	54
4.4	Maxwell's Equations - Electromagnetic Waves	55
4.4.1	Historical Perspective	55
4.4.2	The Finite-Difference Time-Domain Method (FDTD)	55
4.4.3	Strengths of FDTD Modeling.....	55
4.4.4	Weaknesses of FDTD Modeling	56
4.4.5	Case Study - 1D Maxwell Equation	57
4.4.5.1	Boundary Conditions	57
5	Appendix A	59
5.1	Routine for Inverse Distance Weighted Interpolation (Shepard's Method).....	59

List of Tables

Table 2.8.1	Main Parameters of the Test Stages (P1) and (P2).....	23
-------------	---	----

List of Figures

Figure 1.1.1	Current Research Area in CFD.....	6
Figure 1.5.1	Integrated Simulation for Nuclear Engineering.....	8
Figure 2.1.1	Interpolation on a matrix manifold.....	10
Figure 2.5.1	Modal Decomposition of 2D Incompressible Flow Over a Flat-Plate Wing ($Re = 100$ and $\alpha = 30$ deg). This example shows complex nonlinear separated flow being well represented by only two POD modes and the mean flow field. Visualized are the streamwise velocity profiles.....	15
Figure 2.5.2	POD Analysis of Cylinder Flow. (a) Original Flow Field (vorticity shown). (b) First 8 dominant POD modes	16
Figure 2.6.1	Comparison between velocity and pressure High Fidelity (HF)-ROM	19
Figure 2.6.2	Comparison of the Drag Coefficient Obtained with the High Fidelity (HF) and ROM Simulations.....	19
Figure 2.8.1	1D vs 3D Analysis.....	21
Figure 2.8.2	Turbine Flow Design Process	22
Figure 2.8.3	Profile of Blades.....	23
Figure 2.8.4	Typical Cooling System Network for Airflow Rate	24
Figure 3.2.1	Domain Representation	29
Figure 3.2.2	Different type of Support domains	30
Figure 3.3.1	1-D SPH Characterization.....	30
Figure 3.3.2	The choice of Different Smooth Kernel in 1D ($h=1$).....	32
Figure 3.3.3	Ghost Particles, Velocities are formed Symmetrically (slip wall).....	32
Figure 3.3.4	Virtual Particles	33
Figure 3.3.5	Example of a 1D task, particle j is Situated in the Near Boundary Area.....	33
Figure 3.3.6	Comparison with FDM with SPH for Lid Driven Cavity	34
Figure 3.3.7	The diagram of global domain Ω , local support domain Ω_s of point x_s , global points x and local point x_i	35
Figure 3.5.1	Lagrange particle-based fluid structure in 2D.....	36
Figure 3.5.2	The default kernel and its derivatives in one dimension for $h=1$	37
Figure 3.5.3	The leap-frog mechanism.....	38
Figure 3.5.4	Comparison of ISPH (upper), FEM (middle) and WCSPH (lower) velocity contours for the angle of attack of 15 degrees at $Re = 570$ (Courtesy of Shadloo ¹⁰⁵)	40
Figure 3.5.5	Dam-Break Flow of water	40
Figure 3.5.6	Geometry and Water surface profile of the 2D dam-break problem at $t = 7.2$ s.....	41

Figure 3.5.7	Snapshots of the Evaporating Drop at different times using SPH	43
Figure 3.5.8	Evolution of Dynamic Drop impact on a hot surface using SPH.....	44
Figure 4.2.1	Illustrates the various classes of conductors	49
Figure 4.2.2	Modern Semiconductor.....	50
Figure 4.2.3	Thermal Management of Semiconductor (courtesy of Mentor CFD)	50
Figure 4.2.4	An Example of an Egg Frying on a CPU	51
Figure 4.3.1	Right Hand Rule for MHD	53
Figure 4.3.2	Schematic Diagram of Numerical Domain.....	54
Figure 4.3.3	Contour plots of vorticity snapshot at $Re_d = 160$ and at Hartmann number as indicated.....	54
Figure 4.4.1	Illustration of a Standard Cartesian Yee cell used for FDTD for Electric and Magnetic Field.....	57



1 Introduction

1.1 Preliminaries

As evident in [Error! Reference source not found.](#) below, there is no shortage of active research area in CFD. Besides the regular on-going research in new algorithms, there are ever expanding of new activities, some mentioned here but not all. Some of the more prominent researches are shown in [Figure 1.1.1](#) and out of them, some defined below.



1.2 Computational Predictively plus Verification & Validation

This includes relatively well-defined tasks such as verification of the correctness of computer codes and uncertainty quantification as well as more hazy ones like validations of the model being used. As codes become more complex their verification becomes more challenging. Methods such as the

Method of Manufactured Solutions are one way. In its simplest form, uncertainty quantification is simply the propagation of uncertainties in parameters, properties and models to the final solution. Although conceptually simple, this is a formidable task both because we need to know all elementary uncertainties and because of the number of computations involved. Other avenues are obtaining the **Sensitivity Analysis** of solution with respect to design variables of interest. This can be achieved with attaining the 1st order differentials which indicates the max/min of function. Statistical variations certainly has proved its value in many areas, such as quality control in manufacturing and uncertainty quantification is likely to become increasingly more important in the use of simulations in design.

1.3 Multiscale/Multiphysics

Multiscale is a broad term that usually means what the user intends it to. In most cases, however, it is used to mean phenomenon where some aspects of the physics that we wish to compute must be described by a different physical model. This can include contact lines in multiphase flow simulations represented by molecular or phase field models, reaction zones, shocks in rarified gases and so on. While we often think of multiscale representing different physical processes, such as continuum and non-continuum descriptions, it also applies to the same physics but modeled in different ways, such as when small drops are modeled as point particles. Numerical challenges include how to blend one description with another.

1.4 Mesh Free Methods for CFD

While the generation of meshes has always posed challenges for computational scientists, the problem has become more acute in recent years. While algorithms have seen great advances, mesh generation has lagged behind, creating a computational bottleneck. For industry and government looking to impact current and future products with simulation technology, mesh generation imposes great challenges. Many generation procedures often lack automation, requiring many man-hours, which are becoming far more expensive than computer hardware. More automated methods are less reliable for complex geometry with sharp corners, concavity, or otherwise complex features. Most mesh generation methods to date require a great deal of user expertise to obtain accurate simulation results. Since the application of computational methods to real world problems appears to be paced by mesh generation, alleviating this bottleneck potentially impacts an enormous field of problems¹. **Meshless methods** applied to computational fluid dynamics is a relatively new area of research designed to help alleviate the burden of mesh generation. Despite their recent beginning, there exists no shortage of formulations and algorithms for meshless schemes in the literature. A brief survey of the field reveals varied approaches arising from diverse mathematical backgrounds applied to a wide variety of applications. All meshless schemes attempt to bypass the use of a conventional mesh entirely or in part by discretizing governing partial differential equations on scattered clouds of points or collection of smooth blob of particles.

There are two different approaches which are called meshless. One contains methods like surface panel methods, boundary element methods, etc. which do not contain a volume grid. The other types are those which use an arbitrary distribution of points in the computational domain. **Particle methods** also belong to this category where the particles themselves act as discretization points. The method is called meshless because the points need not form any grid and they do not have to be arranged in any particular manner. The main motivation of meshless methods is that it is much easier to generate a *point* mesh. The accuracy of grid-based methods depends on the quality of the grid and so you have to ensure orthogonality, or make sure that elements are not highly skewed, while meshless methods are not very much affected by how the points are distributed².

A brief survey of the field reveals varied approaches arising from diverse mathematical backgrounds

¹ Aaron Jon Katz, ResearchGate, "Meshless methods for computational fluid dynamics", January 2009.

² From CFD Online Forum.

applied to a wide variety of applications. Sorting and classifying the many meshless methods is no simple task. To add to the confusion, meshless schemes fall under many other names *including mesh free, grid free, grid less, generalized finite difference*, and *Smooth Particle Hydrodynamics (SPH)*. We try to adapt the *mesh free* vocabulary here. From the above methods, smooth particle hydrodynamics (SPH) is distinctive in mesh free methods. It is where the fluid mass is lumped into smoothed blobs that are moved using Newton's second law directly, without an underlying mesh. In SPH the fluid is modeled as a collection of smooth "blobs" or particles³.

1.5 Integrated Simulations of Complex Systems

Engineers have long desired to have computational models that describe systems consisting of many coupled components. At the simplest level such simulators model the dynamics of connected rigid bodies, lumped models of chemical and power plants and so on. As computers become more powerful we are seeing growing efforts to attempt much more complex modeling, such as of rockets (the Illinois ASCI center) or a nuclear power plant (CASL), and other DOE research hub funded efforts. Other examples include the Human Body Simulator Project in Japan (lead by S. Takagi) and possibly the recently announced Living Earth Simulator proposal by D. Helbing. Overall there is very limited theoretical basis for how to do the coupling (with some exceptions such as for solid/fluid problems) and that there is considerable room for significant progress. As the ASCI programs, CASL and other effort suggest, this is going to be a very significant area in the future. (See [Figure 1.5.1](#)).

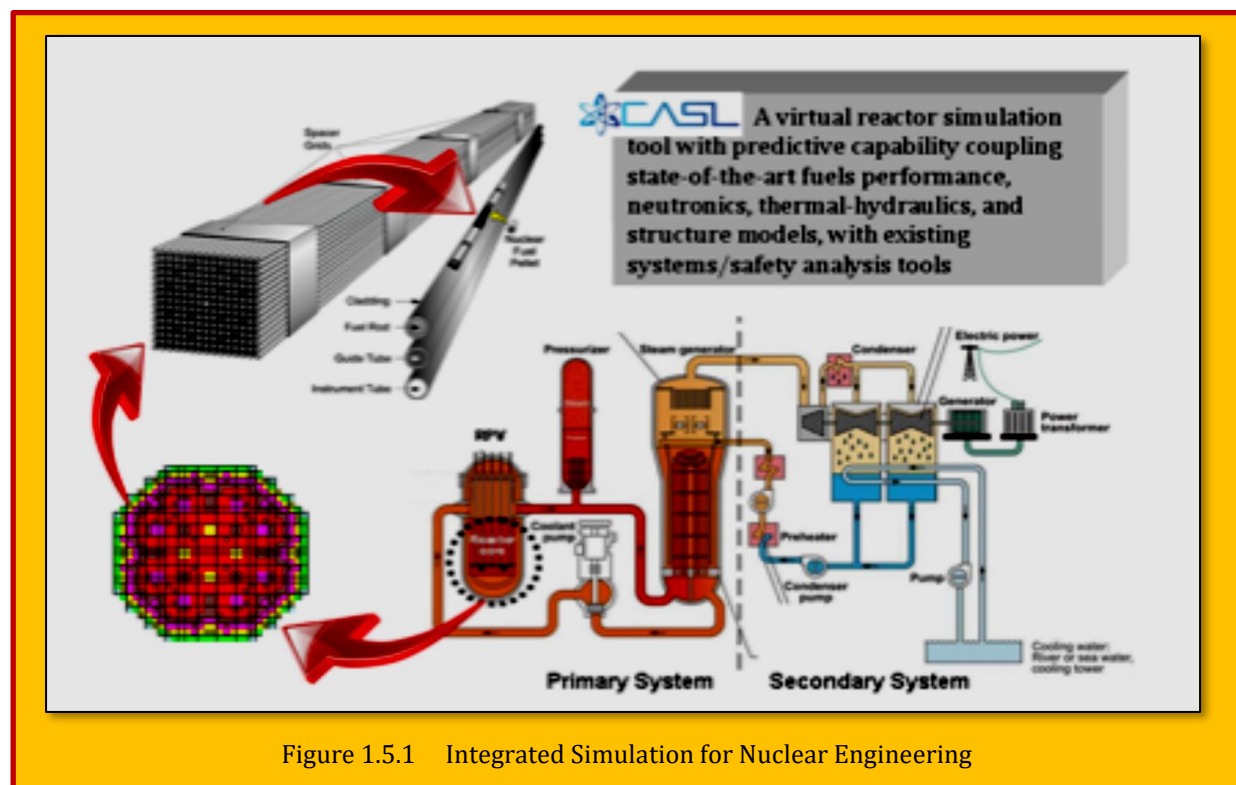


Figure 1.5.1 Integrated Simulation for Nuclear Engineering

1.5.1 Case Study - CAD Embedded CFD

While pushing simulation forward in the design cycle is a noble concept that has the potential to reduce costly engineering change orders, getting there is not as simple as making computational fluid dynamics (CFD) analysis available to designers within the 3D CAD interfaces they are familiar with. A direct CAD interface with CFD in and of itself does not provide any new or improved solutions to the

³ Grétar Tryggvason, "Smooth Particle Hydrodynamics", Lecture Series 2013.

biggest challenges of CFD simulation, i.e., geometry simplification and cleanup, extracting the fluid region from what is typically a 3D model of the solids involved, ensuring simulation accuracy through high-quality meshing, and CFD process quality controls. However, one particularly beneficial aspect of CAD-embedded or CAD-linked CFD that should be noted is its potential for enabling and streamlining parametric studies⁴. Prime examples of CAD embedded CFD are:

- SolidWorks Flow Simulation
- Autodesk CFD
- ANSYS Discovery Live
- NX-embedded CFD Simulation FloEFD for NX



⁴ Stewart Bible and Caleb Triece, P.E, “*CAD Embedded CFD: Overhyped, but a Good Value for Some*”, Digital Engineering, 2018.

2 Reduced Order Modeling (ROM)

2.1 Introduction

Many modern mathematical models of real-life processes pose challenges when used in numerical simulations, due to complexity and large size (dimension). Model order reduction aims to lower the computational complexity of such problems, for example, in simulations of large-scale dynamical systems and control systems. By a reduction of the model's associated state space dimension or degrees of freedom, an approximation to the original model is computed. This **Reduced Order Model (ROM)** can then be evaluated with lower accuracy but in significantly less time. Reduced order models (ROM) can be thought of as computationally inexpensive mathematical representations that offer the potential for near real-time analysis. While most ROMs can operate in near real-time, their construction can however be computationally expensive as it requires accumulating a large number of system responses to input excitations. Furthermore, **ROM usually lack robustness with respect to parameter changes and therefore must often be rebuilt for each parameter variation.**

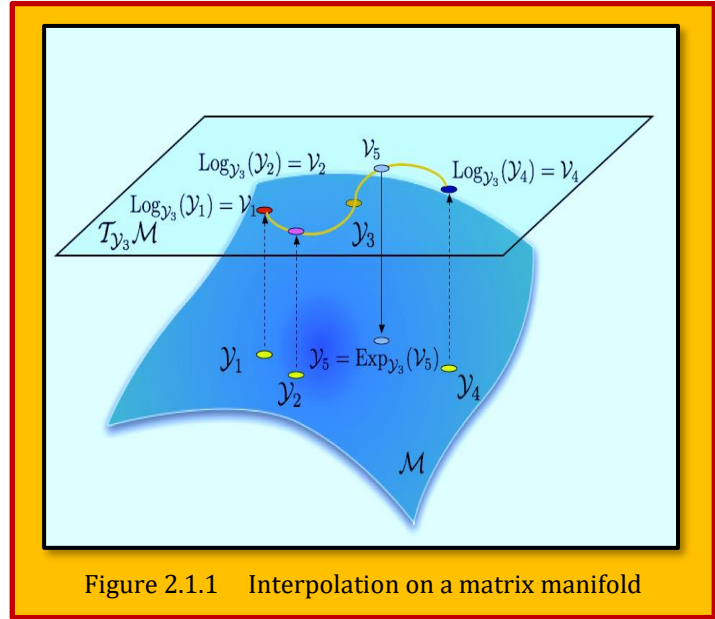


Figure 2.1.1 Interpolation on a matrix manifold

Together, these two issues underline the need for a fast and robust method for adapting pre-computed ROMs to new sets of physical or modeling parameters. However, ROMs and their corresponding Reduced Order Bases (ROB) are quantities that typically belong to nonlinear, matrix manifolds. As such, classical interpolation methods fail, as they are not able to enforce the constraints characterizing those manifolds. The first part of the project consists of designing a suitable interpolation method enforcing those constraints. A schematic representation of the algorithm is shown in **Figure 2.1.1**. It relies on identifying the correct manifold for the given application, constructing the appropriate logarithm mapping to move the interpolation data to a tangent space to this manifold where a standard multivariate interpolation algorithm can be applied, and constructing the appropriate exponential mapping to bring back the computed result to the manifold of interest⁵. The purpose of reduced order models (ROMs) is:

- taking advantage of redundancies
- identifying 'genuine' degrees of freedom
- giving low dimensional approximations (few modes)
- preserving a satisfactory accuracy
- decreasing the computational resources (time & storage)

Reduced Order Models (ROMs) based on statically non-linear flow solutions, but with a dynamically time linear approach have been developed. Thus unsteady flows that are a small perturbation about a steady flow with shocks and separations are modelled. This makes ROMs ideal for applications such as flutter clearance and aero-servo-elasticity. To generate a ROM about a particular non-linear mean solution, the dynamically time linear response must be extracted from the CFD code. In the

⁵ Farhat Research group.

study of fluid mechanics, there can be distinct physical features that are shared across a variety of flows and even over a wide range of parameters such as the Reynolds number and Mach number. Examples of common flow features and phenomena include *von Kármán shedding*⁶⁻⁷⁻⁸, *Kelvin-Helmholtz instability*⁹, and *vortex pairing/merging*. The fact that these features are often easily recognized through simple visual inspections of the flow, even under the presence of perturbations or variations, provides us with the expectation that the features can be extracted through some mathematical procedure. We can further anticipate that these dominant features provide a means to describe in a low-dimensional form that appears as a complex high-dimensional flow. Moreover, as computational techniques and experimental measurements are advancing their ability in providing large-scale high-fidelity data, the compression of a vast amount of flow field data to a low-dimensional form is ever more important in studying complex fluid flows and developing models for understanding and modeling their dynamical behavior¹⁰.

2.2 Various Techniques

There is a large variety of ROMs in the market. They are also known as *surrogate* models. A common approach for model order reduction is *Projection-Based* reduction. The following methods fall into this class:

- The classic *Proper Orthogonal Decomposition (POD)* with *Galerkin projection*.
- Either *Singular Value Decomposition (SVD)* or high-order singular value decomposition (HOSVD), possibly combined with interpolation.
- Reduced Basis Method.
- Balanced Truncation.
- Approximate Balancing.
- Matrix Interpolation.
- Transfer Function Interpolation.
- Piecewise Tangential Interpolation.
- Loewner Framework.
- Empirical (Cross Gramian)¹¹.
- Krylov Subspace Methods.

Among those, the application of the *POD-Galerkin reduced order modelling* for Finite Volume discretization technique is gained more industrial fields acceptance.

2.3 Common Features Shared by Reduced Order Methods (ROM)

All reduced bases require the solution of high-fidelity and therefore very expensive discrete state and/or sensitivity equations and/or adjoint equations. The idea is that these expensive calculations can be done off-line before a state simulation or the optimization of the design parameters or feedback control is attempted. Moreover, one hopes that a single reduced basis can be used for

⁶ Strouhal, V., "On One Particular Way of Tone Generation," Annalen der Physik und Chemie (Leipzig), Series , Vol. 241, No. 10, 1878, pp. 216–251.

⁷ Rayleigh, L., "Acoustical Observations," Philosophical Magazine and Journal of Science, Vol. 7, No. 42, 1879.

⁸ Taneda, S., "Experimental Investigation of the Wakes Behind Cylinders and Plates at Low Reynolds Numbers," Journal of the Physical Society of Japan, Vol. 11, No. 3, 1956, pp. 302–307.

⁹ von Helmholtz, H., "On Discontinuous Movements of Fluids," Philosophical Magazine, Vol. 36, No. 244, 1868.

¹⁰ Kunihiro Taira, Steven L. Brunton, Scott T. M. Dawson, Clarence W. Rowley, Tim Colonius, Beverley J. McKeon, Oliver T. Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S. Ukeiley, "Modal Analysis of Fluid Flows: An Overview", AIAA Journal, Vol. 55, No. 12, December 2017.

¹¹ In control theory, the cross Gramian is a Gramian matrix used to determine how controllable and observable a linear system is.

several state simulations or in several design or control settings¹². All reduced-basis sets are global in nature, i.e., the support the basis functions globally. Therefore, solving the state or sensitivity or adjoint equations with respect to any of the reduced bases requires the solution of dense linear and nonlinear systems. Thus, unless the dimension of a reduced basis is “small,” it cannot be used without some further processing. Unfortunately, in order to obtain meaningful approximations, it is often the case that the use of reduced bases requires the use of a relatively large number of basis functions. However, it is often the case that reduced bases contain “redundant” information in the sense that the dynamics of the state should be well approximated by a set of functions of much lower dimension. The question then arises: how can one extract a reduced basis of smaller dimension that contains all the essential information of a reduced basis of larger dimension? This is *where Proper Orthogonal Decompositions (POD)* and *Cantorial Voronoi Tessellations (CVT)* come in and, in this sense, they are reduced-reduced basis methods. Unfortunately, there is no adequate theoretical foundation for reduced-order methods, even in state simulation settings. However, it is certain that without an inexpensive method for reducing the cost of state computations, it is unlikely that the solution of 3D optimization and control problems involving complex systems, e.g., the Navier-Stokes system, will become routine anytime soon. Thus, it is also certainly true that these methods deserve more study from the computational and theoretical points of view.

2.4 Reduced Basis Methods

All reduced-order methods are reduced basis methods. However, there is a class of methods that use *Lagrange* bases, *Hermit* bases, *Taylor* bases, and *Snapshot* bases (or more precisely, snapshot sets) that have come to be known as *Reduced-Basis Methods*.

2.4.1 Lagrange

Lagrange bases consist of state solutions corresponding to several different values of the parameters (Reynolds number, design parameters, etc.). These solutions are obtained by standard (and expensive) techniques such as finite element or finite volume methods. For example, if one has the design parameters $\{\alpha_j\}_{j=1,J}$, one obtains an approximate state solutions for n sets of parameter values to form the n -dimensional Lagrange reduced basis¹³.

2.4.2 Hermit

Hermit bases consist of the state variables and the first derivatives of the state variables with respect to parameters (the sensitivities) determined for different values of the parameters. The state and sensitivity approximations are obtained through standard (and expensive) techniques such as finite element or finite volume methods. Thus, again, if one has the design parameters $\{\alpha_j\}_{j=1,J}$, one chooses M sets of parameter values and then one obtains the corresponding M approximate state solutions and the corresponding MJ sensitivity derivative approximations. The $n = M(J + 1)$ state and sensitivity approximations form the Hermit reduced basis of dimension n .

2.4.3 Taylor

Taylor bases consist of the state and derivatives of the state with respect to parameters (sensitivities and higher-order sensitivities) determined for a fixed set of design parameters. The state and derivative approximations are obtained through standard (and expensive) techniques such as finite element or finite volume methods. The Taylor basis may be somewhat complicated to program due to the complexity of the partial differential equations that determine the higher-order sensitivities. In addition, the number of higher-order derivatives grows very rapidly with the number of design

¹² John Burkardt, Qiang Du, Max Gunzburger & Hyung-Chun Lee, “Reduced order modeling of complex systems”, NA03 Dundee 2003.

¹³ John Burkardt, Qiang Du, Max Gunzburger & Hyung-Chun Lee, “Reduced order modeling of complex systems”, NA03 Dundee 2003.

parameters, e.g., if one has 10 design parameters, there are 55 different second derivative sensitivities. Thus, the dimension of the Taylor reduced basis grows quickly with the number of parameters and the number of derivatives used.

2.4.4 Snapshot Sets Method

The state of a complex system is determined by parameters that appear in the specification of a mathematical model for the system. Of course, the state of a complex system also depends on the independent variables appearing in the model. Snapshot sets consist of state solutions corresponding to several parameter values and/or evaluated at several values of one or more of the dependent variables. For example, steady-state solutions corresponding to several sets of design parameters or a time-dependent state solution for a fixed set of design parameter values evaluated at several time instants during the evolution process. Or several state solutions corresponding to different sets of parameter values evaluated at several time instants during the evolution process. Snapshot sets are often determined by solving the full, very large-dimensional discretized system obtained via finite volume or finite element discretization. Experimental data have also been used to determine a snapshot set. Snapshot sets often contain “redundant” information; therefore, snapshot sets must usually be post-processed to remove as much of the redundancy as possible before they can be used for reduced-order modeling. POD and CVT may be viewed as simply different ways to post-process snapshot sets.

Since snapshot sets are the underpinning for POD and CVT, we briefly discuss how they are generated in practice. At this time, the generation of snapshot sets is an art and not a science; in fact, it is a rather primitive art. The generation of snapshot sets is an exercise in the design of experiments, e.g., for stationary systems, how does one choose the sets of parameters at which the state (and sensitivities) are to be calculated (using expensive, high-fidelity computations) in order to generate the snapshot set? Clearly, some a priori knowledge about the types of states to be simulated or optimized using the reduced-order model is very useful in this regard. The large body of statistics literature on the design of experiments has not been used in a systematic manner. For time-dependent systems, many (ad hoc) measures have been invoked in the hope that they will lead to good snapshot sets. Time-dependent parameters (e.g., in boundary conditions) are used to generate states that are “rich” in transients, even if the state of interest depends only on time-independent parameters. In order to generate even “richer” dynamics, impulsive forcing is commonly used, e.g., starting the evolution impulsively with different strength impulses and introducing impulses in the middle of a simulation. In the future, a great deal of effort needs to be directed towards developing and justifying methodologies for generating good snapshot sets¹⁴. After all, a POD or CVT basis is only as good as the snapshot set used to generate it.

