

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Numerical Simulation of Spark Ignition Engines

Arash Mohammadi  
*KNToosi University of Technology*  
Iran

## 1. Introduction

Optimization processes for internal combustion engines requires application of advanced development tools. In addition to experimental method, numerical calculations are needed in order to obtain an insight into the complex phenomena in-cylinder processes. This chapter was organized for advances in computational fluid dynamic, heat transfer, and chemical reactions as applied to internal combustion engines. It provided an opportunity for specialists in the area of modeling flow and combustion in SI engines to provide powerful computational tools in the area of endeavor surrounding fluid dynamics. We examine computations for reactive flows in general with computational combustion in particular. In reactive flows, the conservation equations for chemical species are added to the Navier - Stokes system of equations. This addition also requires a modification of the energy equation. Furthermore, the sensible enthalpy is coupled with the chemical species, which contributes to the heat source and diffusion of species interacting with temperature. Thus, the chemically reactive flows and combustion require significant modifications of not only the governing equations, but also the existing computational methods discussed in this chapter. The aims of these computational developments have been: (1) to develop reliable engine tools for predicting the flow, temperature, etc fields; (2) to reduce the cost of current experiments that are used for most of the relevant engineering design; (3) to allow for a better understanding of physiochemical processes involved. Details of turbulent fluid motion in engines are required for determining combustion, thermal efficiencies, and level of emissions for development of cleaner, less noisy, and more efficient engines for a variety of designs and fuels.

## 2. Model building

The first step in numerical simulation consists in the construction of model describing and the technical processes. The prerequisite for this is that real process can be divided into single procession sections and hence broken down into partial problems. These partial problems must be physically describable and mathematically formulable. For the construction of parametrical models and for the simulation of temporally and spatially variable fluid, temperature and concentration fields with chemical reactions, the knowledge of thermodynamics, fluid dynamics, and combustion technology is an essential prerequisite.

Essentially, the modeling procedure can be subdivided into the following steps:

1. Define the system and boundaries from the environment, determine the relevant reservoirs as well as mass and energy flow between them.

2. Draw up balance sheets according to the unified scheme.
3. With the help of physical laws, describing the mass, momentum and energy flows.
4. Integrate the governing equations of model numerically, i.e. execute the simulation.
5. Validate the model, i.e. compare the calculated data with experimentally that existed or obtained.

### 3. Computational mesh

The discrete locations at which variables are to be calculated are defined by the numerical grid which is essentially a discrete representation of the geometric domain on which the problem should be solved. It divides the solution domain into a finite number of sub domains. Mesh generation is often the most decisive and most strongly limiting factor in CFD calculation today. A good computational mesh should consist of hexahedrons, be well adapted (i.e. also keep to the  $y^+$  rule that will discuss in 4.1.3) and sufficiently fine and adapted, so that the flow structure (free jet, flames,..) can be resolved. Another problem is mesh movement, which is solved by CFD code specifically, so that one has to rely closely in generating moving mesh.

#### 3.1 Grid generation

Regular or structured grid consists of families of grid lines with the property that members of single family do not cross each other and cross each member of the other family at once. The position of any grid point (or control volume) within the domain is uniquely identified by set of two or three indices for two or three dimensional problem, e.g. (i, j, k). An example of structured grid is shown in Fig. 1. The disadvantage of structured grid is that they can be used for geometrically simple domain and for complex geometries like internal combustion engine with complex head and chamber geometry, intake and exhaust port and with valves that are needed to move in simulation, advanced grid generator like, ICEM-CFD should be used.