2.5 Proper Orthogonal Decomposition (POD) Spaces

In order to create a reduced basis space onto which the governing equations are projected, one can find many techniques in literature such as the **Proper Orthogonal Decomposition (POD)**, **Proper Generalized Decomposition (PGD)**, as well as **Reduced Basis (RB)** method with a greedy approach. The POD approach is been selected here. The POD consists into the decomposition of the flow fields into temporal coefficients $a_i(t)$ and orthonormal spatial bases $\varphi_i(x)$:

$$u(x, t) = \sum_{i=1}^{N_s} a_i(t) \varphi_i(x)$$

Eq. 2.5.1

¹⁴ John Burkardt, Qiang Du, Max Gunzburger & Hyung-Chun Lee, “Reduced order modeling of complex systems”, NA03 Dundee 2003.

where $\varphi_i(x)$ are orthonormal spatial bases that minimizes the average of the error between the snapshots, and their orthogonal projection onto the bases and N_s is the number of considered snapshots. The POD space $V_{\text{POD}} = \text{span}(\varphi_1, \varphi_2, \dots, \varphi_{N_s})$ is then constructed solving the following minimization problem:

$$V_{\text{POD}} = \arg \min \frac{1}{N_s} \sum_{n=1}^{N_s} \left\| u_n(x) - \sum_{i=1}^{N_s} (u_n(x), \varphi_i(x))_{L^2(\Omega)} \varphi_i(x) \right\|_{L^2(\Omega)}^2$$

where $(\varphi_i(x), \varphi_j(x)) = \delta_{ij}$

Eq. 2.5.2

where u_n is a general snapshot of the velocity field at time $t = t_n$. The snapshot can be numerical solutions of the NSEs (typical from LES and DNS simulations or even by the RANS equations) or they are obtained from experimental results. The POD basis minimizes the difference between the snapshots and the projection of the snapshots on the spatial modes in the X-norm, given the orthonormality of the modes. If the L^2 -norm is chosen, the POD basis is optimal considering the energy contained in the snapshots. Following development in¹⁵. It can be shown that this problem can be solved computing a singular value decomposition of the so called snapshots matrix.

The snapshot based method enables us to perform the decomposition in a computationally tractable manner when the dimension of an individual snapshot is much larger than the total number of snapshots. In performing the POD here, we first subtract the mean from all snapshots, so that we can focus on modal structures associated with fluctuations. The extracted spatial POD modes $\varphi_i(x)$ of Eq. 2.5.1 capture regions where fluctuations appear in the flow. Since this cylinder flow example is a periodic ow, these spatial modes appear in pairs. This also suggests that the modes are based on advective physics with oscillator-type dynamics¹⁶.

2.5.1 Case Study 1 - 2D Laminar Separated Flow Over a Flat-Plate Wing

To briefly illustrate these ideas, let us provide a preview of modal decomposition. In **Figure 2.5.1**, we present a modal decomposition analysis of 2D laminar separated flow over a flat-plate wing¹⁷⁻¹⁸. By inspecting the flow field, we clearly observe the formation of a von Kármán vortex street in the wake as the dominant unsteady feature. A modal decomposition method discussed, (see Sec. III of [Taira et al.]¹⁹), can extract the important oscillatory modes of this flow. Moreover, two of these most dominant modes and the mean represent (reconstruct) the flow field very effectively, as shown in the bottom figure. Additional modes can be included to reconstruct the original flow more accurately, but their contributions are much smaller in comparison to the two unsteady modes shown in this example. What is also encouraging is that the modes seen here share a striking resemblance to the

¹⁵ Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, Gianluigi Rozza, "Advances in Reduced order modelling for CFD: vortex shedding around a circular cylinder using a POD-Galerkin method", Communications in Applied and Industrial Mathematics ISSN 2038-0909, 2017.

¹⁶ Kunihiko Taira, Maziar S. Hemati, Steven L. Brunton, Yiyang Sun, Karthik Duraisamy, Scott T. M. Dawson, and Chi-An Yeh, "Modal Analysis of Fluid Flows: Applications and Outlook", AIAA, 2019.

¹⁷ Taira, K., and Colonius, T., "Three-Dimensional Flows Around Low-Aspect-Ratio Flat-Plate Wings at Low Reynolds Numbers," Journal of Fluid Mechanics, Vol. 623, 2009, pp. 187–207.

¹⁸ Colonius, T., and Taira, K., "A Fast Immersed Boundary Method Using a Null space Approach and Multi-Domain Far-Field Boundary Conditions," Computer Methods in Applied Mechanics and Engineering, Vol. 197, Nos. 25–28, 2008, pp. 2131–2146.

¹⁹ Kunihiko Taira, Steven L. Brunton, Scott T. M. Dawson and Clarence W. Rowley, Tim Colonius, Beverley J. McKeon, Oliver T. Schmidt, Stanislav Gordeyev, Vassilios Theofilis, Lawrence S. Ukeiley, "Modal Analysis of Fluid Flows: An Overview", AIAA Journal, Vol. 55, No. 12, December 2017.

dominant modes for three-dimensional turbulent flow at a much higher Reynolds number of 23,000 with a different airfoil and angle of attack (see Sec. III.B.1 of [Taira et al.]²⁰).

We refer to modal decomposition as a mathematical technique to extract energetically and dynamically important features of fluid flows. The spatial features of the flow are called (spatial) modes, and they are accompanied by characteristic values, representing either the energy content levels or growth rates and frequencies. These modes can be determined from the flow field data or from the governing equations. We will refer to modal decomposition techniques that take flow field data as input to the analysis as data-based techniques.

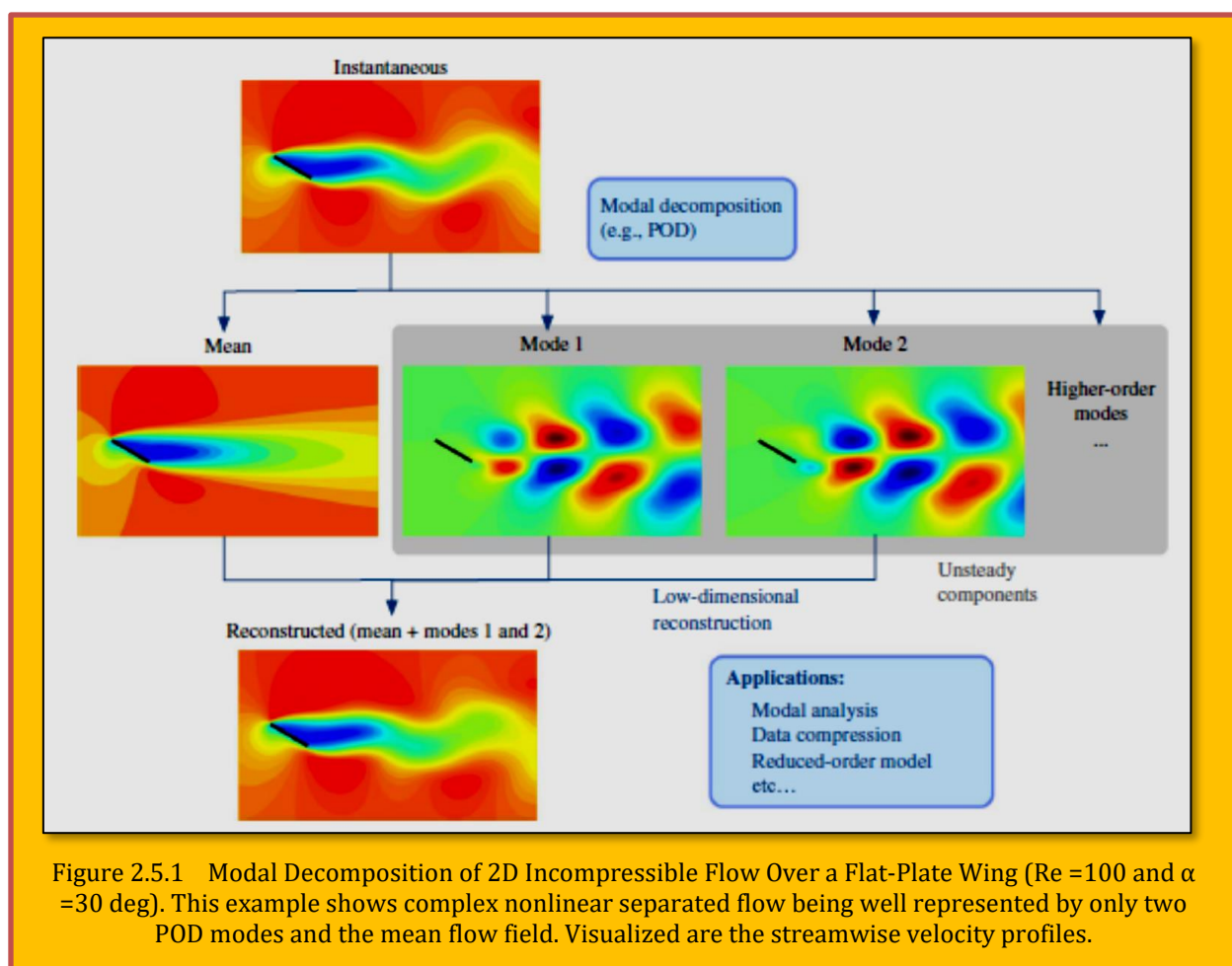


Figure 2.5.1 Modal Decomposition of 2D Incompressible Flow Over a Flat-Plate Wing ($Re = 100$ and $\alpha = 30^\circ$). This example shows complex nonlinear separated flow being well represented by only two POD modes and the mean flow field. Visualized are the streamwise velocity profiles.

2.5.2 Case Study 2 - Cylinder Wakes

The POD analysis reveals that the fluctuations in the flow field can be captured well with only a small number of mode pairs as illustrated in **Figure 2.5.2**. The first 2, 4, and 6 modes capture 94:84%, 98:68%, and 99:85%, respectively, of the flow fluctuations in terms of the kinetic energy. With 8 modes, this percentage reaches 99:97%, which is essentially 100%. This means that the high-dimensional flow field can be accurately expressed with only 6 or 8 spatial modes, suggesting the possibility for significant compression of the flow field data. That is, we reduce the representation of the flow field from the number of grid points (times the number of flow variables) to merely the

²⁰ Kunihiro Taira, Steven L. Brunton, Scott T. M. Dawson and Clarence W. Rowley, Tim Colonius, Beverley J. McKeon, Oliver T. Schmidt, Stanislav Gordeyev, Vassilios Theofilis, Lawrence S. Ukeiley, "Modal Analysis of Fluid Flows: An Overview", AIAA Journal, Vol. 55, No. 12, December 2017.

number of POD modes. The mode shapes associated with the dominant POD modes reveal the dominant energetic spatial structures in the ow. Interestingly, both POD modes 1 and 2 possess a top-down asymmetry, indicating that the dominant energetic structures are associated with the asymmetry of the Karman wake. As it will be discussed in a latter section, these POD modes can serve as a basis to construct a reduced-order model that describes the dynamics of the flow. One of the important properties of the POD modes is the orthogonality of the modes (i.e., $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$), which is attractive for developing sparse reduced-order representation of the flow dynamics.

The above 8 POD modes can capture the flow field very well for the given data. However, if the flow is perturbed and deviates away from the original ow, additional modes may be needed to represent the perturbed flow. To better capture the perturbed flow, POD analysis may be repeated with the perturbed flow field data or alternative techniques such as the Balanced POD analysis²¹ may be utilized (although an adjoint simulation is needed for the latter case). We should keep in mind that the modes extracted from the input flow field data are optimally determined for the provided data and may not be so for the perturbed flows. The modes may deform when the ow is under the influence of perturbation or actuation. This is an important point to remember if modal analysis is to be extended or mode-based models are applied to perform flow control.

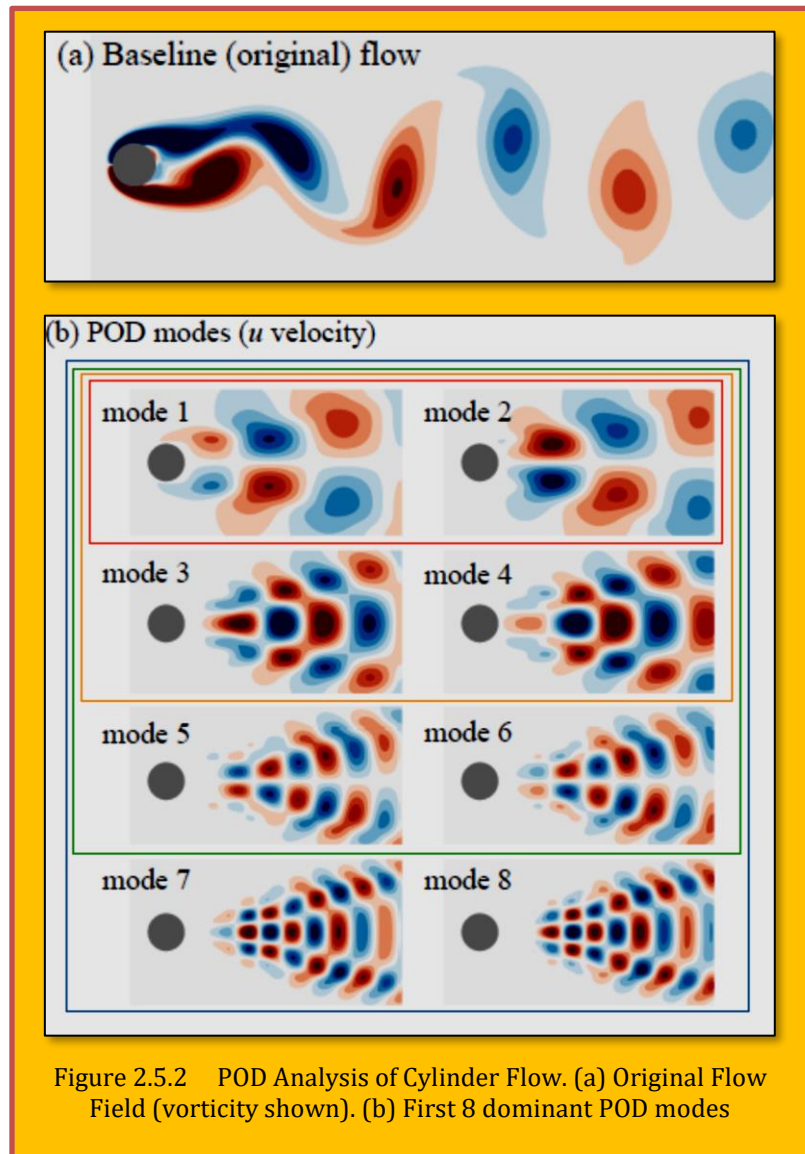


Figure 2.5.2 POD Analysis of Cylinder Flow. (a) Original Flow Field (vorticity shown). (b) First 8 dominant POD modes

2.6 Galerkin Projection into POD Space

In this section the Galerkin projection of the governing equations onto the POD space is highlighted and discussed. The idea here is to consider both the momentum conservation and continuity equation. In order to be consistent with the full order solver, the same set of equations are considered, namely the momentum conservation and the Poisson equation for pressure.

²¹ Rowley, C. W., \Model reduction for fluids using balanced proper orthogonal decomposition," Int. J. Bifurcation Chaos, Vol. 15, No. 3, 2005, pp. 997-1013.

2.6.1 Case Study - Vortex Shedding Around a Circular Cylinder using a POD-Galerkin Method

Vortex shedding around circular cylinders is a well-known and studied phenomenon that appears in many engineering fields. In this work a Reduced Order Model (ROM) of the incompressible flow around a circular cylinder, built performing a **Galerkin projection** of the governing equations onto a lower dimensional space is presented. The reduced basis space is generated using a Proper Orthogonal Decomposition (POD) approach. In particular the focus is into:

- The correct reproduction of the pressure field, that in case of the vortex shedding phenomenon, is of primary importance for the calculation of the drag and lift coefficients;
- For this purpose the projection of the Governing equations (momentum equation and Poisson equation for pressure) is performed onto different reduced basis space for velocity and pressure, respectively;
- All the relevant modifications necessary to adapt standard finite element POD-Galerkin methods to a finite volume framework are presented. The accuracy of the reduced order model is assessed against full order results.

2.6.1.1 Governing Equations

For the moment, we consider the incompressible Navier–Stokes equations without any turbulence treatment as

$$\nabla \cdot \mathbf{u} = 0 \quad , \quad \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = 0$$

Eq. 2.6.1

where $\mathbf{u}$ is the velocity, p is a normalized pressure and ν is the kinematic viscosity. The equations are given in a domain Ω with proper boundary and initial conditions. The Finite Volume method is a discretization method based on a “balance” approach, well suited for the solution of equations based on conservation laws. A local balance, obtained from the discretization of the integral form of the governing equations, is written on each discretization cell. As for details, readers should consult ²²⁻²³. This approach can be interpreted as if the state vector of the variables of interest was expanded as linear combination of state vector spatial modes:

$$\begin{pmatrix} \mathbf{u}(\mathbf{x}, t) \\ F(\mathbf{x}, t) \\ p(\mathbf{x}, t) \end{pmatrix} \approx \begin{pmatrix} \mathbf{u}_r(\mathbf{x}, t) \\ F_r(\mathbf{x}, t) \\ p_r(\mathbf{x}, t) \end{pmatrix} = \sum_{i=1}^{N_r} a_i(t) \begin{pmatrix} \phi_i(\mathbf{x}) \\ \psi_i(\mathbf{x}) \\ \chi_i(\mathbf{x}) \end{pmatrix}$$

Eq. 2.6.2

Replacing the velocity $\mathbf{u}$ with $\mathbf{u}_r$ and p with p_r in **Eq. 2.6.2**, employing the approximated face flux F_r in the convective term, and applying the **Galerkin projection**. The reduced order model of the momentum equation is obtained performing an L2 orthogonal projection onto the reduced bases space V_{POD} spanned by the POD velocity modes with a procedure similar to what presented in

$$(\phi_i, \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p)_{L^2(\Omega)} = 0$$

Eq. 2.6.3

With respect to what presented in Error! Reference source not found. here also the gradient of pressure is considered inside the momentum equation. It is assumed that velocity and pressure modes share the same temporal coefficients. Substituting the POD approximations of $\mathbf{u}$, F and p into **Eq.**

²² Stefano Lorenzi, Antonio Cammi, Lelio Luzzi, Gianluigi Rozza, “*POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations*”, Comput. Methods Appl. Mech. Engr. 311 (2016) 151–179.

²³ Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, Gianluigi Rozza, “*Advances in Reduced order modelling for CFD: vortex shedding around a circular cylinder using a POD-Galerkin method*”, Communication Appl. Ind. Math. 9 (1), 2017, 1–21.

2.6.3 and exploiting the orthogonality of the POD modes, one obtains the following dynamical system of Ordinary Differential Equations (ODEs). The following POD-Galerkin ROM for Finite Volume discretization (POD-FV-ROM) is obtained as:

$$\frac{da_j(t)}{dt} = \nu \sum_{i=1}^{Nr} B_{ji} a_i(t) - \sum_{k=1}^{Nr} \sum_{i=1}^{Nr} C_{jki} a_k(t) a_i(t) - \sum_{i=1}^{Nr} A_{ji} a_i(t)$$

$$B_{ji} = (\phi_j, \Delta \phi_i)_{L^2} \quad , \quad C_{jki} = (\phi_j, \nabla \cdot (\psi_k, \phi_i))_{L^2} \quad , \quad A_{ji} = (\phi_j, \nabla \chi_i)_{L^2}$$

Eq. 2.6.4

2.6.1.2 Details of the Full Order Simulation

The convective term is discretized in space making use of the Gauss's theorem). The face center values of the variables are obtained from the center cell ones, which are the numerical problem unknowns, with an interpolation scheme consisting into a combination of a linear and upwind scheme. The diffusive term is discretized in a similar fashion. In this case though, a central differencing interpolation scheme with non-orthogonality correction is preferred. Also the pressure gradient is discretized making use of Gauss's theorem. Here, the face center pressure values are obtained from the cell center ones by means of a linear interpolation scheme, in which a limiting operation on the gradient is performed so as to preserve the monotonicity condition and ensure that the extrapolated face value is bounded by the neighboring cell values. As for the time discretization, a backward Euler scheme is used. The overall time extent of the simulation is equal to $T = 3645s$, which is sufficiently long to reach a perfectly periodic response of the lift and drag forces. The simulation is run in parallel on 4 Intel R Core™ processors, taking $T_{CPU-HF} = 1483s \approx 25min$ to be completed.

2.6.1.3 Details of the ROM Simulation

The ROM is constructed using the methodologies described in x 3. For the generation of the POD spaces, we considered 120 snapshots of the velocity, mass flux and pressure fields. The snapshots are collected in a time window covering approximately 1.5 periods of the vortex shedding phenomenon. More precisely, the last 73s of the HF simulation are used. The first two modes for velocity and pressure field respectively are presented in **Figure 2.6.1**. The ROM simulations are carried out using different values of the POD velocity space dimension $N_u = 3, 5, 7, 10$. The dimension of the POD pressure and mass u_x space is set equal to the dimension of the velocity POD space $N_u = N_p$. The ROM simulation is run in serial, on the same processor used for the HF simulation. In this case, the time advancing of the ROM problem is carried out using the *Matlab ODE* suite. Reproducing the full 3645s extent of the high fidelity (HF) simulation requires, using the ROM model with the highest dimension of the POD space, approximately $T_{CPU-ROM} = 9.10s$. This corresponds to a speedup $SU \approx 650$.

2.6.1.4 Analysis of the Results

Using the settings described in the previous paragraph, four different ROM simulations are run, each featuring a different value of the POD space dimension. The results are compared with those of the High Fidelity (HF) simulation in terms of history of the lift and drag coefficients. The time window used for the comparison is the same window used for the collection of the snapshots. The lift coefficient comparison is reported in²⁴, while the drag coefficient time histories is presented in

²⁴ Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, Gianluigi Rozza, "Advances in Reduced order modelling for CFD: vortex shedding around a circular cylinder using a POD-Galerkin method", Communication Appl. Ind. Math. 9 (1), 2017, 1-21.

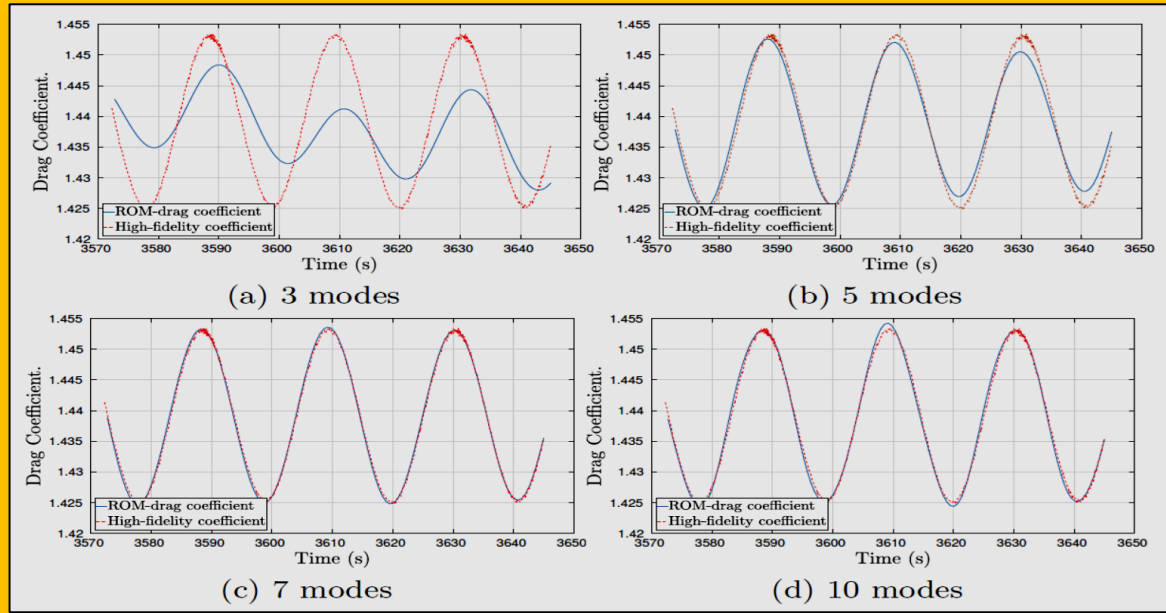


Figure 2.6.2 Comparison of the Drag Coefficient Obtained with the High Fidelity (HF) and ROM Simulations

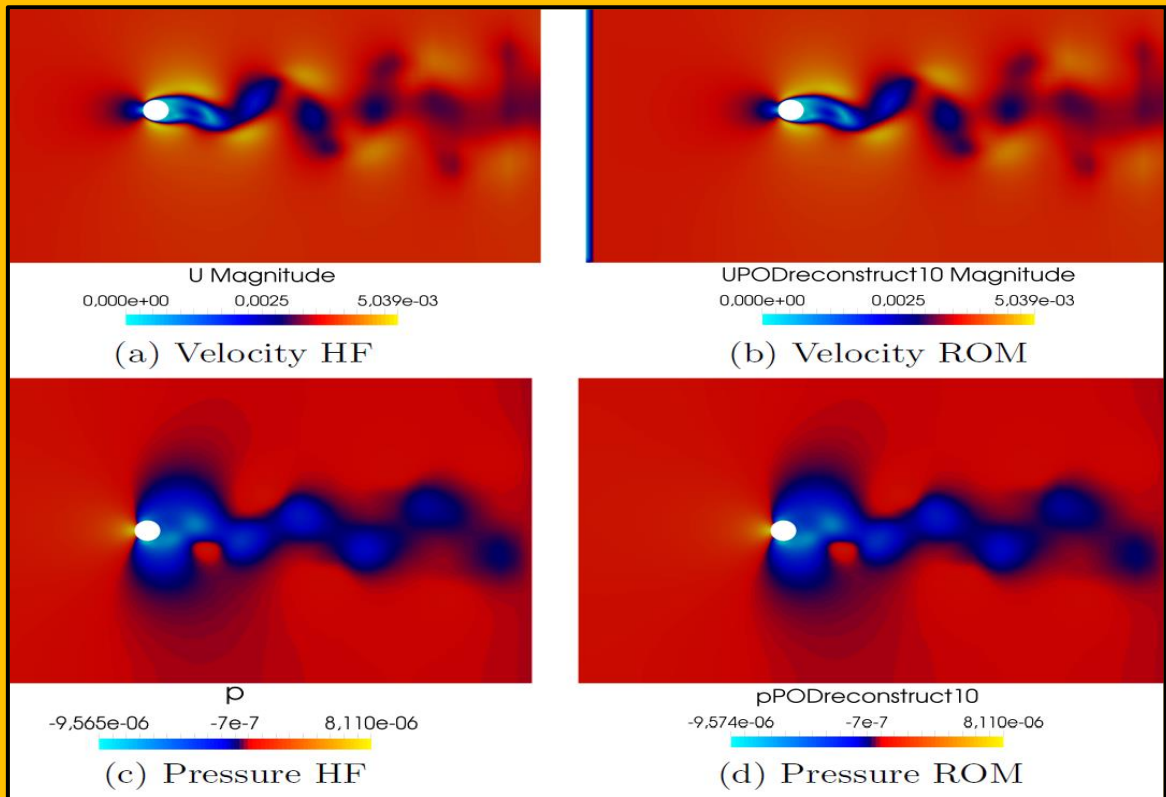


Figure 2.6.1 Comparison between velocity and pressure High Fidelity (HF)-ROM

Figure 2.6.2. **Figure 2.6.1** the comparison is shown directly on the velocity and the pressure fields. In this case, the time step considered is the last one of the simulations corresponding to $T = 3645s$. The left plot in **Figure 2.6.1** refer to the velocity (top) and pressure (bottom) fields computed with the high fidelity simulations. The right plots refer to the velocity (top) and pressure (bottom) fields computed with the ROM, in which the POD space dimension has be set to $N_u = 10$. The plots show that, at a glance the HF and ROM solutions cannot be distinguished.

2.7 Addressing Challenges in Reduced-Order Modeling

One of applied mathematics' great contributions is the foundation it provides for simulating physical phenomena. From the derivation of consistent, stable, and convergent discretization schemes to the development of efficient parallel solvers, mathematical advances have enabled the ubiquitous nature of modeling and simulation in applications ranging from protein-structure prediction to aircraft design. Today, the predictive capability of validated computational models allows simulation to replace physical experimentation in many scenarios, which facilitates the realization of deeper analyses and better designs at lower costs. However, there is a catch: the resolution required to achieve such high fidelity leads to large-scale models whose simulations can consume weeks on a supercomputer. This creates a massive gap between the simulation times of high-fidelity models and the rapid time-to-solution demands of time-critical (e.g., real-time analysis) and many-query (e.g., uncertainty quantification) applications in engineering and science²⁵.

To bridge this gap, researchers have pursued reduced-order modeling which integrates techniques from data science, modeling, and simulation as a strategy for reducing the computational cost of such models while preserving high levels of fidelity. First, these methods execute analyses (e.g., simulating the model, solving Lyapunov equations) during an off line 'training' stage; these analyses generate data that can be mined to extract important physical features, such as low-dimensional solution manifolds and interpolation points for approximating nonlinear functions. Next, these techniques reduce the dimensionality and computational complexity of the high-fidelity model by projecting the governing equations onto the low-dimensional manifold and introducing other approximations where necessary. The resulting reduced-order model (ROM) can then be rapidly simulated during an online 'deployed' stage. While significant advances have been made in reduced-order modeling over the past fifteen years, many outstanding challenges face the community, especially with respect to applying model reduction to parameterized nonlinear dynamical systems.

To address this, One workshop theme focused on applying ROMs to truly large-scale nonlinear problems in engineering and science. To motivate this, an invited speaker provided a number of compelling examples in which the computational cost incurred by such models poses a major bottleneck to design engineers across the naval, aerospace, and automotive industries. A number of challenges arise in this case. First, ROM techniques must be tightly integrated with the original high fidelity simulation code because most nonlinear ROM methods realize computational savings by performing computations with the high-fidelity model on a small subset of the computational domain. Second, ensuring accurate ROM solutions can be challenging due to the complex dynamics (e.g., stiffness) exhibited by many large-scale dynamical systems. Finally, when the model is very large scale, the computational costs of both the offline training and online deployment can remain prohibitive; devising ways to reduce them is often essential.

A second major workshop theme focused on applying ROMs to design optimization. These many-query problems which are often formulated as mathematical optimization problems constrained by partial differential equations can require hundreds of simulations (and sensitivity analyses) of the computational model. Thus, rapid model evaluations are necessary when faced with time or resource

²⁵ Kevin Carlberg, "Addressing Challenges in Reduced-Order Modeling", SIAM News March 2016.

constraints. [Louis Durlofsky]²⁶ proposed a related method based on the Trajectory Piece Wise Linear (TPWL) ROM, and showed promising results on oil-production optimization under water injection. Despite the many challenges, model reduction remains an exciting research area that is making rapid progress toward bridging the gap between high-fidelity models and time-critical applications in engineering and science.

2.8 Reduced Order CFD Simulation

The unsteady Euler and Navier-Stokes solutions have thousands of degrees of freedom. This means that the costs of unsteady flow studies are prohibitive. Schemes that retain the accuracy of the full non-linear methods, but at a reduced cost will make such studies feasible. This is the rationale for Reduced Order Models (ROM) which is based on statically non-linear flow solutions, but with a dynamically time linear approach have been developed. Thus unsteady flows that are a small perturbation about a steady flow with shocks and separations are modelled. This makes ROMs ideal for applications such as flutter clearance and aero-servo-elasticity. To generate a ROM about a particular non-linear mean solution, the dynamically time linear response must be extracted from the CFD code. A system identification and reduction scheme is then used to construct the ROM, a state space system, from the pulse responses. This system is of much lower order than the original non-linear CFD scheme, but is able to reproduce its behavior. The ROMs are in state space form and so can easily be coupled to a structural model for aero elastic and aero servo elastic calculations. One advantage of the current approach is that the aerodynamic model is constructed independently of the structural model and thus a redesigned structure does not require a new ROM. It could be shown that the flutter boundary of a 2D airfoil can be reproduced by ROM of order 18, where the original CFD is of order 27,000. The use of ROMs enabled each flutter point to be calculated in less than 1/100th of the computing time compared to the full CFD.

Over the years, 1D, 2D, 3D CFD software's solutions have been used successfully for modeling thermo-fluid systems in automotive, aerospace, oil, gas, power and energy industries. 1D CFD systems allows analyses of a wide range of complex engineering problems. For example, engineers

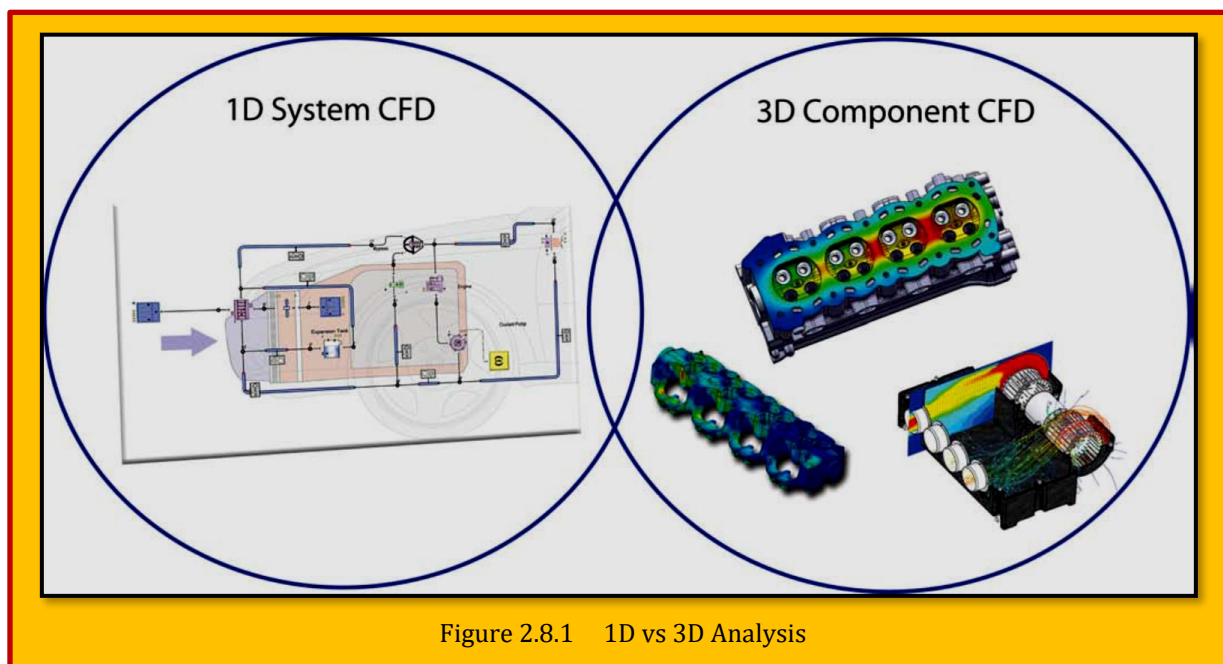


Figure 2.8.1 1D vs 3D Analysis

²⁶ He, J., & Durlofsky, L.J. "Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization". SPE Journal, 201

can rapidly and accurately analyze piping network of almost any size or complexity to establish design integrity. 2D CFD cross-section simulation is mostly used in airfoil design for aircraft or blade in pumps or compressors, turbines of turbo-machinery. However, the 3D CFD phenomena associated with these designs cannot be resolved in 2D simulations. In real world, all fluid flow problems are 3D in nature and very with time, however, with thoughtful care, the simulation of many components and systems can be run in fewer dimensions. If done properly, the results from a simplified solution can give just as meaningful insight, but with a fraction of computational effort. While the 1D CFD is best used for system level analysis to understand how different parts of a of system will interact, the 3D CFD is used for component level analysis to understand design tradeoffs of detailed parts design, as shown [Figure 2.8.1](#). In summary, while 1D CFD are typically much faster than 3D CFD calculations where it may take only minutes to perform and provide a relatively quick system overview, the 3D CFD is mainly used for design of individual components allowing engineers to understand how detail flow interacts with all manner complex geometry²⁷.

The question arises when to use 1D CFD vs. 3D CFD? While there is not a definitive answer, the strength and weakness of each approach lend themselves to two fairly defined arguments. When designing a single component or small subset of components, every inch of length or degree of curvature can make a difference. In these cases, when small changes to a single part of a system are crucial, or there are significant flow variations in multiple dimensions, 3D CFD is the obvious choice because of its ability to analyze complex geometry with extreme accuracy. However, these benefits come with drawbacks, which become more evident as the scale of design increases. When the design reaches beyond the component level, the computation requirement becomes too high and the simulations take too long to fit within development schedule. This is when 1D CFD is a good choice. Because the 1D approach simplifies the 3D geometry to the component level, usually characterized by some sort of performance data. This uses much less computing power and is usually faster than a comparable 3D model.

2.8.1 Case Study 1 - Designing Parameters of Test Stage Axial Turbine

At the present, turbo-machine element design using integrated software is developing intensively. Using 3D simulation in turbine flow path remains very labor intensive and sufficiently hampers its usage. Therefore, unidirectional (1D) and axisymmetric (2D) analyses are still widely used. Gas turbine engine qualitative characters are determined by the concepts taken into account on early phases of engine component design. The turbine multidisciplinary optimization problems are topic of different research. After 1D mean line calculation, a stage-by-stage 2D (axisymmetric) calculation was performed to determine the twist laws of blades which provide the highest efficiency. The first

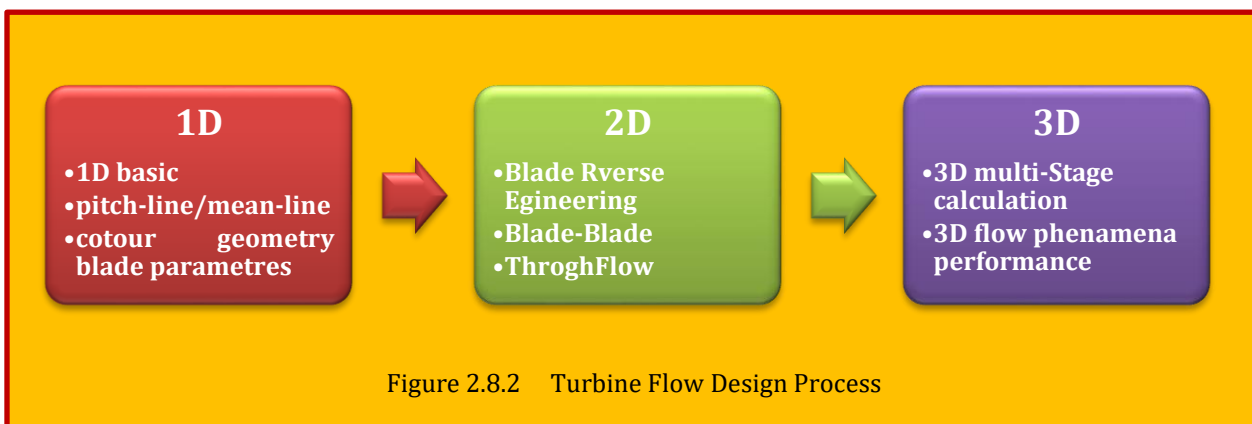


Figure 2.8.2 Turbine Flow Design Process

²⁷ Mentor Graphics Corporation®.

design stage (P1), was a prototype of Intermediate Pressure turbine (IP) last stage of the large steam turbine with reaction at mean radius such that provide axial flow exit from the stage and had a twist by the law of free-vortex design. The second design of stage (P2) was purposed for testing the possibility of increase a load, preserving axial flow exit. The main parameters of the stage put on trial are presented in **Table 2.8.1**. In addition to the stage integral characteristics, axisymmetric computations provide the flow parameters distribution in axial gap along radius. At this method of loss components estimation along the radius is a subject of importance. The secondary losses were connected at the blade tip by a special algorithm. The secondary losses were calculated for each station along the blade height fitting a local profile loss magnitude. In summary, the process can be envisioned through **Figure 2.8.2**.

Stage Design	P1	P2
Inlet Pressure, Pa	117000	130000
Inlet Temperature, K	373	373
Outlet Pressure, Pa	100000	100000
Rotation frequency, 1/s	7311	8212
Nozzle vane mean diameter, m	0.2978	0.2978
Nozzle vane length, m	0.0822	0.0822
Blade mean diameter, m	0.2986	0.2986
Blade length, m	0.0854	0.0854
Nozzle vane outlet gauge	20	17.2
Nozzle vane at mean radius	24	17.5
Nozzle vane at peripheral radius	28	17.8
Blade outlet gauging angle near hub	32	41
Blade at mean radius	29.7	26
Blade at peripheral radius	26	19

Table 2.8.1 Main Parameters of the Test Stages (P1) and (P2)

2.8.1.1 Blade Reverse Engineering as applied to Geometry Definition

The airfoil planner shape can be derived by six control points using **NURBS ((Non-Uniform Rotational B-Splines))**, defined as preliminary design. The airfoil geometry is generated on a planner design sections with sections arranged along the blade height following a selected rule²⁸. A turbine designer may choose an approach for profiling, when the sections are profiled along the direction of streamlines. Then, airfoil centroids are placed upon a radial line where a skeleton generated from the section is covered with a surface that is a **NURBS**. In a process of planner sections constructions, a technique of profile shape optimization on the geometry and aerodynamics was applied. The blades that were used in the test turbine stage are the subject of particular interest from several points of view. First of all, the nozzle vane cascades are assembled from the profiles supplied with the trailing edge extensions. This is characterized by heightened strength properties at reasonably high efficiency and low sensitivity to inlet

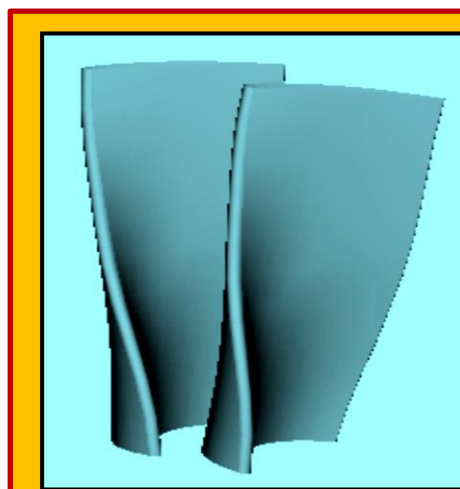


Figure 2.8.3 Profile of Blades

²⁸ Moroz, L., Govoruschenko, Y., Pagur, P., "Proceedings of GT2005 ASME Turbo Expo 2005: Power for Land, Sea and Air", June 6-9, 2005, Reno-Tahoe, Nevada, USA.

flow angle variation. Then, the specialty profiled cascades with divergent channels in hub zone capable of to provide hub reacting at moderate loss were used that permits increase loading. (see [Figure 2.8.3](#)).

2.8.1.2 3D Aerodynamic Computation

Experience proves that any problem solved in 3D formulation which obviating 1D and 2D analyses is fought with a danger of missing in flow rate and efficiency determination, particularly when the shape defined with low accuracy. At the same time, unidirectional and axisymmetric components feature high reliability, high speed of operation and accuracy sufficient for conventional turbine design. 3D analysis is a laborious and sophisticated tool and modeling time invested is several orders of magnitude larger than 1D and 2D model. In addition, the designer needs to process and maintain specialize skilled for mesh generation, turbulence model selection, boundary application, etc. Indeed, all forthcoming of 3D analyses is compensated by its capabilities to quantitatively count the flow nuances such as secondary effect in the cascade and flow separation, which cannot be precisely detected in the low fidelity models.

2.8.2 Case Study 2 - Cooling Air Flow Rate

The flow rate of cooling air through the heat exchangers is obviously a key parameter defining the performance of the system. It comes from two sources – the fans and the ram effect generated by the movement of the vehicle through the atmosphere. As far as the ram air is concerned there is a trade-off between a desired high flow rate for good heat exchanger performance and a low flow rate for minimization of overall vehicle drag. The flow of air through the front end of the vehicle adds a typical 5% to the vehicle drag. There are three basic approaches to the establishment of front end air flow rate under particular operating conditions. At the most sophisticated level complete CFD analyses can be performed which model the detail of the air flow around the outside of the complete vehicle and through the vehicle front end and engine compartment including the various heat exchangers and even through the rotating fans. [Figure 2.8.4](#) shows the CFD analysis of flow through vehicle front-end with streamlines and pressure contours. This method delivers a great deal of information about the system but is demanding in terms of computing effort²⁹. At the intermediate level of complexity commercial software exists that allows networks of 1D components to be set up and the air flow distribution through them to be calculated. This can be valuable when problems arise such as air recirculation or significant temperature and/or flow rate distribution.

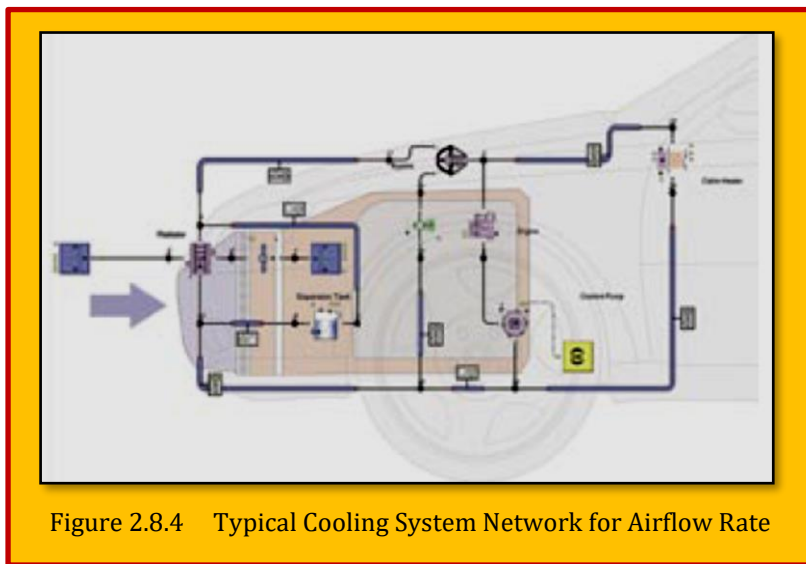


Figure 2.8.4 Typical Cooling System Network for Airflow Rate

2.9 Reduced Order Model Using Empirical Relationship

Despite the sophistication of these development tools there exists a much simpler tool for the prediction of front end air flow that proves to be capable of delivering considerable insight into the

²⁹ Mentor Graphics Corporation 2012.

way the system is behaving. It is based in a 1D model that characterizes the face air flow velocity through the heat exchangers, v_R , in terms of a few non-dimensional constants:

$$v_R = \left[\frac{F v_0^2 + \psi_0 u_0^2}{1 + \zeta_R + \zeta_{sys} + \zeta_F} \right]^{\frac{1}{2}} \quad \text{Eq. 2.9.1}$$

Where:

F measure of the effectiveness of the front end shape in delivering

V_0 vehicle velocity

ψ_0 maximum non-dimensional pressure coefficient of the fan

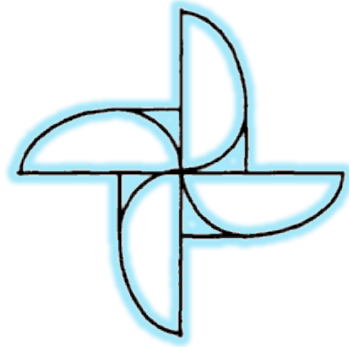
u_0 fan tip speed

ζ_R pressure drop coefficient of the heat exchangers

ζ_{sys} pressure drop coefficient of the remainder of the system (grill, engine compartment, etc.).

ζ_F pressure drop coefficient for the fan itself.

The appropriate values for the unknown system and fan constants ($F, \psi_0, \zeta_{sys}, \zeta_F$) are determined from wind tunnel measurements of air flow rates through the heat exchangers for ranges of different vehicle speeds and fan speeds. The pressure drop coefficients of the heat exchangers as functions of air flow rate are already known. The values of the system constants are extracted from the experimental dataset using non-linear *optimization* techniques. Knowledge of the values of these parameters for a system allows the air flow rate through the heat exchangers to be explored for any vehicle speed or fan speed and even allows the effects of different heat exchangers to be evaluated³⁰.



³⁰ See previous.

3 Mesh Free Methods for CFD

While algorithms have seen great advances in CFD, mesh generation methods has lagged behind, creating a computational bottleneck. For industry and government looking to impact current and future products with simulation technology, mesh generation imposes great challenges. Many generation procedures lack automation, requiring many man-hours, which are becoming far more expensive than computer hardware. More automated methods are less reliable for complex geometry with sharp corners, concavity, or otherwise complex features. Most mesh generation methods to date require a great deal of use expertise to achieve proper stretching, resolution, and structure³¹. The motivation behind meshless methods lies in releasing the burden of mesh generation. Since the application of computational methods to real world problems appears to be paced by mesh generation, alleviating this bottleneck potentially impacts an enormous field of problems. It is not clear at this point how effective meshless methods will be at alleviating meshing problems. While a rigid mesh is not required, sufficiently dense point distributions are still required. Moreover, points must be grouped locally to form clouds. Obtaining optimal clouds for different methods is also a non-trivial problem. However, recent progress in the area of point distribution and cloud generation by L'ohner and others ^{32,33} has shown great promise in this area. Several of the most notable meshless methods are:

- Smooth Particle Hydrodynamics (SPH)
- Mesh free Local Petrov-Galerkin (MLPG)
- Methods based on Radial Basis Functions (RBF)
- Finite Point Methods (FPM)
- Mesh free Boundary schemes
- Reproducing Kernel Particle Method (RKPM)

These methods are also summarized in works by Liu³⁴ and Liu and Gu³⁵.

3.1 Smooth Particle Hydrodynamics (SPH)

The method of SPH, introduced by Monaghan³⁶, makes use of an integral representation of a function at a point given a set of surrounding points, called a kernel approximation. It uses no mesh, and points are free to move past one another consistent with a Lagrangian approach. While SPH was first developed to handle astrophysical phenomena in open space, the method was later applied to structures, fracture simulation, fluid flow, and other fields. Monaghan ³⁷ showed that the SPH method with artificial viscosity could accurately capture shock waves in one-dimensional shock tube problems. Methods based on an SPH formulation are well-suited for problems of infinite domain in which the problem size is not known in advance.

³¹ Aaron Jon Katz, "Meshless Methods for Computational Fluid Dynamics", A dissertation submitted to the department of aeronautics and astronautics and the committee on graduate studies of Stanford university in partial fulfillment of the requirements for the degree of doctor of philosophy, January 2009.

³² R. L'ohner and E. O'neate. An advancing front point generation technique. Communications in Numerical Methods in Engineering, 14:1097–1108, 1998.

³³ R. L'ohner, C. Sacco, and E. O'neate. A general advancing front technique for filling space with arbitrary objects. Int. J. Numerical Meth. Engineering, 61:1977–1991, 2004.

³⁴ G. R. Liu. Mesh Free Methods: Moving Beyond the Finite Element Method. CRC Press, 2003.

³⁵ G. R. Liu and Y. T. Gu. An Introduction to Mesh free Methods and Their Programming. Springer, 2005.

³⁶ J. J. Monaghan and R. A. Gingold. "Shock simulation by the particle method SPH", Journal of Computational Physics, 52:374–389, 1983.

³⁷ See previous.

While SPH has become popular for intensely dynamic problems in which a static or even dynamic mesh may not properly resolve relevant physics, certain implementation difficulties are inherent in the method. These difficulties include the selection of a proper domain of influence with weighting functions, efficient nearest neighbor particle searching, and the determination of a smoothing length for force computations at each particle. The Reproducing Kernel Particle Method (RKPM), introduced by Liu, Jun and Zhang³⁸, is very similar to the SPH method in that it uses a finite integral representation to discretize the governing PDEs. However, RKPM adds a correction function to the base kernel approximation, improving the accuracy especially near boundaries³⁹. The RKPM method has been applied to fluids, structures, and acoustics. Lesoinne and Kaila⁴⁰ used RKPM to compute aero elastic effects of aircraft with large control surface deflections. Zhang, Wagner, and Liu⁴¹ showed that RKPM was well suited for domain decomposition for large-scale parallel computing.

3.1.1 Mesh free Local Petrov-Galerkin

The MLPG method has arisen from the finite element community and is based on the weak form of a given PDE. While the use of the weak form of PDEs relaxes consistency requirements of field variable approximation, many algorithms in CFD bypass the rigorous use of weak forms. Weak forms require the use of numerical integration since they satisfy global integral forms of the governing equations. Numerical integration, along with other rigorous aspects of weak forms makes them computationally inefficient compared with simple FDM or FVM approaches. Jameson⁴² showed the equivalency of one FVM scheme with a Galerkin method, most development in CFD has been based on strong forms of the governing equations, which lead to simple and efficient conservative schemes. Nonetheless, an immense mathematical foundation has been developed based on weak forms used for a variety of FEM applications.

Developed by Atluri and others⁴³⁻⁴⁴, the MLPG method is based on a Petrov-Galerkin formulation in which weight and trial functions used in the weak form of the equations need not be the same. This gives the method a “local” nature in which the integral in the weak form is satisfied over a local domain. The MLPG method thus requires a local “background grid” to perform the integral as demanded by the weak form. However, the integral is performed locally, relieving the need for a global background integration as is used in related methods. The local background grid may be simple shapes, such as circles or squares. By all practical measures, MLPG is essentially meshless. Approximation of the field variables for the MLPG method is constructed using a moving least squares approach. Least squares representations of a function do not pass through the discrete sampling points of the function. Instead, they construct a smooth representation which minimizes the error of approximation. This fact has posed some difficulties in obtaining accurate and stable boundary conditions for the MLPG approach. The MLPG scheme is very general and has been applied to various problems. Specific to fluid mechanics, it have used MLPG to solve the incompressible

³⁸ W. K. Liu, S. Jun, and Y. F. Zhang. “*Reproducing kernel particle methods*”, International Journal for Numerical Methods in Fluids, 20:1081–1106, 1995.

³⁹ F. C. Gunther and W. K. Liu. “*Implementation of boundary conditions for meshless methods*”, Computer, Methods Appl. Mech. Engineering, 163:205–230, 1998.

⁴⁰ M. Lesoinne and V. Kaila. “*Meshless aero-elastic simulations of aircraft with large control surface deflections*”, AIAA paper 2005-1089, AIAA 43rd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2005.

⁴¹ L. T. Zhang, G. J. Wagner, and W. K. Liu. “*A parallelized mesh free method with boundary enrichment for large-scale cfd*”, Journal of Computational Physics, 176:483–506, 2002.

⁴² A. Jameson, T. J. Baker, and N. P. Weatherill. “*Calculation of inviscid transonic flow over a complete aircraft*”, AIAA paper 1986-0103, AIAA 24th Aerospace Sciences Meeting, Reno, NV, January 1986.

⁴³ S. N. Atluri and T. Zhu. “*A new meshless local petrov-galerkin (mlpg) approach in computational mechanics*”, Computational Mechanics, 22:117–127, 1998.

⁴⁴ S. N. Atluri, H. G. Kim, and J. Y. Cho. “*A critical assessment of the truly meshless local petrov-galerkin (mlpg) and local boundary integral equation (lbie) methods*”, Computational Mechanics, 24:348–372, 1999.

Navier-Stokes equations used an up winding scheme for stabilization of the convection operator in the stream wise direction.

3.1.2 Mesh free Methods Based on Radial Basis Functions

Radial basis functions are functions which have no preferred direction, but only depend on norms in space. Most often, the Euclidean distance is used as the norm. Common RBFs include Gaussians, thin plate splines, and multi-quadrics. In general, RBFs are smooth and continuously differentiable. When used for interpolation purposes, RBF approximations are constructed such that they pass through data points exactly. It is difficult to prove any order of accuracy of such approximations since RBFs are not based on Taylor series or polynomial expansions. While RBFs have been widely used in scattered data interpolation, their application to the solution of PDEs is relatively new. The symmetric and un-symmetric forms were compared independently by and compared an RBF method to the finite element method in terms of accuracy and efficiency, showing improved accuracy of the RBF method over FEM. Sharan has used the popular multi-quadric RBFs to solve elliptic PDEs. In a similar work, [Sarler] formulated a solution method for diffusion problems based on RBFs. In a more general work, integrated the theory of Galerkin methods with radial basis functions. More recently, [Divo and Kassab] have used RBFs to model convective viscous flows and heat transfer problems. [Chinchapatnam] has used a localized RBF method to compute incompressible viscous flows. Radial basis methods for compressible flows are much less common, however Shu has recently proposed such a method based on an upwind approach.

3.1.3 Finite Point Methods

By far, the most prevalent meshless schemes for CFD have been the so-called *finite point methods*. Finite point methods are usually based on the strong form of the governing PDEs and have given rise to several variants. In general, FPMs are based on least squares fitting of functions to discrete points. These approximate functions form the basis of discretization methods for PDEs. Least squares techniques have been widely used in traditional CFD methods as a means of reconstructing high order solutions, as discussed by [Mavriplis]⁴⁵. However, the use of least squares as the primary mechanism for PDE discretization in the meshless sense is relatively new. Finite point methods were originally derived as generalizations of FDM for irregular point distributions by [Chung]⁴⁶.

Finite point methods may be categorized into two main classes: methods derived from *Taylor series*, and methods based on *polynomial basis functions*. Actually the Taylor series approach is a specific case of a polynomial method in which the approximated function is constrained to pass through the local cloud center. The Taylor approach is intuitive and has formed the basis for many schemes, including the ***Least Squares Kinetic Upwind Method (LSKUM)***. Other approaches based on Taylor series expansions includes the order of accuracy of the Taylor method for an upwind scheme. The methods based on polynomial basis functions are equally numerous as the Taylor based methods. [Batina] was one of the first to use a polynomial basis in conjunction with least squares to compute derivatives for the Euler and Navier Stokes equations. He used an unweighted least squares approach. A similar method was proposed a few years later by [Liu and Su]. Others developed a more rigorous method based on polynomial basis functions. Their method incorporated different least squares weighting methods to improve the accuracy of derivatives and formulations for higher order methods. They applied their method to subsonic compressible inviscid and viscous flows. [Löhner and others] extended the method of Oñate to compressible aerodynamic applications with shocks in three dimensions. They implemented their scheme with the van Leer approximate Riemann solver , gradient reconstruction for high resolution, and limiters to capture shocks.

⁴⁵ D. J. Mavriplis. "Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes", AIAA paper 2003-3986, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.

⁴⁶ K. C. Chung. "A generalized finite-difference method for heat transfer problems of irregular geometries", Numerical Heat Transfer, 4:345-357, 1981.

3.1.4 Meshless Boundary Schemes

Many of the methods discussed above have been used to enforce boundary conditions for embedded boundary systems. Embedded boundaries arise with the use of nobody-conforming grids, such as Cartesian grids. Meshless methods have been used in place of cut cells and other related methods. One of methods used is polynomial least squares method to compute inviscid slip boundary conditions using embedded Cartesian meshes. They presented encouraging results for two and three dimensional inviscid test cases. It has been implemented meshless embedded boundary conditions for high Reynolds number viscous flows using the concept of a sub-grid to resolve boundary layers. The sub-grid adds additional resolution near the surface, providing points on which to perform meshless computations. All these methods appear to provide attractive alternatives to Cartesian cut cells or other methods of embedded boundary conditions⁴⁷.

3.2 Solution Procedure for Mesh free Methods ⁴⁸

The procedure of mesh free methods consists of four basic steps:

- Domain representation
- Function approximation
- Formation of system equations
- Solving the global equations

3.2.1 Domain representation

First, the domain and its boundary is modeled (not discretized) using sets of arbitrarily distributed nodes (see **Figure 3.2.1**) in the domain and its boundary. The nodal distribution is usually not uniform. The density of nodes depends on the accuracy requirement of the analysis. Because the nodes carry the values of a field variable (e.g. density, velocity, etc.), they are often called **field nodes**. Further in the text, a field variable will be referred to as a **field function**.

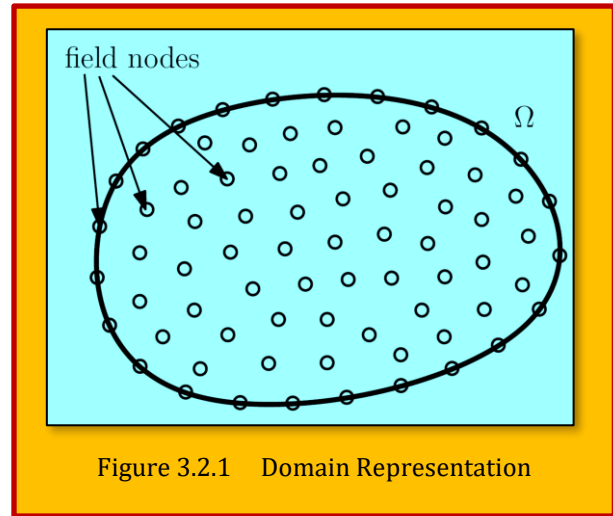


Figure 3.2.1 Domain Representation

3.2.2 Function Approximation

The field function u at any point at $x = (x, y)$ within the domain is approximated using the values at its nodes within the “small” local domain of the point x , i.e.

$$u(x) = \sum_{i=1}^n \phi_i(x) u_i$$

Eq. 3.2.1

Where n is the number of nodes included in a local domain of the point at x , u_i is the nodal field function at the i th node in the local domain, and $\phi_i(x)$ is the shape function of the i th node. The “small” local domain of x will be called the support domain of x and denoted Ω_x . The size of support domain defines the number of field nodes approximating x . Some possible shapes of support domains are shown in **Figure 3.2.2** where spherical is the most common one.

⁴⁷ For excellent survey of literature in “*Meshless methods*”, see 77.

⁴⁸ P. Niedoba, L. Čerma, and M. Jícha, Meshfree methods for computational fluid dynamics, EPJ Web of Conferences 45 01068 (2013), DOI: 10.1051, epjconf/201345 01068.

3.2.3 Formation of System Equations

System equations can be formulated using the shape functions and strong or weak formulation⁴⁹. These equations are assembled into the global system matrices for the entire problem domain. For static problems, the global system equations are a set of algebraic equations. For general dynamics problems, it is a set of differential equations.

3.2.4 Solving the Global Equations

The last step depends on the type of equations (algebraic, differential, etc.). Note that the global equations for computational fluid dynamics problems are basically nonlinear.

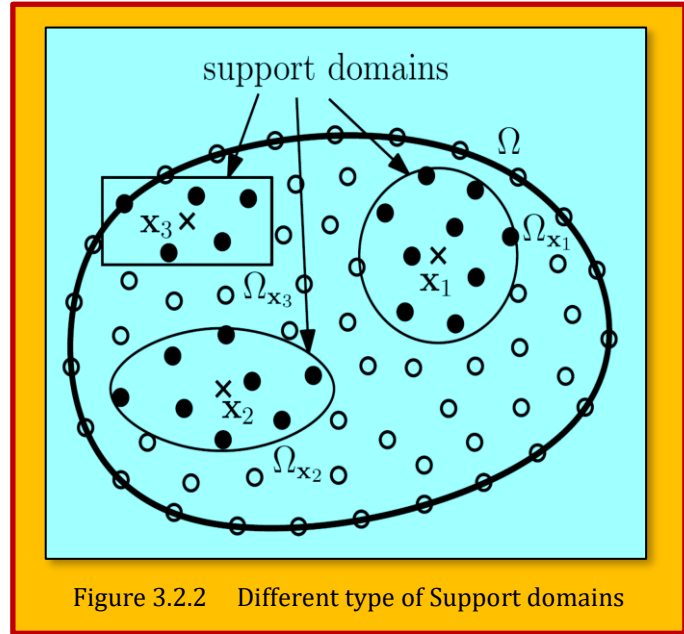


Figure 3.2.2 Different type of Support domains

3.3 Method of Smooth Particle Hydrodynamics (SPH)

The smoothed particle hydrodynamics method belongs to basic mesh free methods. It is used for solving partial differential equations. The SPH is basically an interpolation method. The interpolation is based on the theory of integral interpolants using kernels that approximate a delta function. The fluid mass is lumped into smoothed blobs that are moved using Newton's second law directly, without an underlying mesh. In SPH the fluid is modeled as a collection of smooth "blobs" or particles as depicted in **Figure 3.3.1**. A system of ordinary differential equations is produced after approximation of unknown functions (field function) and their spatial derivatives. This system is most often solved by explicit numerical methods.

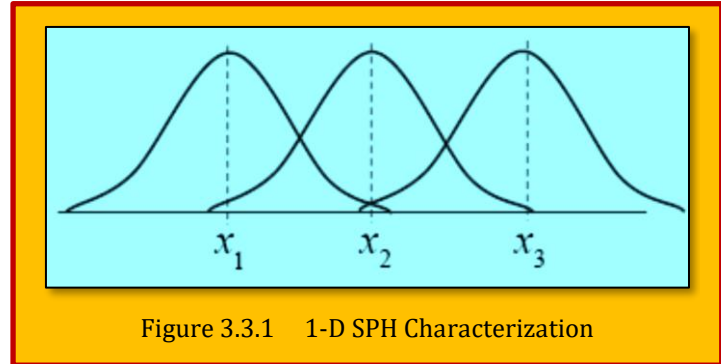


Figure 3.3.1 1-D SPH Characterization

3.3.1 Formulation

Function approximation of the field function $u(x)$ is based on an integral representation of the function and is given by the equation

$$\langle f(x) \rangle = \int_{\Omega_x} f(\xi) W(x - \xi, h) d\xi$$

Eq. 3.3.1

Where $W(x-\xi, h)$ is the weight function (i.e., smoothing function, kernel function), h being the smoothing length, which defines the size of the support domain Ω_x , i.e. the smoothing length determines the number of particles approximating the function at x . **Eq. 3.3.1** is usually referred to as kernel approximation, or SPH approximation of function $f(x)$. For practical calculation, **Eq. 3.3.1** must be discretized as follows

⁴⁹ See 94.

$$\langle f(x) \rangle = \sum_{j=1}^n f(\xi_j) V_j W(x - \xi_j, h) = \sum_{j=1}^n \frac{m_j}{\rho_j} f(\xi_j) W(x - \xi_j, h) \quad \text{Eq. 3.3.2}$$

Where m_j and ρ_j are mass and density of the j th particle in Ω_x (i.e., $V_j = m_j/\rho_j$ is the volume of j - particle). **Eq. 3.3.2** is called a particle approximation of field function $f(x)$. Note that the approximation (**Eq. 3.3.2**) corresponds to the approximation introduced for a general mesh free method. The shape function in this case has the form of

$$\varphi_j(x) = W(x - \xi, h)_j \frac{m_j}{\rho_j} \quad \text{Eq. 3.3.3}$$

Approximation of the spatial derivatives of the field function can be obtained by replacing the function $f(x)$ in **Eq. 3.3.1** with its spatial derivative $\nabla f(x)$. Using the per-parts, the Green theorem and a discretization we obtain a particle approximation of the spatial derivative of the field function in the form of

$$\langle \nabla f(x) \rangle = \sum_{j=1}^n f(\xi_j) \nabla_x W(x - \xi_j, h) \frac{m_j}{\rho_j} \quad \text{Eq. 3.3.4}$$

Where $\nabla_x W(x - \xi_j, h)$ is the spatial derivative of the weight function with respect to the variable x . We can observe that an approximation of the spatial derivative of a field function is determined using only field function values and derivatives of the weight function. In the same fashion we obtain the Laplacian as:

$$\langle \nabla^2 f(x) \rangle = \sum_{j=1}^n f(\xi_j) \nabla_x^2 W(x - \xi_j, h) \frac{m_j}{\rho_j} \quad \text{Eq. 3.3.5}$$

3.3.2 Smoothing Kernels

The use of *different kernels in SPH is analogue to using different difference schemes in finite difference methods*, thus the choice of smoothing kernel for a specific problem is significant. The derivatives of the smoothing kernels have an important impact for different SPH estimations, but we will now focus on the kernels and their required properties. It is required that a suitable kernel must have the following two properties,

$$\int_{\Omega_x} W(x - \xi, h) d\xi = 1 \quad \text{and}$$

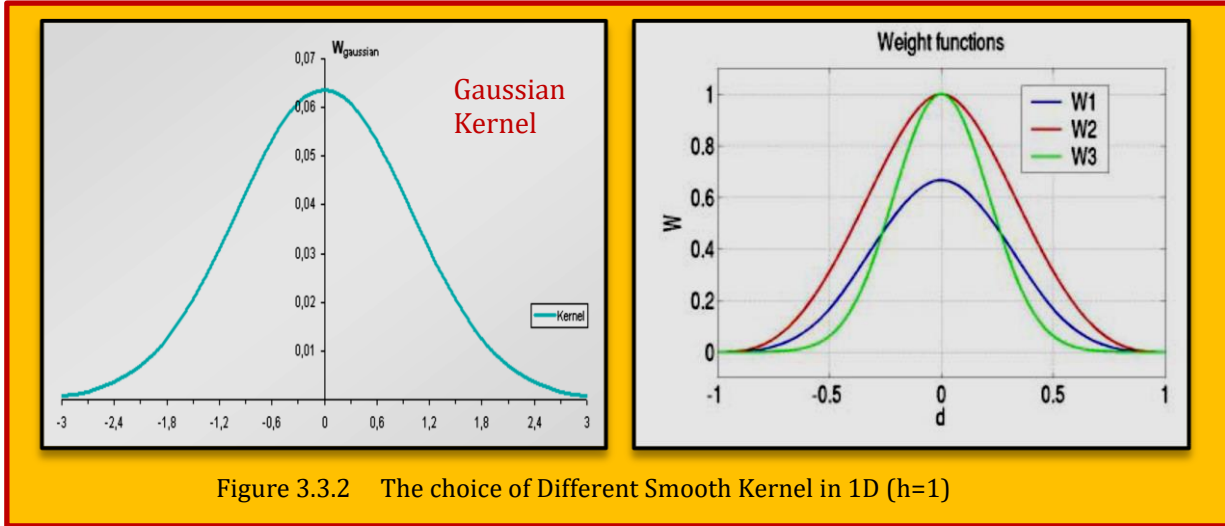
$$\lim_{h \rightarrow 0} W(x - \xi, h) = \delta(x - \xi) = \begin{cases} \infty & \|x\| = 0 \\ 0 & \text{otherwise} \end{cases}$$

Eq. 3.3.6

Eq. 3.3.6 states that the kernel must be normalized, and that the unit integral ensures that maxima and minima are not enhanced. The kernel must also be positive to ensure that it is an averaging function. If the kernel is even, then rotational symmetry is enforced, which is useful to ensure invariance under rotations of the coordinate system.

$$W(x - \xi, h) \geq 0 \quad \text{and} \quad W(x - \xi, h) = -W(x - \xi, h) \quad \text{Eq. 3.3.7}$$

If these conditions are met, the interpolation is of second order accuracy⁵⁰ that is the error of approximating is 2nd order or better. It is also suggested that a suitable kernel should have a limited or compact support radius, in order to ensure zero kernel interactions outside the computational



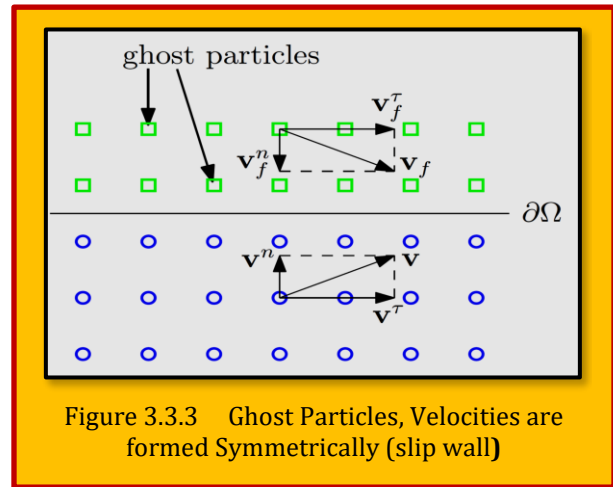
range of the radius. We use the kernel width h as the compact support radius for all smoothing kernels, which implies $W(x - \xi, h) = 0$, $r > h$. The first golden rule of SPH states that if a new interpretation of an SPH equation is to be found, it is always best to assume the kernel is a Gaussian⁵¹. The isotropic Gaussian kernel in n dimensions is given by

$$W(x - \xi, h) = \frac{1}{(2\pi h^2)^{3/2}} e^{-\left(\frac{\|r\|^2}{2h^2}\right)}, \quad h > 0 \quad \text{Eq. 3.3.8}$$

Which is depicted in **Figure 3.3.2 - left**. Even though a Gaussian kernel has very nice mathematical properties, it is not always the best kernel to use, e.g. it does not have a compact support for our purpose, and it requires the evaluation of the expensive exponential function. There are other choices of kernels such as $W1$ = piece wise cubic spline, $W2$ = quadratic spline, and $W3$ = exponential function which also shown in **Figure 3.3.2 - right**, where $d = (\xi - \xi_i)/h$.

3.3.3 Updating of Smoothing Length h

To update h , we can use either the constant or as a variable.



⁵⁰ J. J. Monaghan. "Smoothed Particle Hydrodynamics". *Annual Review of Astronomy and Astrophysics*, 30, pp. 543-574, 1992.

⁵¹ See above.

3.3.3.1 Constant

- h too small, n too small, results no accurate
- h too big, local information smoothed out

3.3.3.2 Variable

- known at the beginning: h_i^0
- updated solving:

$$\frac{Dh_i^n}{Dt} = -\frac{h_i^n}{\rho_i^n N} \frac{D\rho_i^n}{Dt} = -\frac{h_i^n}{\rho_i^n N} \sum_{j=1}^n m_j \left(\frac{m_i^n}{\rho_i^n} - \frac{m_j^n}{\rho_j^n} \right) \cdot \nabla_i^n W(\xi_i - \xi_j) \quad \text{Eq. 3.3.9}$$

Where N is number of dimensions. It is ok for slow varying density, more complicated procedure or fast expansion/contraction (e.g. in gases)⁵².

3.3.4 Boundary Treatment

The issue of boundary conditions is generally very difficult in the SPH method. We answer the question of properly defining the boundary condition that prevented particles from escaping out of the domain. Furthermore, we discuss consistency near the boundary of the domain (*near boundary area*)

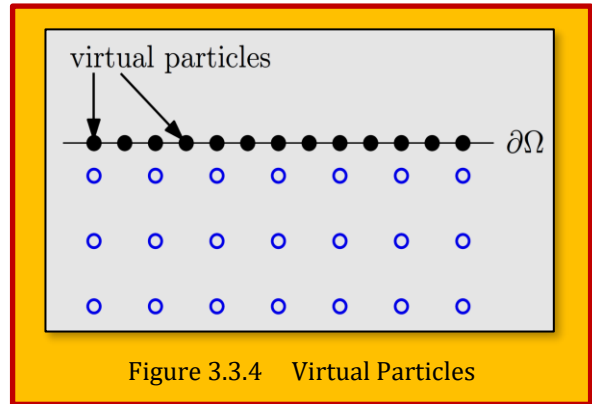


Figure 3.3.4 Virtual Particles

3.3.5 Virtual Particles

The first approach is the use of virtual particles. These particles are situated on the boundary and by repulsive force acting on the particles in the near boundary area (*near boundary particles*). Hence, virtual particles prevent an unphysical penetration through the boundary. (see [Figure 3.3.4](#)). Unfortunately, this approach violates the condition for C^1 consistency of the SPH approximation in the near boundary area. This fact is due to the undesirable “cutting off” of the weight function support, see [Figure 3.3.5](#). Thus, the appropriate weight function is not an even function⁵³.

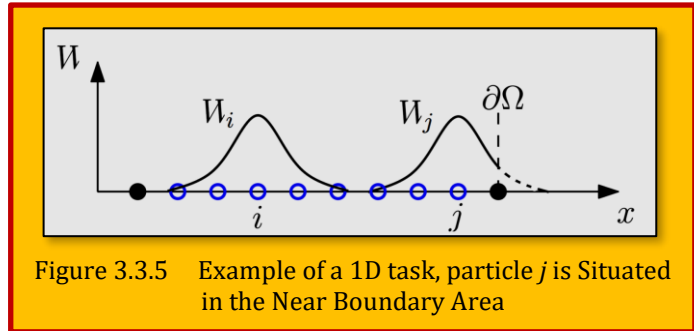


Figure 3.3.5 Example of a 1D task, particle j is situated in the Near Boundary Area

3.3.6 Ghost Particles

A much better way is to use ghost particles as a boundary condition. In contrast to virtual particles, this approach creates a dynamic wall that is constructed at each time step. Ghost particles are formed symmetrically (according to the boundary) to the near boundary particles as “twin” particles, see [Figure 3.3.3](#). Using ghost particles ensures C^1 consistency of the SPH approximation, because the shape functions of the near boundary particles can be even functions.

⁵² Remo Minero, “Mesh Free Methods for Fluid Dynamics Problems”, 17 Dec, 2003.

⁵³ P. Niedoba, L. Čerma, and M. Jícha, Meshfree methods for computational fluid dynamics, EPJ Web of Conferences 45 01068 (2013), DOI: 10.1051, epjconf/201345 01068.

3.3.7 Summery and Recap⁵⁴

- Smoothed particle hydrodynamics is an interpolation method that can approximate continuously field quantities and their derivatives by using discrete sample points, called smoothed particles.
- Particles carry mass, m , position, x , and velocity, u , but can also hold SPH estimated quantities, e.g. mass-density, ρ , pressure, p , etc.
- The following relation between volume, mass, and mass-density applies, and can be used to determine the volume occupied by a particle, $V=m/\rho$.
- The following properties must hold for a smoothing kernel: being normalized, positive and even.
- We only use smoothing kernels with a compact support radius h . The basis formulation of SPH to approximate any quantity field and their derivatives. SPH is originally designed for compressible flow problems.

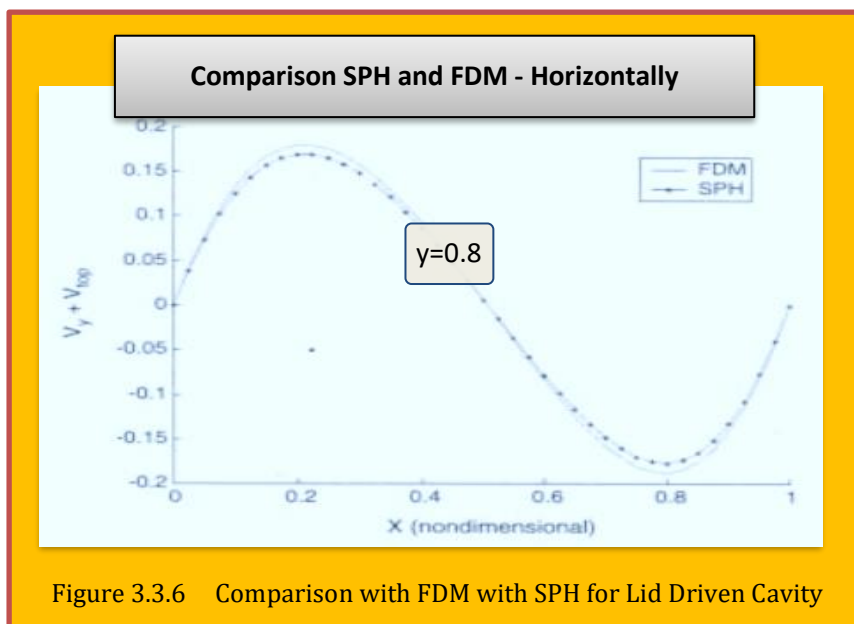


Figure 3.3.6 Comparison with FDM with SPH for Lid Driven Cavity

Readers are encourage to consult [Liu & Liu]⁵⁵ for detailed information and recent trends in SPH methodology.

3.3.8 Case Study 1 - Lid Driven Cavity Problem

To validate, the bench mark case of the lid driven cavity is considered and the results are compared with FDM on the same number of particles for $Re = 10$ and 41×41 particles (See [Figure 3.3.6](#)).

3.3.9 Case Study 2 - Two-dimensional Convection-Diffusion Problem

A meshless Local Method of Approximated Particular Solutions (LMAPS) is used to analyze problem described by the convection diffusion equation by [Mužík & Holíčková]⁵⁶. The method solves the steady convection-diffusion equation with reaction term. The discretized system of equations is derived via interpolation procedure and radial basis functions (RBF). The solution of the equation is performed over simple geometry with non-uniform velocity field and results are presented in the article. The LMAPS method is capable to produce stable solutions with results comparable to the analytical solutions.

⁵⁴ Micky Kelager, "Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics", January 9, 2006.

⁵⁵ M.B. Liu · G.R. Liu, "Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments", Arch Comput Methods Eng (2010) 17: 25–76, DOI 10.1007/s11831-010-9040-7.

⁵⁶ Juraj Mužík, and Martina Holíčková, "Two-dimensional convection-diffusion problem solved using method of localized particular solutions", MATEC Web of Conferences · January 2017.

The local method of approximated particular solution (LMAPS) was proposed by [Cheng et al.]⁵⁷ and was applied to elliptic problems and non-linear problems⁵⁸. In LMAPS the domain is covered by cloud of scattered nodes. In the work on LMAPS reported so far, the support of the any computational node is taken to be a simple subdomain in a shape of a circle though in theory the domain can be of any shape, with the computational node in the center of the circle. The most often used interpolation for field variables were the moving least-squares, though some researchers used different schemes for interpolation of the field variable and gradients over the circular boundaries. The area of interest Ω with the boundary $\partial\Omega$ is covered

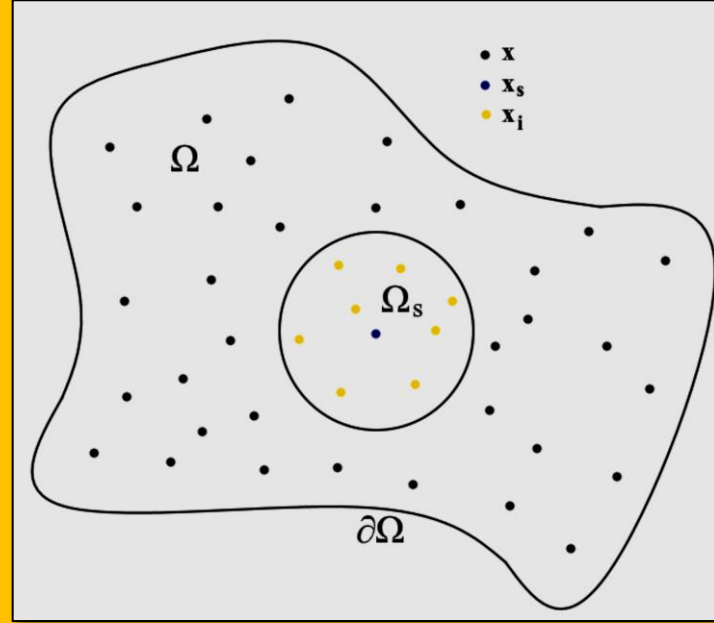


Figure 3.3.7 The diagram of global domain Ω , local support domain Ω_s of point x_s , global points x and local point x_i

by points within the area and also on the global boundary (see [Figure 3.3.7](#)). Consider a local circular (or any simple shape e.g. rectangle) sub-domain Ω_s centered at every point s . This sub-domain is called support domain and using the points in a particular support domain any function can be expressed using just nodal values⁵⁹.

3.4 RKPM Method

The reproducing kernel particle method belongs to the category of finite integral methods, and is a modification of the SPH method. This method adds the so-called correction function to the SPH formulation to ensure certain order of consistency. The particle approximation of the function $f(x)$ is defined

$$\langle f(x) \rangle = \sum_{j=1}^n \frac{m_j}{\rho_j} f(\xi_j) C(x, \xi_j) W(x - \xi_j, h)$$

where $C(x, \xi_j)$ is correctionfactor

Eq. 3.4.1

3.5 Lagrangian Description of Fluid Dynamics Using SPH

Interactive fluid dynamics is of essential interest in real-time applications, such as computer games or virtual surgery simulators. Using the smoothed particle hydrodynamics (SPH) method, a stable particle-based approach to solve the motion of interactive fluids using Lagrangian description. With focus on the simulation part we provide a thorough insight of the mathematical theory of particle-

⁵⁷ C.S. Chen, C.M. Fan, P.H. Wen, "Numerical Methods for Partial Differential Equations", 28, 506–522, (2012).

⁵⁸ C.S. Chen, M.A. Golberg, M. Ganesh, "A.H.-D. Computers and Mathematics with Application", 359–378, (2002)

⁵⁹ See 105.

based fluids. The basic Eulerian formulation of an incompressible, isothermal fluid for 2-D flow with constant properties express as

$$\nabla \cdot \mathbf{u} = 0 \quad , \quad \rho \left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \mathbf{u} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{u}) + \mathbf{f} \quad \text{Eq. 3.5.1}$$

Where μ is the viscosity of the fluid, and $\mathbf{f}$ is the sum of external force-densities acting on the fluid, e.g. gravity. Using particles instead of a grid simplifies the equations significantly. We assume that the amount of particles is constant during the simulation, and by keeping the mass fixed for each particle, it implies that mass conservation is guaranteed, and that conservation of mass can be omitted.

Figure 3.5.1 depicts a basic layout of a particle-based fluid, which has been reduced to two-dimensions for reasons of clarity. The particles are represented by the dots. The circles represent the volume of each particle. In the Lagrangian formulation of a fluid the particles completely define the fluid, which implies that the particles move with the fluid. Compared to the Eulerian view this means that any field quantity now depends on time, t , only. The particles carry mass, position, and velocity, and will hold smoothed quantity approximations obtained from SPH. The acceleration for a Lagrangian fluid particle becomes the ordinary time derivative of its velocity. This is why the total derivative term (D/Dt) is reduced to a simple d/dt in the Lagrangian view. The basic Lagrangian formulation of the Navier Stokes equations for an incompressible, isothermal fluid is given by

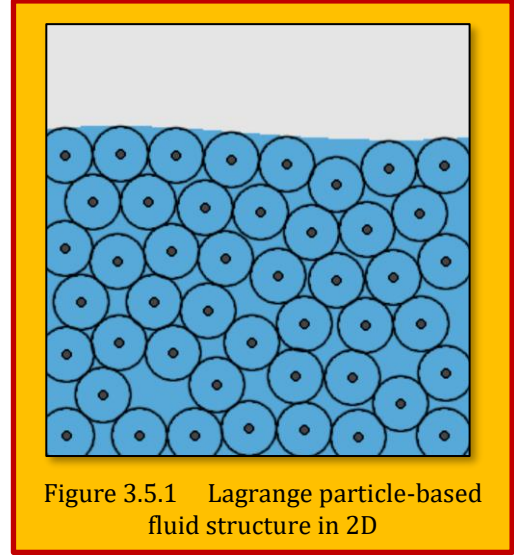


Figure 3.5.1 Lagrange particle-based fluid structure in 2D

$$\rho \frac{d\mathbf{u}}{dt} = \underbrace{-\nabla p + \mu \nabla^2 \mathbf{u}}_{\mathbf{f}_{\text{internal}}} + \underbrace{\mathbf{f}}_{\mathbf{f}_{\text{external}}} \quad , \quad \mathbf{F} = \mathbf{f}_{\text{internal}} + \mathbf{f}_{\text{external}} \quad , \quad \mathbf{a}_i = \frac{d\mathbf{u}_i}{dt} = \frac{\mathbf{F}_i}{\rho_i}$$

Eq. 3.5.2

Where $\mathbf{a}_i$ is the particle acceleration, $\mathbf{f}_{\text{internal}}$ denotes to pressure and viscous forces, and $\mathbf{f}_{\text{external}}$ assigned to gravity.

3.5.1 Default Kernel

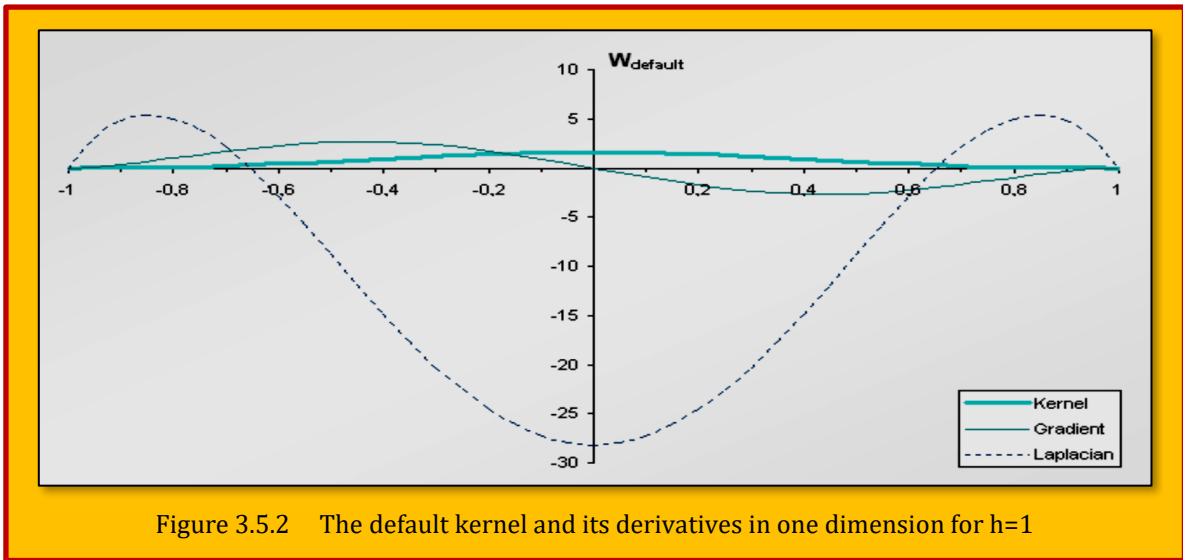
We learned about the first golden rule of SPH, and we also concluded that the isotropic Gaussian kernel was not fit to be used for our purpose. We need a default smoothing kernel with compact support for the inter-particle-based SPH computations required to solve. Several suggestion discussed in ⁶⁰ for SPH kernels. Among them are the B-Spline and Q-Spline kernels, where the Q-Spline is concluded to be the best kernel in terms of computational accuracy. However, the Q-Spline kernel requires the evaluation of the square root, which can be expensive if the kernel is often used. Instead we will use the 6th degree polynomial kernel suggested by⁶¹ as default kernel, which is given by

⁶⁰ J. Hongbin and D. Xin. "On criterions for smoothed particle hydrodynamics kernels in stable field". *Journal of Computational Physics*, 202, pp. 699–709, 2005.

⁶¹ M. Müller, D. Charypar, and M. Gross. "Particle-Based Fluid Simulation for Interactive Applications". *Proceedings of 2003 ACM SIGGRAPH Symposium on Computer Animation*, pp. 154-159, 2003.

$$\begin{aligned}
 W_{\text{default}}(\mathbf{x}, h) &= \frac{315}{64\pi 4^9} \begin{cases} (h^2 - \|\mathbf{x}\|^2)^3 & 0 \leq \|\mathbf{x}\| \leq h \\ 0 & \|\mathbf{x}\| > h \end{cases} \\
 \nabla W_{\text{default}}(\mathbf{x}, h) &= \frac{945}{32\pi 2^9} \mathbf{x} (h^2 - \|\mathbf{x}\|^2)^2 \\
 \nabla^2 W_{\text{default}}(\mathbf{x}, h) &= \frac{945}{32\pi 2^9} (h^2 - \|\mathbf{x}\|^2) (3h^2 - 7\|\mathbf{x}\|^2)
 \end{aligned}
 \tag{Eq. 3.5.3}$$

The default kernel and its derivatives in one dimension can be depicted as:



The default kernel and its derivatives are used for all smoothed quantity field approximations, except for the internal fluid force fields. For further information regarding various smoothing kernel and its application, please consult⁶².

3.5.2 Numerical Time Integration

To simulate the fluid flow, each particle is advanced through time using a global fixed time step Δt , [Eq. 3.5.4](#) is employed to compute the particle acceleration, and the new particle position is obtained from integrating the acceleration numerically. In this section three different integration schemes will be introduced.

3.5.2.1 The Implicit Euler Scheme

The Implicit Euler scheme is actually a semi-implicit method, as it is only the position update that is implicit. Semi-implicit Euler is based on the explicit Euler scheme, which probably is the most common integration method. In explicit Euler the position and velocity are updated in parallel. The semi-implicit Euler is no longer independent of the position and velocity updates as

⁶² Micky Kelager, "Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics", January 9, 2006.

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \mathbf{u}_{t+\Delta t}$$

Eq. 3.5.4

3.5.2.2 The Verlet Scheme

The velocity update is the same, but the position update uses the result from the velocity update to predict the new position,

$$\mathbf{x}_{t+\Delta t} = 2\mathbf{x}_t - \mathbf{x}_{t-\Delta t} + \Delta t^2 \mathbf{u}_t \quad \text{Eq. 3.5.5}$$

The Verlet scheme is one of the computationally fastest integrators and it is usually very stable, as the velocity is given implicitly and will not get out of sync with the position. However, collision responses are not trivial to handle, as it includes modifying positions rather than velocities.

3.5.2.3 The Leap-Frog Scheme

The leap-frog integration has got its name from the fact that the velocities leap over the positions, and vice versa, as illustrated in [Figure 3.5.3](#) where the horizontal line represents time t , and the subscripts on the positions and velocities $\mathbf{u}$ indicate the specific time. The integration structure is implicit Euler and yields to:

$$\mathbf{u}_t \approx \frac{\mathbf{u}_{t-\Delta t/2} + \mathbf{u}_{t+\Delta t/2}}{2}$$

where $\mathbf{u}_{t+\Delta t/2} = \mathbf{u}_{t-\Delta t/2} + \Delta t \mathbf{a}_t$ and $\mathbf{u}_{-\Delta t/2} = \mathbf{u}_0 - \frac{1}{2} \Delta t \mathbf{a}_0$

Eq. 3.5.6

In theory, a time integration scheme will follow Newton's 1st law, but numerical dissipation can reluctantly damp the linear motion of the particles. Typically, this is not a problem in physics-based animation, because the damping can be explained as a small scale air resistance or friction. Especially the Verlet scheme is easily influenced by numerical damping. We have chosen not to introduce any explicit damping in the time integrators, due to the different ways integrators handle damping. We rely on the viscosity force to provide the necessary numerical damping⁶³.

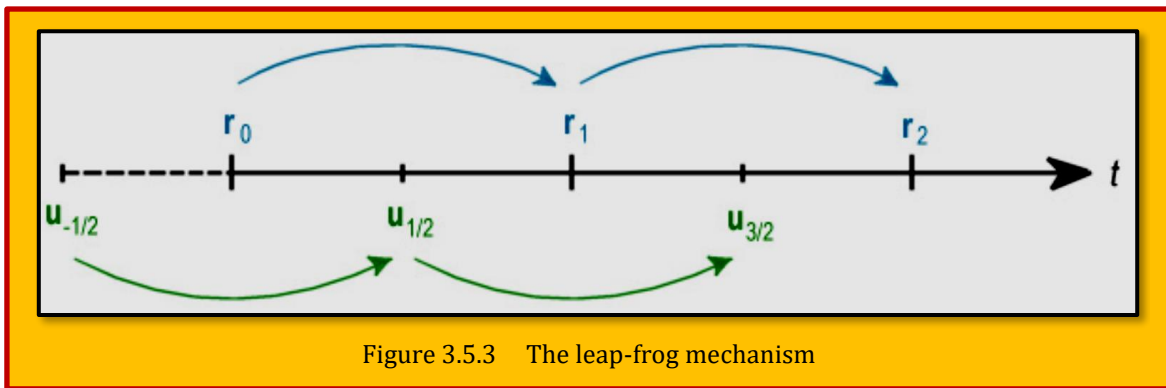


Figure 3.5.3 The leap-frog mechanism

3.5.3 Collision Handling

The small-scale working domain of interactive Lagrangian fluids is limited. A practical way of meeting a convincing environment of the fluid is to constraint the particle system within well-defined boundaries. Boundary containers, such as boxes, spheres, and capsules, are commonly used to

⁶³ Micky Kelager, "Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics", January 9, 2006.

constraint a fluid. When particles collide with a container they must stay inside its boundaries. Likewise, if particles collide with an obstacle, they may not penetrate or gain access to the interior of the object. Collision handling can be divided into two sub parts; **collision detection** and **collision response**. Further discussion is avoided here and interested readers are encourage to read⁶⁴.

3.5.4 Case Study 1 – Comparison of Weakly Compressible and Incompressible SPH

The comparative study for the Weakly Compressible (WCSPH) and Incompressible (ISPH) Smoothed Particle Hydrodynamics methods over an airfoil is investigated by [Shadloo, et. al,]⁶⁵. WCSPH and ISPH simulation results are compared and validated with those of a finite element method (FEM). The quantitative comparisons of WCSPH, ISPH and FEM results in terms of Strouhal number, and velocity gradients on the airfoil boundaries as well as the lift and drag values for the airfoil geometry indicate that the WCSPH method with the suggested implementation produces numerical results as accurate and reliable as those of the ISPH and FEM methods.

3.5.4.1 Formulation of Problem

The SPH method relies on the idea of smoothing field properties over a bounded domain through the devised as in Eq. 3.5.7 which is referred to as the kernel approximation to an arbitrary function $f(\mathbf{r}_i)$. In fact, this arbitrary function can be any hydrodynamic transport property such as temperature, enthalpy, density, viscosity and so forth. Here, $W(\mathbf{r}_{ij}, h)$ is a kernel function, the angle bracket $\langle \rangle$ denotes the kernel approximation $\langle \rangle$, $\mathbf{r}_i$ is the position vector defining the center point of the kernel function, r_{ij} is the magnitude of the distance vector between the particle of interest i and its neighboring particles j , $d^3(\mathbf{r}_j)$ is a differential volume element within the total bounded volume of the domain Ω , and the length h defines the support domain of the particle of interest. The SPH technique in Equation (8.18) assumes that the fields of a given particle are affected only by that of other particles within a cutoff distance of the particle of interest with a smoothing radius kh where k is a coefficient associated with the particular kernel function. A smoothing kernel function is a piecewise spline that should satisfy several conditions: the normalization, the Dirac-delta function, compactness, spherical symmetry, and positive and even function properties. A thorough discussion on the details of these attributes of the kernel function can be found in⁶⁶ and the references therein. In SPH literature, it is possible to find different forms of piecewise smoothing kernel functions possessing the above-listed properties such as Gaussian, cubic or quantic kernel functions. Throughout the present simulations, the compactly supported two dimensional quantic spline kernel.

$$f(\mathbf{r}_i) \approx \langle f(\mathbf{r}_i) \rangle \equiv \int_{\Omega} f(\mathbf{r}_j) W(\mathbf{r}_{ij}, h) d^3\mathbf{r}_j \quad \text{where}$$

$$W(\mathbf{r}_{ij}, h) = \frac{7}{478\pi 7^2} \begin{cases} (3-s_{ij})^5 - 6(2-s_{ij})^5 + 15(1-s_{ij})^5 & \text{if } 0 \leq s_{ij} < 1 \\ (3-s_{ij})^5 - 6(2-s_{ij})^5 & \text{if } 1 \leq s_{ij} < 2 \\ (3-s_{ij})^5 & \text{if } 2 \leq s_{ij} < 3 \\ 0 & \text{if } s_{ij} \geq 3 \end{cases}$$

and $s_{ij} = r_{ij} / h$

⁶⁴ See above.

⁶⁵ Mostafa Safdari Shadloo, Amir Zainali, Mehmet Yildiz, and Afzal Suleman, "A robust weakly compressible SPH method and its comparison with an incompressible SPH", *Int. J. Numer. Meth. Engng*, (2011).

⁶⁶ Liu MB, Liu GR. Smoothed Particle Hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering* 2010.

Eq. 3.5.7

3.5.4.2 Results

Figure 3.5.4 compare the velocity contours of ISPH (upper), FEM (middle) and WCSPH (lower) for the angles of attack of 5 and 15 degrees (contours show the velocity magnitude, m/s) for the $Re = 570$. Similar to the previous benchmark problem, both WCSPH and ISPH results are in good agreement with those of the mesh dependent FEM technique. In all simulations, the results of WCSPH are as accurate as the ISPH ones. The figures further illustrate that the proposed algorithm is also very successful in simulating the flow around the airfoil geometry with different angles of attack across the flow field⁶⁷.

3.5.5 Case Study 2 - Dam Break Water Flow using Lagrangian Description

The analysis of fluid flow is more an area of interest for physicists than computer scientists.

However, in order to be convinced that the Lagrangian fluid method can produce realistic fluid motion we will examine the fluid flow. We will study the velocity flows produced by the dam-break problem for the water. In a classic dam-break problem the fluid is constrained inside a dam, and when

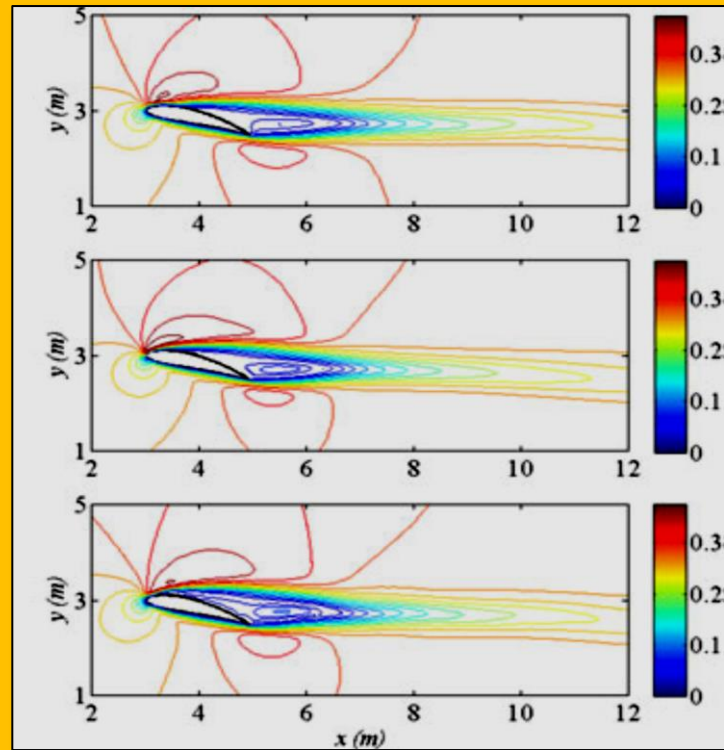


Figure 3.5.4 Comparison of ISPH (upper), FEM (middle) and WCSPH (lower) velocity contours for the angle of attack of 15 degrees at $Re = 570$ (Courtesy of Shadloo¹⁰⁵)

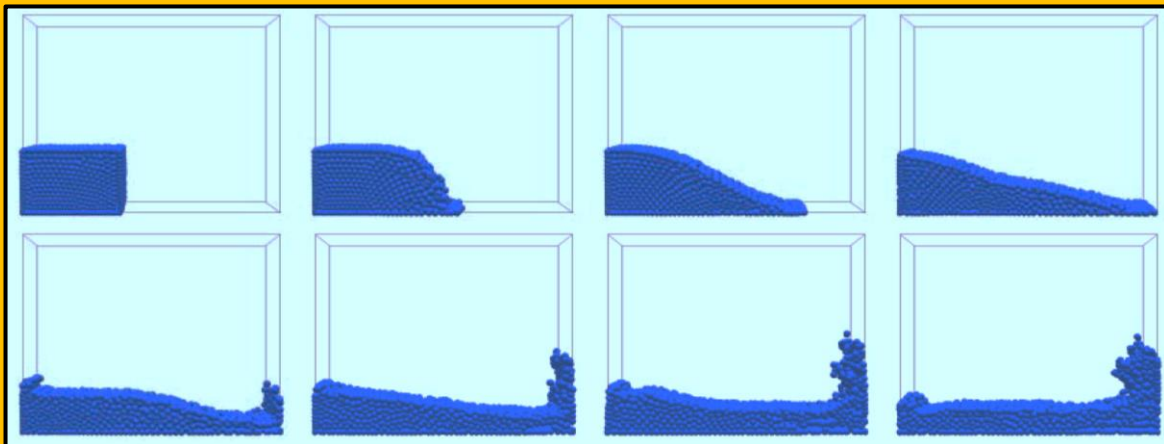


Figure 3.5.5 Dam-Break Flow of water

⁶⁷ Mostafa Safdari Shadloo, Amir Zainali, Mehmet Yildiz, and Afzal Suleman, "A robust weakly compressible SPH method and its comparison with an incompressible SPH", *Int. J. Numer. Meth. Engng.*, (2011).

the fluid is at rest the dam is broken, or the barricade that constrains the fluid is removed. The fluid now flows freely and often collides with a vertical wall. Frames from the dam-break of water simulated by particles are depicted on [Figure 3.5.5](#) where the flow of water simulation time interval is 0.1s between each frame, from left to right, top to bottom. This is just a survey of how the visible water particles flow in the dam-break problem. Frames from the dam-break of water simulated by 2250 particles.

3.5.6 Case Study 3 - Dam Break using MLPG-RBF and Shallow Water Equations

The application of the meshless local Petrov-Galerkin (MLPG) method to solve the shallow water equations (SWE) is investigated by [Mužík1 and Holíčková]⁶⁸. The shallow water equations (which also called the de Saint-Venant equations) are used to describe flow behaviors in bodies of water where the horizontal length scales are much greater than the flow depth, therefore, the 3D problem can be assumed as 2D. This localized approach is based on the meshless weak formulation with the use of radial-basis functions (RBF) as the trial functions. In this work, the numerical model is applied to simulate a dam-break problem as one of most descriptive benchmark problems for SWE. As a result, the adopted meshless method not only shows its algorithm applicability for class of problems described by SWE, but also brings more efficiency than several conventional mesh-based methods. The problem models a partial dam-break for a rapid opening of a sluice gate with a non-symmetric breach and its ability to simulate discontinuous flows. The computational domain is a 200 m by 200 m region. A dam is located in the middle of the domain with 10 m thickness. The initial water depth is 10 m on one side and 5 m on the other side of the dividing wall. At time $t = 0$, the dam fails, and the water is released through the 75 m wide non-symmetric breach, as shown in [Figure 3.5.6 - \(left\)](#). When the downstream water depth is 5 m, the flow is subcritical everywhere. The boundary conditions at $x=0$ and $x=200$ m are assumed to be transmissivity and all other boundaries are considered as reflective. At the instant of the dam break, water is released through the breach, forming a positive wave propagating downstream and a negative wave spreading upstream. We compare our results by at $t = 7.2$ s ([Figure 3.5.6 - right](#)), when the waves have not yet reached all the boundaries, with least-squares finite-element method (LSFEM). The left moving positive wave and right moving negative wave are both well resolved. The results were confirmed more stable to

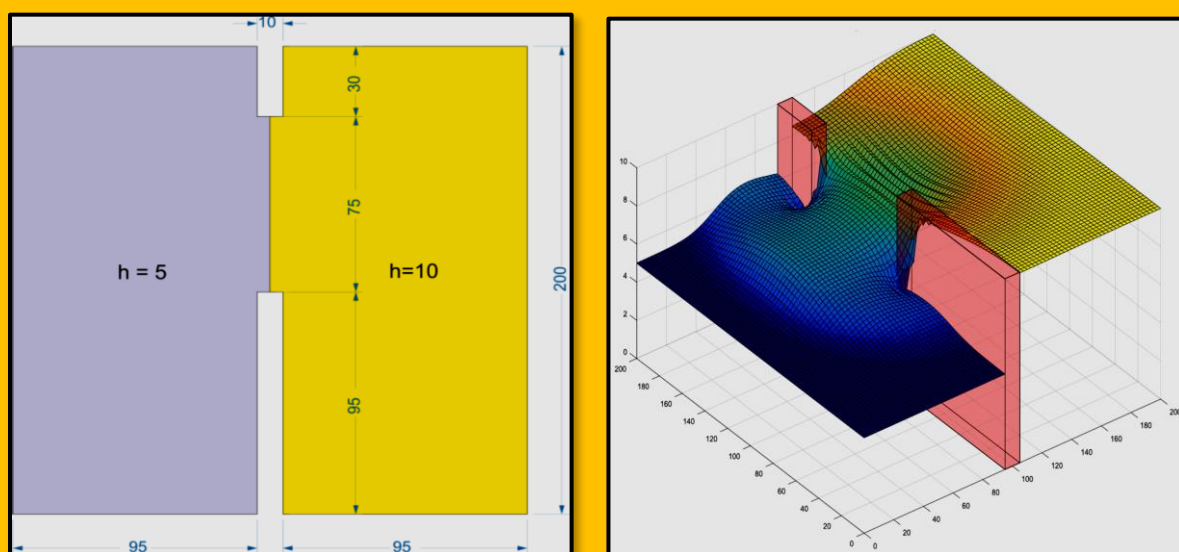


Figure 3.5.6 Geometry and Water surface profile of the 2D dam-break problem at $t = 7.2$ s.

⁶⁸ Juraj Mužík1, Martina Holíčková, "Meshless simulation of dam break using MLPG-RBF and shallow water equations", MATEC Web of Conferences 117, 00127 (2017).

capture the fine details of the flow. The behavior of the numerical scheme is in satisfactory agreement with computed results of these researches.

3.5.7 Case Study 4 - SPH Method for Evaporating Multiphase Flows

Because evaporation is encountered in many engineering applications, such as fuel droplets in engines, liquid sprays, and material processing, a numerical method to accurately predict liquid evaporation is of great importance. Common engineering models for predicting droplet evaporation assume that the liquid droplet is a point source with homogeneous properties. The primary concern of these models is the mass transfer rate without consideration of the gradient in the droplet or the liquid-gas interface. While such models are useful in engineering applications, advanced numerical methods are needed to reveal the details of the evaporation process. The dynamics of evaporating flows involves phase change and energy transfer at the liquid-gas interface, diffusion of vapor species in the gas phase, and multiphase flows with sharp interfaces. Because of the complexity of the evaporation problem, it is challenging to make a detailed numerical simulation. The main numerical challenges in simulating evaporating flows include the treatment of phase change and the sharp discontinuity of fluid properties at the liquid-gas interface. Phase change due to evaporation causes mass transfer from one phase to another phase. The discontinuity at the liquid-gas interface, of variables such as density ratio, also leads to numerical difficulties.

The intent of [Xiufeng Yang & Song-Charng Kong]⁶⁹ this work is to provide a numerical method, based on smoothed particle hydrodynamics (SPH), to simulate multiphase flows with evaporation. The SPH method is a Lagrangian mesh-free particle method. In SPH, a continuous fluid is discretized using SPH particles, which carry physical properties, such as mass, density, pressure, viscosity, and velocity. Since SPH is a mesh-free method, a smoothing kernel is introduced to connect the neighboring particles. The variables and their spatial derivatives are discretized in summations over particles. In the SPH method developed for this study, the SPH particles near the interface are allowed to change their mass to model the process of evaporation at the interface. The rate of mass change of SPH particles due to evaporation depends on the vapor mass fraction in the gas phase and the saturated vapor mass fraction at the interface. The saturated vapor mass fraction can be predicted by the (Clausius-Clapeyron) correlation. During the process of evaporation, the mass of a liquid SPH particle at the interface increases, while the mass of a gas SPH particle decreases. To constrain the mass of individual SPH particles, a particle will split into smaller particles if its mass is large enough or merge into a neighbor particle if its mass is small enough.

3.5.7.1 Basic Formulations of the SPH Method

In SPH, the value of a function $f(r)$ at point r_a can be approximated using the following integration:

$$f(r_a) \approx \int f(r) W(r_a - r, h) dV \quad \text{Eq. 3.5.8}$$

where W is a kernel function and dV is a differential volume element. The parameter h is referred to as a smoothing length, which determines the size of the integral domain. In this paper, the following hyperbolic-shaped kernel function in two-dimensional space is used:

$$W(s, h) = \frac{1}{3\pi h^2} \begin{cases} s^2 - 6s + 6 & 0 \leq s < 1 \\ (2-s)^3 & 1 \leq s < 2 \\ 0 & 2 \leq s \end{cases} \quad \text{Eq. 3.5.9}$$

⁶⁹ Xiufeng Yang And Song-Charng Kong, "Smoothed Particle Hydrodynamics Method for Evaporating Multiphase Flows", Physical Review E 96, 033309 (2017).

In the SPH method, a continuous fluid is discretized into properties, such as mass m , density ρ , velocity u , and viscosity μ . Then the integration of Eq. (5.17) is discretized in particle summation as follows

$$f(\mathbf{r}_a) \approx \sum_b f(\mathbf{r}_b) W(\mathbf{r}_a - \mathbf{r}_b, h) \frac{m_b}{\rho_b}$$

Eq. 3.5.10

3.5.7.2 Evaporation of a Static Drop

The evaporation of a static drop was simulated using the proposed SPH method. The initial radius of the drop is $R_0 = 0.15$ mm. The initial temperature of the drop is 353 K. The drop was located at the center of a square computational domain, which was filled with gas. The length of the square was 1.2 mm. The initial temperature of the gas was 373 K. The temperature of the boundary was also 373 K, and did not change during the simulation. These temperatures were chosen in order to be consistent with and to allow comparisons with the conditions in the literature. The initial vapor mass fraction in the gas phase was zero. The vapor mass fraction of the boundary remained zero. The initial particle spacing was 0.02 mm. **Figure 3.5.7** shows that the size of the drop decreased slightly. The decrease in the drop size, as compared with the result from a two-dimensional (2D) axisymmetric level-set method. It should be noted that the 2D circle used in this study corresponded to the cross section of a three-dimensional (3D) cylinder of infinite length, while the 2D axisymmetric circle used corresponded to a 3D sphere.

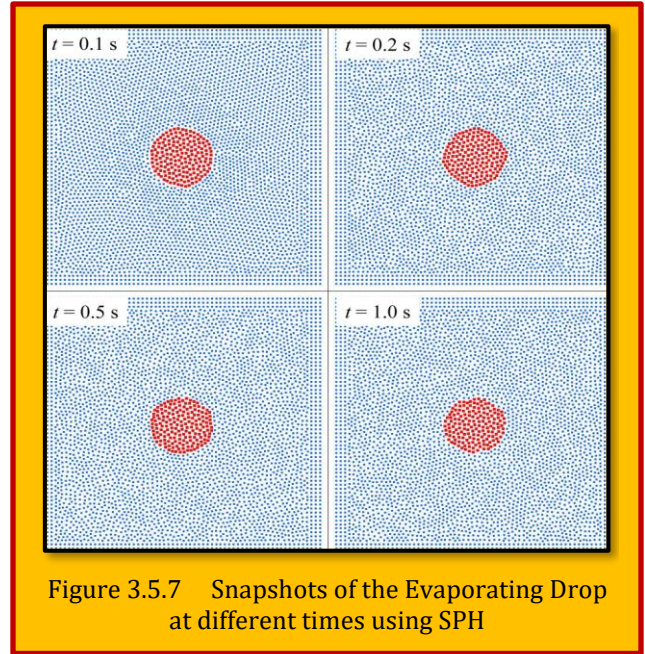


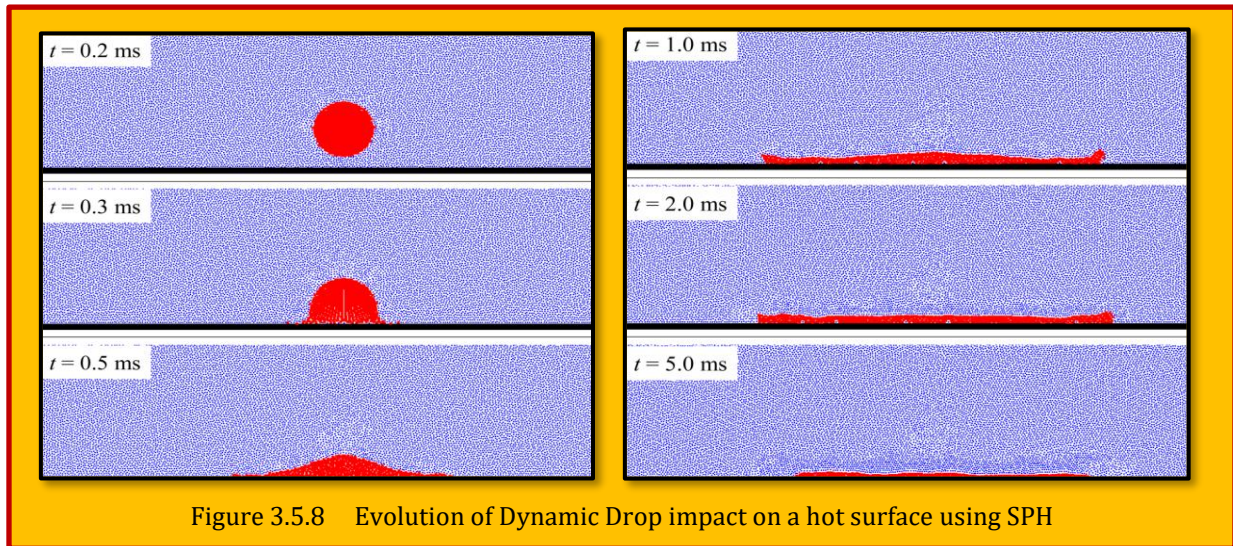
Figure 3.5.7 Snapshots of the Evaporating Drop at different times using SPH

3.5.7.3 Evaporation of a Dynamic Drop Impacting on a Hot Surface

The proposed method was also used to simulate the evaporation of a drop impacting a hot surface. The initial radius of the drop was $R = 0.25$ mm and the initial velocity of the drop was $U = 2$ m/s. The height and length of the computational domain were 1.5 and 5.0 mm, respectively. The drop was located at the center of the domain and was surrounded by gas. The initial temperature of the drop was 353 K. The initial temperature of the gas was 373 K. The temperature of the boundaries was also 373 K, and did not change during the simulation. The initial vapor mass fraction in the gas phase was zero. The vapor mass fraction of the boundary remained zero. The initial particle spacing was 0.02 mm. **Figure 3.5.8** shows the evolution of drop impact on a hot surface. After the drop touched the surface, it spread and formed a film on the surface. At approximately 1.0 m/s, a tiny crown like structure was formed around the rim. Later, the crown merged with the film, and the film receded. Finally, the film reached an equilibrium size.

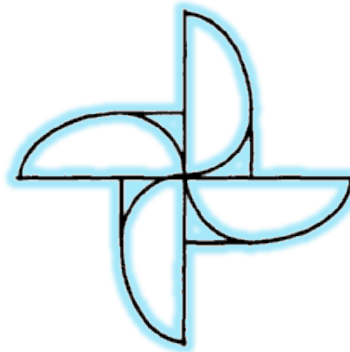
Since the initial temperature of the drop was lower than the gas temperature, the heat transfer from the surrounding gas to the drop led to the decrease in the local gas temperature. However, the drop temperature also decreased slightly because evaporation consumed energy, as discussed earlier. When the drop spreads on the hot surface and forms a film, heat transfer from the hot surface to the film increased the temperature of the film. The intent of this paper was to present an SPH method to simulate evaporating multiphase flows. This method accurately models the process of evaporation at the liquid-gas interface and the diffusion of the vapor species in the gas phase. An evaporating mass rate was derived to calculate the mass transfer at the interface. To model the process of phase change from the liquid phase to the gas phase, mass was allowed to transfer from a liquid SPH particle

to a gas SPH particle. Thus this proposed method, unlike the traditional SPH method, allows change in the mass of an SPH particle. Additionally, particle splitting and merging techniques were developed to avoid the large difference in the SPH particle mass.



3.5.7.4 Concluding Remarks

In general, the results show that the method proposed in this paper successfully replicated the physical process of evaporating flows, such as heat and mass transfers and the diffusion of the vapor species. The example was to simulate the evaporation of a static drop—because of evaporation, the present SPH method predicts the decreases of both the temperature of the interface and the size of the drop. The last example was to simulate the evaporation of a drop impacting a hot surface. The temperature of the liquid-gas interface decreased at first because of evaporation, especially at the rim of the film. Then the temperature increased because of the heat transfer from the hot surface to the liquid. In summary, the results of this study indicate that the numerical method proposed in this paper can be successfully used to produce an evaporating flow simulation. Additional information can be attained from [Yang & Kong]⁷⁰.



⁷⁰ Xiufeng Yang And Song-Charng Kong, “Smoothed Particle Hydrodynamics Method for Evaporating Multiphase Flows”, Physical Review E 96, 033309 (2017).

4 CFD Applications in Other Areas

Recently CFD finds very wide application in different areas of science and engineering; some examples are⁷¹:

- Aerodynamics of Aircraft and Vehicles - Lift and Drag
- Hydrodynamics of Ships
- Power plant - Combustion in Internal Combustion Engines (ICE) and Gas Turbines
- Turbo machinery - Flows inside rotating passages, Diffusers etc.
- Electrical and Electronics Engineering - Cooling of Equipment Including Microcircuits.
- Chemical Process Engineering - mixing and separation and polymer molding.
- Marine Engineering - loads on off-shore structure.
- Environmental Engineering - Distribution of Pollutant and Effluents.
- Hydrology and Oceanography - flows in rivers, estuaries and oceans.
- Meteorology - Weather Prediction.
- Biomedical Engineering - blood flows through arteries and veins.
- Food Processing
- External and internal environment of buildings: wind loading, ventilation analysis and heating/cooling load calculations.

4.1 Food Processing

CFD applications in food industry may assist in a better understanding of the complex physical mechanisms. [Schott]⁷², [Quarini]⁷³ have reviewed the general application of CFD to the food processing industry. Moreover, other literatures are also available on specific CFD application areas such as: Clean-room design, Refrigerated transport, Static mixers, and Pipe flow. Since CFD technique can be of great benefit to the food processing industry, fast development has taken place in the past few years. CFD, as a tool of research for enhancing the design process and understanding of the basic physical nature of fluid dynamics can provide benefits to the food processing industry in many areas, such as **Drying, Sterilization, Mixing, Refrigeration, Crystallization, Pasteurization** and other application areas⁷⁴.

4.1.1 Drying

Drying is a common food manufacturing process. The drying rate is a strong function of air flow or air velocity. Therefore, it is of great importance to know the air flow and velocity in the drying chamber, thus leading to know the areas of adequate air velocities for proper drying. However, air flow and air velocity are difficult to measure during operation because several sensors are needed to be placed at various directions of air flow and locations. Since there are some difficulties in modelling the complex phenomena, especially the gas turbulence, CFD is a powerful tool to aid the prediction of drying process. CFD has been used to predict the air flow and velocity during drying. Drying tests of several fruits were performed and the result showed that the degree of fruit dryness depended on its position within the drier. Determination of pressure profiles and air velocities by CFD showed that the main cause of the variations in drying rates and moisture contents was the lack of spatial

⁷¹ Versteeg, H., "An Introduction to Computational Fluid Dynamics", Pearson Publications. ISBN 978-81-317-2048-6, (2009).

⁷² Scott GM (1977), "Simulation of the flow of non-Newtonian foods using computational fluid dynamics", Campden & Chorleywood Food Research Association R & D Report No. 34, UK.

⁷³ Quarini J (1995), "Applications of Computational fluid dynamics in food and beverage production". Food Sci Technol Today 9: 234-237.

⁷⁴ Bin Xia, Da-Wen Sun, "Applications of computational fluid dynamics (CFD) in the food industry: a review", Computers and Electronics in Agriculture 34 (2002) 5-24.

homogeneity of air velocities within the drier. With the aid of CFD, researchers studied velocity fields in a modern sausage drier in order to provide information on air circulation inside the drier, which showed that CFD was able to predict the effects of filling level on air-flow patterns and also to identify measurement errors in areas where the main air flow direction was horizontal⁷⁵.

However, the quantitative comparison between the simulated and measured air velocities showed wide discrepancy with means of absolute differences of about 0.6 m/s. Although, the flow pattern and air velocity in the drier can be predicted using CFD modelling, further study on how to control the drying process and to reduce the energy cost is still a research topic for CFD modelling. Meanwhile, more attention should be paid on the assumptions such as spatial homogeneity because of such assumptions could lead to inaccuracy in prediction. CFD has also been used to investigate the performance and the design of spray dryers in the food industry. Spray dryers are used to produce products such as milk and coffee powder, as well as detergents. However, the design of spray dryers for the food industry is difficult because the performance of spray dryers is heavily influenced by the complexity of air and spray flow patterns inside the dryers.

Therefore, there is considerable scope for the application of CFD simulation including optimum design of spray dryers and solutions for operational problems, such as wall deposition. In the past several years, researches, such as modelling and measuring the air flow pattern in a co-current pilot plant spray dryer (Kieviet et al., 1997) and analyzing the effects of air inlet geometry and spray cone angle on the wall deposition rate in spray dryers have been performed. All these studies show that there appears to be a large scope for using CFD for other purposes. For example, CFD can be used to simulate the air flow in a spray dryer in two dimensions and calculate the trajectories and the course of the drying process of the atomized particles. Straatsma⁷⁶ developed a drying model utilizing turbulence model to calculate the gas flow field and showed that the drying model was an effective tool in giving indications of how to adapt the modelling in industrial dryers to obtain a better product quality or to optimize the drying performance of the unit. However, as the applications and specifications of dryers become more and more complex, so does the need for improved test work in pilot plants, and CFD simulations become more important in providing quick and valuable information.

4.1.2 Sterilization

It is known that consumer demands for food products focus on safety, product quality and cost⁷⁷. Therefore, it is of great necessity to enhance quality and assure safety of the food supply. Sterilization is an important technique for food storage and preservation. CFD can be used to study both temperature distribution and flow pattern of food in the process of sterilization so as to optimize the quality of food products. Thermal processing remains the most significant technique of sterilization which results in microbial inactivation, but in the meantime, quality loss and flavor development. Excessive heating will affect food quality and its nutritive properties. With the application of CFD, there has been a number of studies to optimize the thermal sterilization of foods. These studies had led to substantial improvement on the optimal control of the process and the retention of the nutritional and sensory quality of the food. Another researches carried out a series of research work in canned food sterilization with CFD simulation. The work varied from those simulating the changes of bacteria diffusion and their transient spatial distribution during sterilization process to those simulating natural convection heating within a can of liquid food during sterilization. It is only in recent years that the food pouches have been introduced to the market and, therefore, little or no

⁷⁵ See above.

⁷⁶ Straatsma, J., Houwelingen, V.G., Steenberg, A.E., Jong, P.D., "Spray drying of food products: Simulation model", *Journal of Food Engineering* 42 (2), 67–72.

⁷⁷ Bin Xia, Da-Wen Sun, "Applications of computational fluid dynamics (CFD) in the food industry: a review", *Computers and Electronics in Agriculture* 34 (2002) 5–24.

study has been executed on sterilization of food in pouches. CFD code was used for the purpose to simulate the transient temperature, velocity profiles and the shape of the slowest heating zone in sterilization of carrot soup in pouches. The modelling of a continuous sterilization process to optimize the quality of safe food has also been developed and the results showed that CFD modelling could be of significant help to the liquid food sterilization.

However, all of these investigations about CFD application in sterilization are on the thermal sterilization in the limited area of liquid foods. There are still remains many challenges in the area of sterilization with the application of CFD. For instance, Ultra-violet, visible and infra-red light surface sterilization, plasma/corona sterilization, electrons and X rays sterilization, nascent oxygen/ozone sterilization of fruits and vegetables, pressure sterilization of fresh fruit juices and cooked ham. The application of CFD in these sterilization fields of food is still to be developed in the future. Moreover, assumptions are normally made to simplify CFD modelling. For example, specific heat, thermal conductivity and volume expansion coefficient were assumed to be constants in the study by Abdul Ghania et al. (1999a) although, all the parameters are temperature dependent. More studies should be carried out to minimize these assumptions and thus to improve the accuracy of CFD prediction.

Another area for the application of CFD is the real time control of the sterilization. Effective real-time monitoring of sterilization will improve the quality and safety of foods. Above all, the ultimate objective is to optimize the sterilization process of the food and to obtain food with excellent quality and safety. With the aid of CFD application, the sterilization process can be improved.

4.1.3 Mixing

In the food processing industry, mixing is one of the most common operations. Mixing applications involve the substances of gas, liquid and solid. And the mixing of fluids is one of the most important unit operations for the food processing industry. However, mixing is a complicated process as regards to the multiphase turbulence during mixing and the design of a mixer. CFD is a powerful tool for the modelling of mixing processes. It provides a natural method to link food process and fluid flow information. During mixing, a common method of enhancing the process is to use some kind of stirrer or paddle. CFD codes have been applied in optimizing the mixing process to minimize energy input and to shorten the processing time. Therefore, research has been carried out on the distribution of energy in mixing vessel and on the effects of mixing quality when the stirrer is in different position. Such prediction of the mixing process within these units was impossible in the past. Recently, CFD modelling of mixing in stirred tanks has been carried out by [Sahu]⁷⁸, with several important points about impeller-vessel geometry, energy balance and the linkage between the flow field and the design objective being addressed. Although no experiments were carried out in the study, the predicted values of mixing time were compared with published experimental data and the agreement was within 5–10%. This study will benefit the design of the stirred tanks, and some technical problems about the impeller types, mixing time and equipment size can be avoided.

The design of mixing devices is an important topic in analyzing the mixing process. Therefore, some research work focusing on the application of CFD on the design of mixing devices, for instance, shallow bubble columns, has been investigated. The results of these studies will provide benefits including easy measurement of the drop size distribution, the velocities of the phases and the degree of mixing, and accurate description of the turbulence, swirling and vortices generated in the mixer. Thus, all the development of CFD application on the mixing in the food processing industry will lead to more accurate monitoring, control and optimizing of mixing process. In the meantime, it will form a good basis for mixing process improvement.

⁷⁸ Sahu, A.K., Kumar, P., Patwardhan, A.W., Joshi, J.B., "CFD modelling and mixing in stirred tanks", Chemical Engineering Science 54 (13–14), 2285–2293, 1999.

4.1.4 Refrigeration

The consumption of frozen foods has increased continually in the past years because frozen foods have demonstrated good food quality and safety record. Refrigeration can slow down bacterial growth and preserve food. Therefore, researchers have recently applied CFD in the modelling of heat and mass transfer in foods during refrigeration (chilling and freezing). They have developed the modelling of air blast and vacuum cooling, chilling, cold chain, cold store, refrigerated room and refrigerated display cabinets. CFD simulation of heat and moisture transfer for predicting cooling rate and weight loss of cooked ham during air blast chilling process has been investigated. Both experimental and predicted results showed that the core temperature of the cooked ham was cooled down from 74.4 to 4°C within approximate 530 min. The experimental accumulative weight loss was 4.25%, while the simulated results were 4.07 and 4.22%, respectively, obtained from standard k - ϵ model and LRN k - ϵ model. At the same time the effect of fluctuation in inlet airflow temperature was studied, indicating that setting the boundary condition of airflow temperature is an important factor affecting the predicting accuracy. If a constant temperature was assumed for the inlet air, the weight loss (4.37%) was over predicted. Furthermore, the effects of different k - ϵ models and thermocouple positions on the prediction accuracy of CFD modelling of air-blast chilling process were also analyzed. Some developed a two-dimensional simulation model for the airflow in two industrial meat chillers. Recently, it was investigated the temperature increase in frozen food packaged in pallets in the distribution chain by means of CFD modelling. Good agreement was found between the experimental and modelling results with the differences normally within 10%. The study showed that the controlled temperature throughout the cold chain was necessary to ensure a high food quality with long storage duration. Although the modelling of air flow and temperature distribution has been well developed, models for phase transition, such as condensation and evaporation are not yet available.

4.1.5 Crystallization

It is one of the oldest unit operation in the chemical and food industry but the design and operation of crystallization processes still pose many problems. However, until recently, there have been few tools capable of providing the required capabilities. This is because modelling of crystallization processes poses a number of challenges. The key challenge is representing the inherent physical and chemical complexity of crystallization phenomena mathematically and validating the resulting mathematical model against experimental data. CFD helps in modelling of crystallization process and design of crystallizer⁷⁹.

4.1.6 Pasteurization

Pasteurization is a vital unit operation which is used to inactivate the spoilage organisms and enzymes present in the milk. Similarly, CFD analysis for thermal pasteurization of intact eggs. Calculated temperature profiles were found to be in good agreement with experimentally observed data for eggs of different sizes. A generally accepted kinetic inactivation model for *Salmonella enteritis's* was incorporated in the CFD analysis and provided a basis for process assessment. Minimum process times and temperatures to provide equivalent pasteurization effectiveness at 5 log reductions of the target microorganism were obtained on a theoretical basis. Combining a CFD analysis with inactivation kinetics proved to be a very useful approach for establishing process conditions leading to consumer safe eggs. Also, conducted in-package pasteurization for beer microbiological stabilization. A heating process was simulated at 60°C up to 15 PUs (a conventional beer process, in which 1 Pasteurization Unit (PU) is equivalent to 1minute at 60°C). The temperature profile and convection current velocity along the process and the variation of the PUs were evaluated in relation to time considering the cans in the conventional, inverted, and horizontal positions. The package position did not result in process improvement.

⁷⁹ Kaushal and Sharma, "Concept of Computational Fluid Dynamics (CFD) and its Applications in Food Processing Equipment Design", J Food Process Technical 2012, 3:1.

4.2 CFD in Semiconductor Industry

As with most of the technologies, one can know its applications only when it is invented⁸⁰. Today CFD is being used to help in designing in every area where Fluid is involved. CFD has found its application with semiconductor industry as well. CFD solution can help immensely in reducing the number of experiments required to design various chip manufacturing equipment's. After validation with experiments, one can find finer details more easily from CFD than with experiment e.g. temperature distribution over the surface, deposition rate, rate of desorption. Various semiconductor industries have started using CFD calculation to help their design engineers. But it still has a long way to go and gain confidence from everybody to its results. CFD (Computational Fluid Dynamics) could be used to model the thermal system at a board level as well as within a semiconductor chip, so that efficient heat-dissipation mechanisms and sufficient cooling systems could be designed around these systems. CFD could hold interesting possibilities given that we are now looking at three-dimensional (3D) transistor dies as well as multi-die two-dimensional (2D) packages. Heat dissipation is critical for the long-term reliability of semiconductor devices

4.2.1 Brief Description of Semiconductor Devices

Semiconductor devices are electronic components that exploit the electronic properties of semiconductor materials, principally silicon, germanium, and gallium arsenide, as well as organic semiconductors. Semiconductor devices have replaced thermionic devices (vacuum tubes) in most applications. They use electronic conduction in the solid state as opposed to the gaseous state or thermionic emission in a high vacuum. In layman terms, semiconductor is the category of conductors which besides being a conductor of current is also an insulator. As evident from the diagram (see [Figure 4.2.1](#)), the energy band gap between valence band and conduction band is:

- large in case of Insulator
- overlap in case of Metal
- moderate in case of Semiconductor

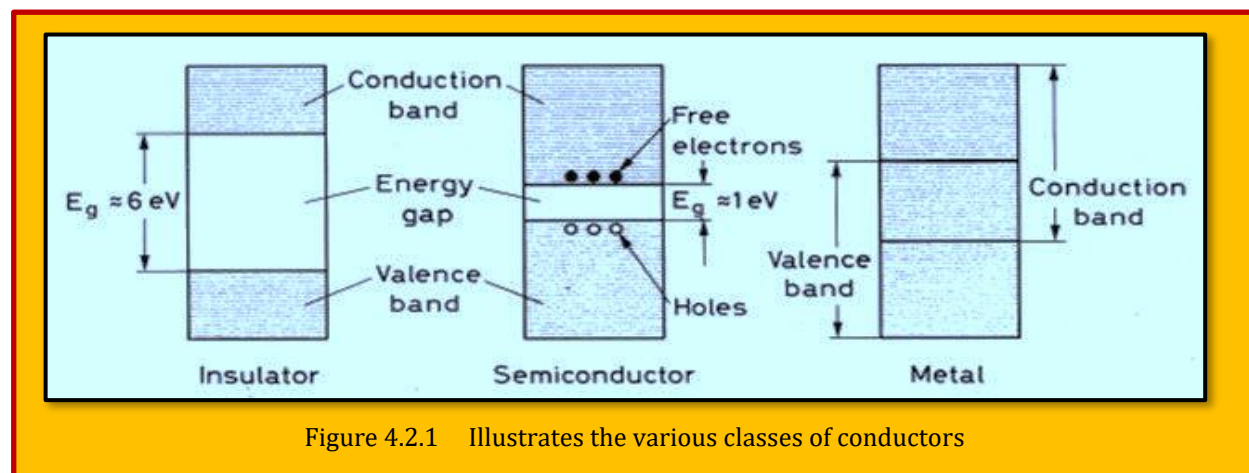


Figure 4.2.1 Illustrates the various classes of conductors

What are they you ask?

1. Valence Band - Band in which electrons reside.
2. Conduction Band - Band to which electrons jump and conduct electricity.

⁸⁰ CFD online.

3. Energy Band - Band which does nothing!

Similarly for an electron to jump to conduction band, it requires energy. Such an amount of energy is almost impossible to provide to insulators but easily achievable in case of metals. In case of semiconductors energy can be tuned so as to make it work like metal or an insulator. The energy provided must be greater than the energy of band gap ($>1\text{eV}$). This feature enables semiconductors to be used as *Switch*. For switching ON the circuit you just need to provide energy greater than 1eV to the semiconductor device. **Figure 4.2.2** demonstrates a semiconductor electronics.



Figure 4.2.2 Modern Semiconductor

4.2.2 Thermal Management in Semiconductors

There are at least ten good reasons to include thermal measurements as a routine step in any electronic component or system design process⁸¹. Amid all the promotion of solid-state superlatives ranging from data rate to feature size to LED light output, one characteristic is never touted: *Junction Temperature*. That's because Junction Temperature (JT) is an undesired but unavoidable side-effect of high currents and/or switching speeds. A p-n junction, whether it is one of millions on a CPU chip or the only one within a power LED, generates heat. In the past two decades the industry has seen heat dissipation increase by orders of magnitude. Faster is better, but faster is also hotter. This trend is not without consequences. A 10° increase in JT can cause a 50% reduction in a semiconductor device's life expectancy. In LEDs, both brightness and color can suffer as JT increases. And of course the twin issues of safety and cooling can impact the design of an entire system, not just the semiconductor device producing the heat. All these facts point toward the need for a thorough grasp of thermal behaviors at the chip level, and beyond. True understanding comes with physical measurements performed on actual devices. This is especially true in the world of semiconductors. You see heat dissipation in semiconductor packages is one of the limiting factors in miniaturization. One of the biggest concerns of circuit designers is reducing power that is continuously increasing due to bandwidths. As a result, the chip temperature increases. This change first modifies and then later destroys the operation of the circuit if the heat is not correctly led out of the device. Being able to understand the true thermal characteristics of a chip that will go inside an enclosure which is jam packed with other heat generating equipment can be very helpful. While most manufacturers publish thermal metrics for their chips, unfortunately not every manufacturer knows how to conduct an appropriate thermal characterization of their devices. So you can't always rely on published metrics⁸². (see **Figure 4.2.3**).

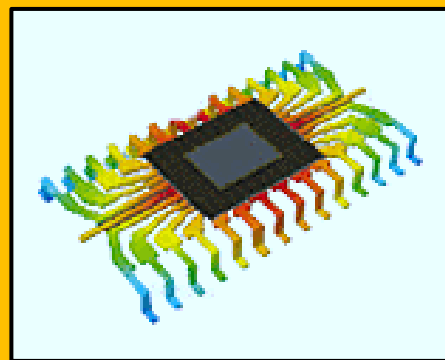


Figure 4.2.3 Thermal Management of Semiconductor (courtesy of Mentor CFD)

4.2.3 Can You Really Fry an Egg on a CPU?

⁸¹ From Mentor CFD Blogs.

⁸² From Mentor CFD Blogs.

An interesting questions arises whether you can really fry an egg in CPUs. Believe or not, somebody already try that⁸³. Solving complex thermal models with computational fluid dynamics (CFD) requires a lot of processing power, and a central processing unit (CPU) under full load generates a fair amount of heat. But can you cook an egg on it? This article describes the model, simulations, and the ultimate conclusion. Solving complex thermal models with CFD requires a lot of processing power and a CPU under full load generates a fair amount of heat. But can you cook an egg on it? Before you throw away your conventional heatsink and fan in favor of a multifunctional omelet, we'll investigate what CFD to predict about the fate of your PC if you do so. (see [Figure 4.2.4](#)). Unfortunately, the CPU junction temperature exceeds 90°C within 6 seconds, at which point the CPU clock would throttle down to reduce the thermal power and prevent damage to the system; less than ideal for a cooling solution. The egg would also burn and catch fire. The central location of the CPU on the board and the large obstacles to air flow in the neighboring memory DIMMS and I/O ports mean limited cold air can passively flow over the hot egg by natural convection. The passive cooling of the egg cannot match the forced convection of the stock cooler. An egg-based cooling solution would only keep the CPU below the maximum 90°C if the CPU performance were throttled down so there are only possible applications in lower power environments with plenty of ventilation. With the requirement of frequently swapping out the egg, it can't this catching on. If the aim is to cook eggs though, CPUs certainly produce enough heat to do so; with thermal throttling, the processor acts as a thermostatically controlled surface at around 90°C, sufficient to cook on. If you value your computer, maybe consider buy a frying pan instead.

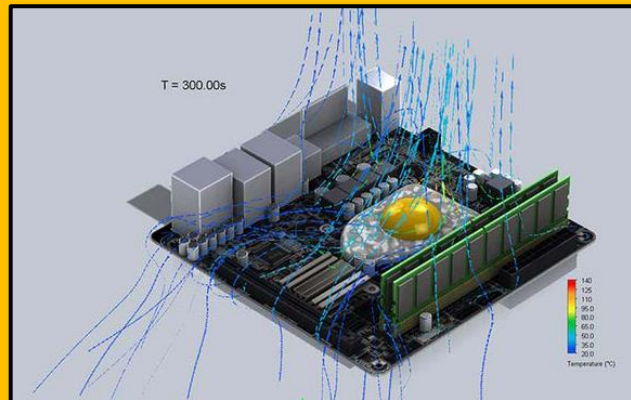
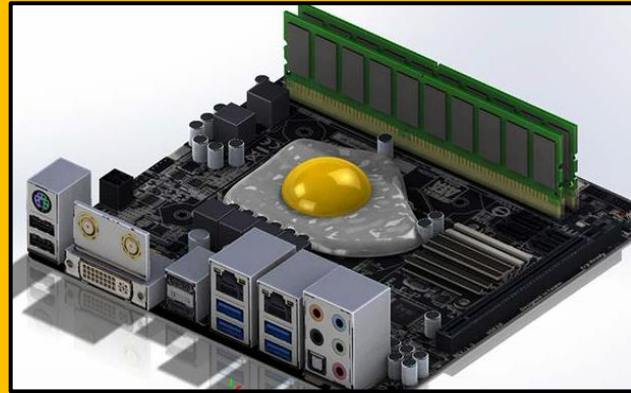


Figure 4.2.4 An Example of an Egg Frying on a CPU

4.3 Magneto-Hydro-Dynamics (MHD)

Magneto-Hydro-Dynamics (MHD), also magneto-fluid dynamics or hydro-magnetics, is the study of the magnetic properties of electrically conducting fluids. Examples of such magneto-fluids include plasmas, liquid metals, salt water and electrolytes. The word "magneto-hydro-dynamics" is derived from *magneto* meaning magnetic field, *hydro*- meaning water, and *dynamics* meaning movement⁸⁴. In a nutshell, MHD is the study of electrically conducting fluids, combining both principles of fluid dynamics and electromagnetism. According to [Battista]⁸⁵, the subject of MHD is traditionally studied as a continuum theory, that is to say, attempts at studying discrete particles in the flows are not at a level such that computation in these regards is realistic. To run "realistic simulations" would require

⁸³ James Forsyth, System-Level Design, Semiconductor engineering.

⁸⁴ From Wikipedia, the free encyclopedia.

⁸⁵ Nicholas A. Battista, "An Introduction to Magnetohydrodynamics", Stony Brook University, December, 2010.

computations of flows with many more particles than current computers are able to handle. Thus, the only way to study MHD seems to be in its continuum form- leading us to its description using the Navier-Stokes fluids equations⁸⁶.

4.3.1 MHD Equations

The ideal MHD equations consist of the continuity equation, the Cauchy momentum equation, Ampere's Law neglecting displacement current, and a temperature evolution equation. As with any fluid description to a kinetic system, a closure approximation must be applied to highest moment of the particle distribution equation. This is often accomplished with approximations to the heat flux through a condition of adiabaticity or isothermally. The main quantities which characterize the electrically conducting fluid are the bulk plasma velocity field $\mathbf{v}$, the current density $\mathbf{J}$, the mass density ρ , and the plasma pressure p . The flowing electric charge in the plasma is the source of a magnetic field $\mathbf{B}$ and electric field $\mathbf{E}$. All quantities generally vary with time t as described by Error! Reference source not found..

- I. The two continuity equations for charge conservation where $\rho_c = 0$ because we are assuming the absence of an external charge distribution.
- II. The Cauchy momentum equation where the Lorentz force term $\mathbf{J} \times \mathbf{B}$ can be expanded using Ampere's law and the vector calculus identity where the first term on the right hand side is the magnetic tension force and the second term is the magnetic pressure force.
- III. The ideal Ohm's law for a plasma.
- IV. Faraday's law.
- V. The low-frequency Ampere's law neglects displacement current.
- VI. The magnetic divergence constraint.
- VII. Energy equation where $\gamma = 5/3$ is the ratio of specific heats for an adiabatic equation of state. This energy equation is, of course, only applicable in the absence of shocks or heat conduction as it assumes that the entropy of a fluid element does not change.
- VIII. Hartmann number (Ha) is the ratio of electromagnetic force to the viscous force first introduced by Hartmann where B is the magnetic field, L is the characteristic length scale, σ is the electrical conductivity, μ_0 is the dynamic viscosity.

$$(I) \quad \frac{\partial \rho_c}{\partial t} = \nabla \cdot \mathbf{J} = 0$$

$$(II) \quad \rho \left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \right) \mathbf{v} = \underbrace{\mathbf{J} \times \mathbf{B}}_{\text{Lorentz Force}} - \nabla p \quad \text{where} \quad \mathbf{J} \times \mathbf{B} = \frac{(\mathbf{B} \cdot \nabla) \mathbf{B}}{\mu_0} - \nabla \left(\frac{\mathbf{B}^2}{2\mu_0} \right)$$

$$(III) \quad \mathbf{E} + \mathbf{v} \times \mathbf{B} = 0 \quad (IV) \quad \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \quad (V) \quad \mu_0 \mathbf{J} = \nabla \times \mathbf{B}$$

$$(VI) \quad \nabla \cdot \mathbf{B} = 0 \quad (VII) \quad \frac{d}{dt} \left(\frac{p}{\rho^\gamma} \right) = 0 \quad (VIII) \quad Ha = \mathbf{B} L \sqrt{\frac{\sigma}{\mu_0}}$$

Eq. 4.3.1

The fundamental concept behind MHD is that magnetic fields can induce currents in a moving conductive fluid, which in turn polarizes the fluid and reciprocally changes the magnetic field itself. The set of equations that describe MHD are a combination of the Navier-Stokes equations of fluid

⁸⁶ See Previous.

dynamics and Maxwell's equations of electromagnetism (see Eq. 4.3.1). These differential equations must be solved simultaneously, either analytically or numerically. Figure 4.3.1 shows a RHR for forces in MHD.

4.3.2 Case Study - Dynamics of a Q2D Wake Behind a Cylinder in Presence of MHD Environment

A confined laminar viscous flow past a two-dimensional bluff body in the presence of a strong uniform magnetic field is investigated by [Hamid, et al.]⁸⁷. The effects of Reynolds number (Re) and Hartmann number (Ha) on the dynamics of the wake are examined, with a focus on the shedding frequency and the distribution of the wake vortices. These two parameters are of primary interest as they play an important role in determining the mixing and heat transfer properties of the downstream flow. The results indicates that the imposed magnetic field significantly alters the dynamic behavior of the wake behind a cylinder. It is well-known that beyond a critical Re , the flow around a circular cylinder generates a regular pattern of vortices known as the Karman vortex street. Analysis of such bluff body wakes are typically divided into three main focus areas: the correlation between drag coefficient, base pressure and shedding frequency; the vortex dynamics, where the formation and re-arrangement process are addressed; and the stability of the mean velocity profile in the wake. When a strong magnetic fluid is imposed to a conducting fluid, the resulting wake possesses a distinct features as compared to the normal hydrodynamic flows. Typical example of such flows is in fusion power-reactor breeding blankets, where an electrically conducting fluid flows in channels within the blankets under a strong plasma-confining magnetic field. This class of flows are known as **Magneto-Hydro-Dynamic (MHD)**. The interaction between induced electric currents and the applied magnetic field results in an electromagnetic **Lorentz force**, which in turn gives a damping effect to the flow and subsequently alters the formation of vortex street.

4.3.2.1 Numerical Method and Geometry

In the current investigation a flow of electrically conducting fluid passing over a circular cylinder placed on the centerline of a duct is considered. Figure 4.3.2 depicts the numerical domain and the corresponding macro-element mesh. The ratio of cylinder diameter to the duct width (i.e. blockage ratio, $b = d/2L$) is fixed at 0.1 throughout this study. Also shown in the figure is a typical Hartmann velocity profile, characterized by a flat profile in the core with velocity U_0 and high gradients in the vicinity of the lateral walls. The length scale is normalized by the half channel width, L . However, for the sake of discussions, the Re and the geometrical length in the succeeding discussions are presented in cylinder diameter scale, d . The use of two different length scales in an MHD cylinder wake flows is inevitable: the two-dimensional linear braking term is govern by Ha and L , whereas the Re and thus

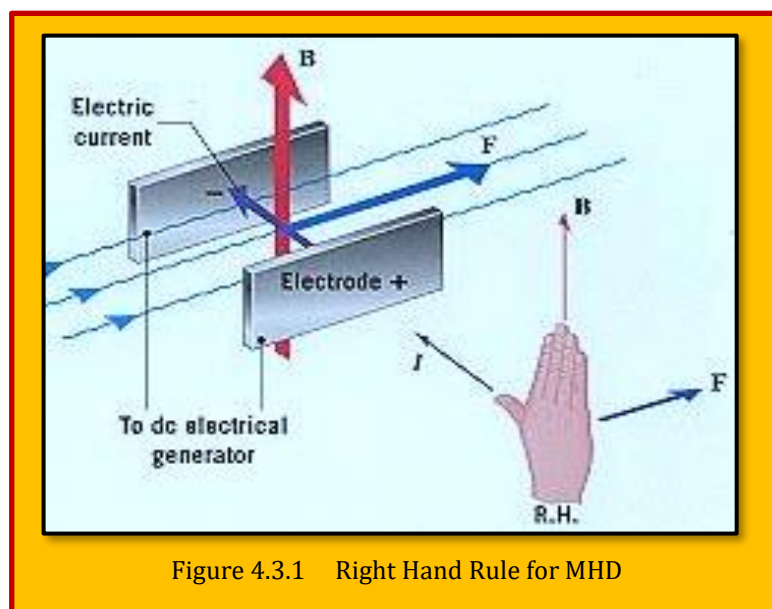


Figure 4.3.1 Right Hand Rule for MHD

⁸⁷ A. H. A. Hamid, W. K. Hussam and G. J. Sheard, "Dynamics of a Quasi-Two-Dimensional Wake Behind a Cylinder in an MHD Duct Flow with a Strong Axial Magnetic Field", 19th Australasian Fluid Mechanics Conference, Melbourne, Australia, 8-11 December 2014.

the structure of the cylinder wake is govern by d⁸⁸. A quasi-two-dimensional (Q2D) model for MHD duct flow is employed⁸⁹. Under this model, the non-dimensional magneto-hydro-dynamic equations of continuity and momentum reduce to

$$\nabla \cdot \mathbf{u} = 0 \quad , \quad \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} - 2 \frac{\text{Ha}}{\text{Re}} \mathbf{u} \quad \text{Eq. 4.3.2}$$

where $\mathbf{u}$ and p are the velocity and pressure fields, respectively. The governing equations are discretized using a high-order, in-house solver based on the spectral-element method.

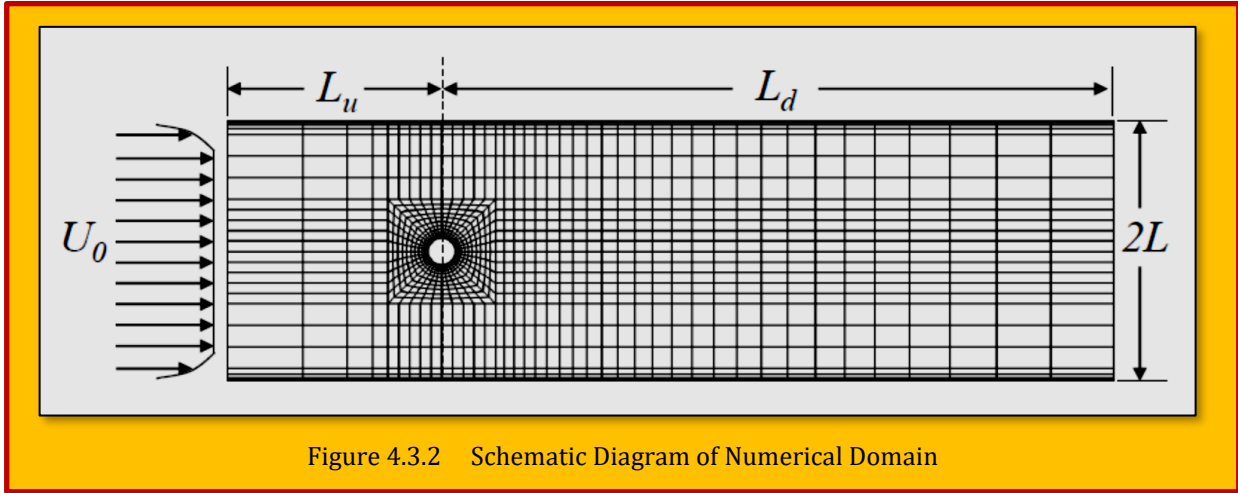


Figure 4.3.2 Schematic Diagram of Numerical Domain

4.3.2.2 Result and Discussion

In all simulations, two basic regions of wake vortices are apparent; a formation region in which the vorticity evolved from cylinder boundary-layers organizes into a vortex street, and a stable region in which the shed vortices convect downstream in a periodic laminar manner. This section presents the results of shedding frequency analysis and vortex distributions. In the current investigation, the effect of axial magnetic field on shedding frequency is of interest. It is to be noted that $H = 0$ correspond to hydrodynamic flows. The dimensionless frequency is represented by the **Strouhal number**, $\text{St} = f d = U_0$, where f is shedding frequency, calculated from the fluctuating lift force imparted on the cylinder due to the nearwake flow unsteadiness.

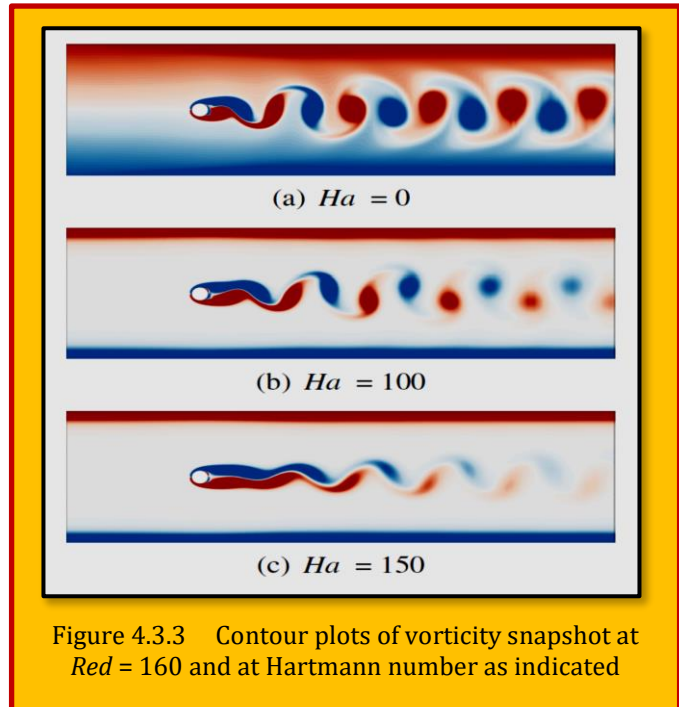


Figure 4.3.3 Contour plots of vorticity snapshot at $\text{Red} = 160$ and at Hartmann number as indicated

⁸⁸ Frank, M., Barleon, L. and M"uller, U., 2001, "Visual analysis of two-dimensional magnetohydrodynamics", *Physics of Fluids*, 13, 2287.

⁸⁹ Sommeria, J. and Moreau, R., 1982, "Why, how, and when, MHD turbulence becomes two-dimensional", *Journal of Fluid Mechanics*, 118, 507–518.

The Strouhal number is dependent on both Ha and Re . In the range of the Ha and Re considered here, St increases with increasing Ha at a given Re . This observation can be attributed to the fact that the imposed magnetic field tends to stretch the shear layer at the near wake, and hence mass conservation requires that the wake advection velocity, U_w is increased. It can be seen in [Figure 4.3.3](#) that stronger magnetic field intensity produces a narrower wake, thus extending the formation region behind the cylinder before the shear layer roll up into a vortex street. For detailed discussion, please see [\[\[Hamid, et al.\]⁹⁰](#). In conclusion, The present study has investigated the characteristics of wakes behind a circular cylinders in a rectangular duct under a strong axial magnetic field using a spectral-element method. It is found that the formation of vortex shedding and the direction of the imposed magnetic field play significant roles in determining the shedding frequency. The present investigation reveals that an axial magnetic field tends to appreciably increase the St , regardless of flow Re . Furthermore, the advection speed of wake vortices is also a strong function of both Ha and Re , whereas U_w is only weakly dependent on Re for hydrodynamic flows.

4.4 Maxwell's Equations - Electromagnetic Waves

4.4.1 Historical Perspective

In 1845, Faraday demonstrated that a magnetic field produces a measurable effect on a beam of light. This prompted him to speculate that light involves oscillation of electric and magnetic field lines, but his limited mathematical ability prevent him from pursuing this idea. Maxwell, a young admirer of Faraday, believed that the closeness of these two numbers, speed of light and the inverse square root of ϵ_0 and μ_0 , was more than just coincidence and decide to develop Faraday's hypothesis. In 1865, he predicted the existence of electromagnetic waves that propagate at the speed of light.

4.4.2 The Finite-Difference Time-Domain Method (FDTD)

The Finite-Difference Time-Domain method (FDTD) is today's one of the most popular technique for the solution of electromagnetic problems⁹¹. It has been successfully applied to an extremely wide variety of problems, such as scattering from metal objects and dielectrics, antennas, micro strip circuits, and electromagnetic absorption in the human body exposed to radiation. The main reason of the success of the FDTD method resides in the fact that the method itself is extremely simple, even for programming a three-dimensional code. The technique was first proposed by [K. Yee]⁹², and then improved by others in the early 70s. The theory on the basis of the FDTD method is simple. To solve an electromagnetic problem, the idea is to simply discretize, both in time and space, the Maxwell's equations with central difference approximations. The originality of the idea of Yee resides in the allocation in space of the electric and magnetic field components, and the marching in time for the evolution of the procedure. To better understand the theory of the method, we will start considering a simple one-dimensional problem. Assume, at this stage, "free space" as propagation medium. In this case, Maxwell's equations can be written as

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon_0} \nabla \times \mathbf{H} \quad , \quad \frac{\partial \mathbf{H}}{\partial t} = \frac{1}{\mu_0} \nabla \times \mathbf{E}$$

Eq. 4.4.1

4.4.3 Strengths of FDTD Modeling

Every modeling technique has strengths and weaknesses, and the FDTD method is no different⁹³.

⁹⁰ See 125.

⁹¹ Lecture Series, Utah ECE.

⁹² Kane Yee (1966). "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media". *IEEE Transactions on Antennas and Propagation*. **14** (3): 302–307.

⁹³ Wikipedia.

- FDTD is a versatile modeling technique used to solve Maxwell's equations. It is intuitive, so users can easily understand how to use it and know what to expect from a given model.
- FDTD is a time-domain technique, and when a broadband pulse (such as a Gaussian pulse) is used as the source, then the response of the system over a wide range of frequencies can be obtained with a single simulation. This is useful in applications where resonant frequencies are not exactly known, or anytime that a broadband result is desired.
- Since FDTD calculates the E and H fields everywhere in the computational domain as they evolve in time, it lends itself to providing animated displays of the electromagnetic field movement through the model. This type of display is useful in understanding what is going on in the model, and to help ensure that the model is working correctly.
- The FDTD technique allows the user to specify the material at all points within the computational domain. A wide variety of linear and nonlinear dielectric and magnetic materials can be naturally and easily modeled.
- FDTD allows the effects of apertures to be determined directly. Shielding effects can be found, and the fields both inside and outside a structure can be found directly or indirectly.
- FDTD uses the E and H fields directly. Since most EMI/EMC modeling applications are interested in the E and H fields, it is convenient that no conversions must be made after the simulation has run to get these values.

4.4.4 Weaknesses of FDTD Modeling

- Since FDTD requires that the entire computational domain be gridded, and the grid spatial discretization must be sufficiently fine to resolve both the smallest electromagnetic wavelength and the smallest geometrical feature in the model, very large computational domains can be developed, which results in very long solution times. Models with long, thin features, (like wires) are difficult to model in FDTD because of the excessively large computational domain required. Methods such as Eigen mode Expansion can offer a more efficient alternative as they do not require a fine grid along the z-direction.
- There is no way to determine unique values for permittivity and permeability at a material interface.
- Space and time steps must satisfy the CFL condition, or the leapfrog integration used to solve the partial differential equation is likely to become unstable.
- FDTD finds the E/H fields directly everywhere in the computational domain. If the field values at some distance are desired, it is likely that this distance will force the computational domain to be excessively large. Far-field extensions are available for FDTD, but require some amount of post processing.
- Since FDTD simulations calculate the E and H fields at all points within the computational domain, the computational domain must be finite to permit its residence in the computer memory. In many cases this is achieved by inserting artificial boundaries into the simulation space. Care must be taken to minimize errors introduced by such boundaries. There are a number of available highly effective absorbing boundary conditions (ABCs) to simulate an infinite unbounded computational domain. Most modern FDTD implementations instead use a special absorbing "material", called a perfectly matched layer (PML) to implement absorbing boundaries.

Because FDTD is solved by propagating the fields forward in the time domain, the electromagnetic time response of the medium must be modeled explicitly. For an arbitrary response, this involves a computationally expensive time density, although in most cases the time response of the medium (or Dispersion (optics)) can be adequately and simply modeled using either the recursive convolution (RC) technique, the Auxiliary Differential Equation (ADE) technique, or the Z-transform technique. An alternative way of solving Maxwell's equations that can treat arbitrary dispersion

easily is the Pseudo Spectral Spatial-Domain method (PSSD), which instead propagates the fields forward in space⁹⁴.

4.4.5 Case Study - 1D Maxwell Equation

In the one-dimensional case, we can use only E_x and H_y , and **Eq. 4.4.1** can be rewritten as

$$\frac{\partial E_x}{\partial t} = -\frac{1}{\epsilon_0} \frac{\partial H_y}{\partial z}, \quad \frac{\partial H_y}{\partial t} = -\frac{1}{\mu_0} \frac{\partial E_x}{\partial z}$$

Eq. 4.4.2

that represents a plane wave traveling in the z direction. Yee's scheme consists in considering E_x and H_y shifted in space by half a cell and in time by half a time step when considering a central difference approximation of the derivatives. In such a case, equations can be written as

$$\frac{E_x^{n+\frac{1}{2}}(k) - E_x^{n-\frac{1}{2}}(k)}{\Delta t} = -\frac{1}{\epsilon_0} \frac{H_y^n\left(k + \frac{1}{2}\right) - H_y^n\left(k - \frac{1}{2}\right)}{\Delta z}$$

Eq. 4.4.3

$$\frac{H_y^{n+1}(k + 1/2) - H_y^n(k + 1/2)}{\Delta t} = -\frac{1}{\mu_0} \frac{E_x^{n+\frac{1}{2}}(k + 1) - E_x^{n+\frac{1}{2}}(k)}{\Delta z}$$

Eq. 4.4.4

Eq. 4.4.3 & **Eq. 4.4.4** show the usefulness of Yee's scheme in order to have a central difference approximation for the derivatives. In particular, the left term in **Eq. 4.4.3** says that the derivative of the E field at time $n\Delta t$ can be expressed as a central difference using E field values at times $(n+1/2)\Delta t$ and $(n-1/2)\Delta t$. The right term in **Eq. 4.4.3** approximates instead the derivative of the H field at point $k\Delta x$ as a central difference using H field values at points $(k+1/2)\Delta x$ and $(k-1/2)\Delta x$. This scheme is known as "leap-frog" algorithm. Practically, it means that to approximate Maxwell's equations in space and time using this algorithm, one should calculate first all H field values, then all E field values, remembering always that E and H are shifted also in space by half of the discretization Δx . **Figure 4.4.1** shows schematically the algorithm.

4.4.5.1 Boundary Conditions

From the previous discussion, it is not clear what happens at the mesh termination. Of course, we cannot simulate the propagation of the signal indefinitely, and we need to terminate somehow the FDTD grid. The problem does not exist in the case of a spatially limited structure, like a waveguide, a resonator, etc., where we need to model a region that "trap" the field inside. In most of the problems, however, we need to simulate open space regions. In these cases, since our simulation region MUST be limited, we need to find a way to "simulate" the open space. These boundary conditions are called Radiation Boundary Conditions (RBCs) or Absorbing Boundary Conditions (ABCs). The absorbing boundary condition for the 1D case can be therefore expressed by

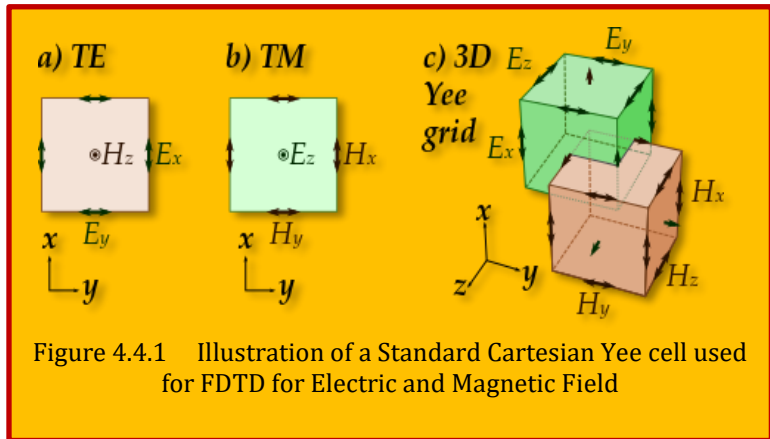


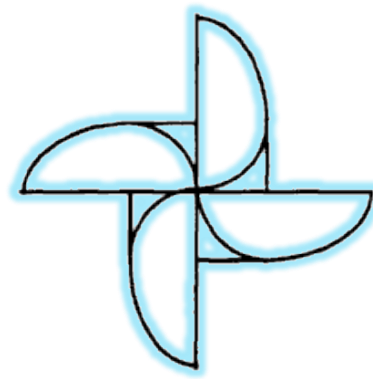
Figure 4.4.1 Illustration of a Standard Cartesian Yee cell used for FDTD for Electric and Magnetic Field

⁹⁴ Wikipedia.

$$E_x^{n+\frac{1}{2}}(1) = E_x^{n-2+\frac{1}{2}}(2) \quad , \quad E_x^{n+\frac{1}{2}}(KE) = E_x^{n-2+\frac{1}{2}}(KE - 1)$$

Eq. 4.4.5

for the right side of the mesh, and KE represents the size of the arrays E and H. With these conditions, in the 1D simulation described in the previous section the wave will be completely “absorbed” by the termination. Of course, “completely” means actually “relatively”, since for numerical errors some small reflections from the boundary (noise) will be observed.



5 Appendix A

5.1 Routine for Inverse Distance Weighted Interpolation (Shepard's Method)

```

#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <math.h>
#include "cse.h"

#define Large 1.0e+30
#define Small -1.0e+30

void RHS (double dx[],double dy[],double dz[],int num,double dum_x,
          double dum_y,double dum_z,double omega[]) {

    int i ;
    double dx_new , dy_new , dz_new , DX , DY , DZ;
    dum_x = dum_y = dum_z = 0.0 ;
    for ( i = 0 ; i < num ; i++ ){
        dx_new = dx[i] ;
        dz_new = dz[i] ;
        if (dz_new <= 0.0 ) {
            dy_new = dy[i] ;
        }
        else {
            dy_new = dy[i]+dz_new*0.25 ;
        }
        DX = dx_new - dx[i] ;
        DY = dy_new - dy[i] ;
        DZ = dz_new - dz[i] ;

        dum_x += omega[i]*DX ;
        dum_y += omega[i]*DY ;
        dum_z += omega[i]*DZ ;
    }
}

void get_weight (double h[], double omega[] , int num) {

    int i ;
    double F2 ,F[num] ,hmax;
    F2 = 0.0 ;

    hmax = Small ;
    for ( i = 0 ; i < num ; i++ ){
        hmax = (h[i] > hmax ) ? h[i]: hmax ;
        omega[i] = 1.0 ;
    }
}

```

```

        for ( i = 0 ; i < num ; i++ ){
            F[i] = ((hmax -h[i])/hmax*h[i]) * ((hmax -h[i])/hmax*h[i]) ;
            F2 += F[i] ;
        }

        for ( i = 0 ; i < num ; i++ ){
            omega[i] = F[i]/F2 ;
            if (h[i] == 0.0 ) omega[i] = 1.0 ;
            if (omega[i] < 0.0 || omega[i] > 1.0 ) {
                fprintf(stderr, " omega[i] = %.3lf \n", omega[i]);
                fprintf(stderr, " Error - The weight function should be between 0 and 1\n");
                exit(1) ;
            }
        }
    }

double Get_R (double x[],double y[],double z[],int num )
{
    double xmax,ymax,zmax,xmin,ymin,zmin,R;
    xmax = ymax = zmax = Small ;
    xmin = ymin = zmin = Large ;
    for ( int i = 0 ; i < num ; i++ ){
        xmax = (x[i]> xmax ) ? x[i]: xmax ;
        ymax = (y[i]> ymax ) ? y[i]: ymax ;
        zmax = (z[i]> zmax ) ? z[i]: zmax ;
        xmin = (x[i]< xmin ) ? x[i]: xmin ;
        ymin = (y[i]< ymin ) ? y[i]: ymin ;
        zmin = (z[i]< zmin ) ? z[i]: zmin ;
    }
    return R = sqrt ((xmax-xmin)*(xmax-xmin) +
                    (ymax-ymin)*(ymax-ymin) +
                    (zmax-zmin)*(zmax-zmin)) ;

}

void ELDW (int num_interface_nodes1, int num_nodes1, int num_outer_nodes1)

/* current global search is insufficient and CPU intensive and should be modified to
a more localized search method
*/
{

    double dum_x,dum_y,dum_z, omega[num_interface_nodes1],h[num_interface_nodes1],
        xx[2],yy[2],zz[2],xmin,xmax,ymin,ymax,zmin,zmax,R_solid,R_fluid,hmin,
        d1,d2,d3,box;
    int num_nodes,num_interface_nodes,line,i_bar,i,j,ii,n_solid;
    bool skip ;
    double dx[num_interface_nodes1],dy[num_interface_nodes1],dz[num_interface_nodes1];
    double x[num_nodes1],y[num_nodes1],z[num_nodes1];

```



```

// Read the data (test - VTK format / CSE )
int index = 0 ;
FILE* file_ptr ;
file_ptr = fopen("dum0", "r") ;
if (file_ptr == NULL) {
    printf (" ** Error opening dum0 file.\n") ;
    exit (1) ;
}
int n0 ;

fscanf (file_ptr, "%d\n", &n0);
for (line = 0 ; line < n0/2 ; line++) {
    fscanf (file_ptr,"%lf %lf %lf %lf %lf %lf",&xx[0],&yy[0],&zz[0],&xx[1],&yy[1],&zz[1])
;

    for (i_bar = 0 ; i_bar < 2 ; i_bar++ ) {
        i = i_bar ;
        if (line > 0 ) i = line * 2 + i_bar ;
        dx[i+index] = xx[i_bar] ;
        dy[i+index] = yy[i_bar] ;
        dz[i+index] = zz[i_bar] ;
    }
}

index = n0 ;
printf(" Finish reading the dum0 file. \n");
fclose (file_ptr) ;

file_ptr = fopen("dum1", "r") ;
if (file_ptr == NULL) {
    printf (" ** Error opening dum1 file.\n") ;
    exit (1) ;
}
int n1 ;
fscanf (file_ptr, "%d\n", &n1);
for (line = 0 ; line < n1/2 ; line++) {
    fscanf (file_ptr,"%lf %lf %lf %lf %lf %lf",&xx[0],&yy[0],&zz[0],&xx[1],&yy[1],&zz[1])
;

    for (i_bar = 0 ; i_bar < 2 ; i_bar++ ) {
        i = i_bar ;
        if (line > 0 ) i = line * 2 + i_bar ;
        dx[i+index] = xx[i_bar] ;
        dy[i+index] = yy[i_bar] ;
        dz[i+index] = zz[i_bar] ;
    }
}
index = n0+n1;
dx[index-1]= -4.6567497253 ;
dy[index-1]= -0.0067161722109 ;
dz[index-1]= -0.55055594444 ;
printf(" Finish reading the dum1 file. \n");

```

```

fclose (file_ptr) ;

//FILE* file_ptr ;
file_ptr = fopen("dum2", "r") ;
if (file_ptr == NULL) {
    printf (" ** Error opening dum2 file.\n") ;
    exit (1) ;
}
int n2 ;
fscanf (file_ptr, "%d\n", &n2);
for (line = 0 ; line < n2/2 ; line++) {
    fscanf (file_ptr,"%lf %lf %lf %lf %lf %lf",&xx[0],&yy[0],&zz[0],&xx[1],&yy[1],&zz[1]) ;
    for (i_bar = 0 ; i_bar < 2 ; i_bar++ ) {
        i = i_bar ;
        int index = n0+n1 ;
        if (line > 0 ) i = line * 2 + i_bar ;
        dx[i+index] = xx[i_bar] ;
        dy[i+index] = yy[i_bar] ;
        dz[i+index] = zz[i_bar] ;
    }
}
index = n0+n1+ n2;
printf(" Finish reading the dum2 file. \n");
fclose (file_ptr) ;

//FILE* file_ptr ;
file_ptr = fopen("dum3", "r") ;
if (file_ptr == NULL) {
    printf (" ** Error opening dum3 file.\n") ;
    exit (1) ;
}
int n3 ;
fscanf (file_ptr, "%d\n", &n3);
for (line = 0 ; line < n3/2 ; line++) {
    fscanf (file_ptr,"%lf %lf %lf %lf %lf %lf",&xx[0],&yy[0],&zz[0],&xx[1],&yy[1],&zz[1])
;
    for (i_bar = 0 ; i_bar < 2 ; i_bar++ ) {
        i = i_bar ;
        int index = n0+n1+n2 ;
        if (line > 0 ) i = line * 2 + i_bar ;
        dx[i+index] = xx[i_bar] ;
        dy[i+index] = yy[i_bar] ;
        dz[i+index] = zz[i_bar] ;
    }
}
index = n0+n1+ n2+n3;
num_interface_nodes = index ;

printf(" Finish reading the dum3 file. \n");

```

```

fclose (file_ptr) ;
printf(" n0 n1 n2 n3 = %d,%d,%d,%d \n", n0,n1,n2,n3);
printf(" num_interface_nodes = %d \n", num_interface_nodes);

file_ptr = fopen("internal", "r") ;
if (file_ptr == NULL) {
    printf (" ** Error opening internal file\n.");
    exit (1) ;
}
fscanf (file_ptr, "%d\n", & num_nodes);

// check for memory
int *cfddpointer ;
cfddpointer = (int*) malloc (sizeof(num_nodes));
if (cfddpointer == NULL) {
    printf (" num_nodes = %d \n", num_nodes);
    printf (" **Error - could not allocate memory for cfd data.\n");
    exit (1);
}

for (line = 0 ; line < num_nodes/2 ; line++) {
    fscanf (file_ptr," %lf %lf %lf %lf %lf
%lf",&xx[0],&yy[0],&zz[0],&xx[1],&yy[1],&zz[1]) ;
    for (i_bar = 0 ; i_bar < 2 ; i_bar++) {
        i = i_bar ;
        if (line > 0 ) i = line * 2 + i_bar ;
        x[i] = xx[i_bar] ;
        y[i] = yy[i_bar] ;
        z[i] = zz[i_bar] ;
    }
}
fclose (file_ptr) ;
printf(" Finished reading the internal file. \n");
printf(" num_nodes = %d \n", num_nodes);

// get max /min
xmax = ymax = zmax = Small ;
xmin = ymin = zmin = Large ;
for ( i = 0 ; i < num_interface_nodes ; i++ ){
    xmax = (dx[i]> xmax ) ? dx[i]: xmax ;
    ymax = (dy[i]> ymax ) ? dy[i]: ymax ;
    zmax = (dz[i]> zmax ) ? dz[i]: zmax ;
    xmin = (dx[i]< xmin ) ? dx[i]: xmin ;
    ymin = (dy[i]< ymin ) ? dy[i]: ymin ;
    zmin = (dz[i]< zmin ) ? dz[i]: zmin ;
}

```

```

    R_solid = Get_R (dx,dy,dz,num_interface_nodes) ;
    R_fluid = Get_R (x,y,z,num_nodes) ;

// loop for each cfd field ( EIDW - Shepard's Method)

    n_solid = 0 ;
    box = R_fluid;
    for (j = 0 ; j < num_nodes ; j++) {
        skip = false ;
        double xf = x[j] ;
        double yf = y[j] ;
        double zf = z[j] ;
// get Euclidian distances and normalized weights
        hmin = Large ;
        for ( i = 0 ; i < num_interface_nodes ; i++ ){
            d1 = xf - dx[i] ;
            d2 = yf - dy[i] ;
            d3 = zf - dz[i] ;
            h[i] = sqrt(d1*d1 + d2*d2 + d3*d3) ;
            hmin = (h[i] < hmin ) ? h[i]: hmin ;
        }

        printf (" pass 1 j = %d \n",j) ;

// get weight function values
        get_weight (h , omega , num_interface_nodes) ;
        RHS (dx,dy,dz,num_interface_nodes,dum_x,dum_y,dum_z,omega) ;
// update new field positions for cfd
        if (skip) {
            x[j] = xf ;
            y[j] = yf ;
            z[j] = zf ;
        }
        else {
            x[j] = dum_x + x[j] ;
            y[j] = dum_y + y[j] ;
            z[j] = dum_z + z[j] ;
        }
    }
    printf (" end of big loop ..... \n");

// output (test - VTK format / CSE )

    file_ptr = fopen("fluid_mesh_new", "w") ;
    if (file_ptr == NULL) {
        printf (" ** Error opening cfd_mesh_new file to write." ) ;
        exit (1) ;
    }
    printf (" trying to write \n") ;
    printf (" cfd.num_nodes =  %d \n", num_nodes) ;

```

```

        for (line = 0 ; line < num_nodes/2 ; line++) {
            for (i_bar = 0 ; i_bar < 2 ; i_bar++) {
                ii = i_bar ;
                if (line > 0 ) ii = line * 2 + i_bar ;
                xx[i_bar] = x[ii];
                yy[i_bar] = y[ii];
                zz[i_bar] = z[ii];
            }
            fprintf (file_ptr,"      %.11f %.11f %.11f %.11f %.11f %.11f\n",
                    xx[0],yy[0],zz[0],xx[1],yy[1],zz[1]) ;
        }
        fclose (file_ptr) ;
        printf(" Done...\n");
    }

int main()
{
#define num_interface_nodes1 30000
#define num_nodes1 400000
#define num_outer_nodes1 50000

// check memory requirements

    int * Workarray = NULL;
    Workarray = (int*) malloc (sizeof(num_interface_nodes1));
    if (NULL == Workarray) {
        printf (" num_interface_nodes1 = %d \n", num_interface_nodes1);
        printf (" **Error - could not allocate memory for cfd data.\n");
        exit (1);
    }
    Workarray = (int*) malloc (sizeof(num_nodes1));
    if (Workarray == NULL) {
        printf (" num_nodes1 = %d \n", num_nodes1);
        printf (" **Error - could not allocate memory for cfd data.\n");
        exit (1);
    }
    Workarray = (int*) malloc (sizeof(num_outer_nodes1));
    if (Workarray == NULL) {
        printf (" num_outer_nodes1 = %d \n", num_outer_nodes1);
        printf (" **Error - could not allocate memory for cfd data.\n");
        exit (1);
    }

    ELDW (num_interface_nodes1,num_nodes1,num_outer_nodes1) ;
    return 0 ;
}

```