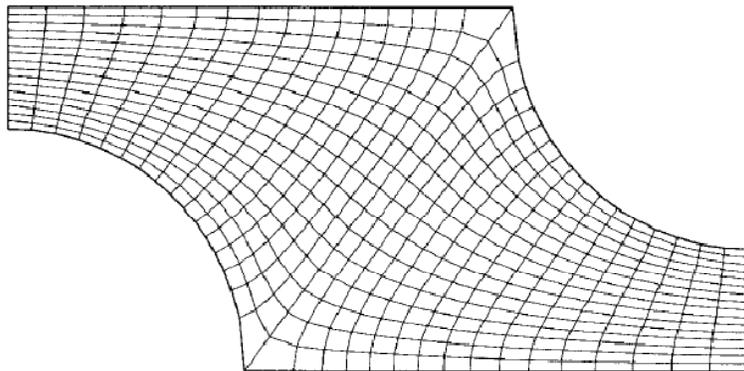


Fig. 1. Example of structured grid generation (Ferziger & Peric, 2002)

It has been shown through analysis and examples that the necessary of the numerical solution is strongly dependent on the grid or mesh used for the discretisation of the computational domain. Large mesh lengths lead to larger truncation error, whereas smaller mesh lengths increase the number of mathematical operations, resulting in larger CPU time and generate round of error. Also, larger gradients in a particular region demand finer grids that those needed coarse grid for regions where the gradients are smaller in order to resolve the changes in the variables. Complicated geometry and have smooth and finer mesh for the

regions with large gradients of variable that expected of the solution of the physical domain also passes challenge for discretisation, particularly near boundaries. Because of all these considerations, it is important to employ a grid which closely approximates a given geometry.

### 3.2 Multi zone grids

Multiple sub domains, each consisting of a specified mesh length, may be used to subdivide the given region in order to concentrate grid points near a thermal source or boundary, where large changes are expected to occur. In a block structured grid, there is two or more level subdivision of solution domain on the coarse level, there are blocks which are relatively large segments of the domain and they may or may not overlap. In Fig.2 a block structured grid with matching at interface is shown that is example of multi block structured grid.

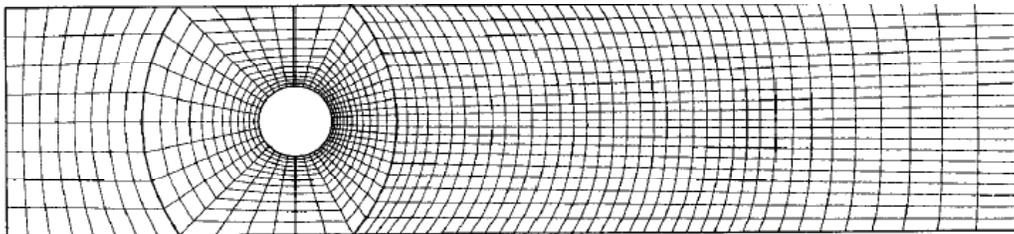


Fig. 2. Example of multi block structured grid (Ferziger & Peric, 2002)

### 3.3 Complex boundaries

In many practical problems, the boundaries are curved or have a shape that cannot be easily modeled by a regular mesh. Specialized discretisation equations may have to be developed for points near or on the boundaries. For more information, interested reader can refer to advanced books of grid generation like (Thompson, 1985).

## 4. Fluid mechanical simulations

CFD simulation is playing an increasingly important role in the simulation of engine processes, as it makes possible the most detailed physical description of the relevant processes. The most diverse processes in the engine field are considered, like charge changing, in cylinder flow, combustion, exhaust gas recirculation, and out flow process. A relatively costly process is still necessary: firstly, computational meshes must be generated, and after a definition of initial and boundary conditions, the actual calculation can be finally be started.

### 4.1 Governing equations

#### 4.1.1 The three dimensional flow fields

The basic equations that govern convective processes are obtained from conservation laws of mass, momentum, and energy. In the following, the basic equations of fluid mechanics will be briefly recapitulated. Considering a given location in the flow, the conservation of mass gives:

$$\frac{\partial \rho}{\partial t} + (\rho \vec{v}) = 0 \quad (1)$$

$\rho$  is the density of the fluid and  $\vec{v}$  the local vector velocity of fluid.

The principle of conservation of momentum, which equates the rate of change of momentum to the forces applied, (Navier - Stokes equation) gives:

$$\frac{\partial(\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \cdot \sigma + \vec{f} \quad (2)$$

$$\sigma = \mu[(\nabla \vec{v}) + (\nabla \vec{v})^T] + \lambda \nabla \cdot \vec{v} I \quad (3)$$

Where  $\nabla$  is operator of gradient,  $p$  the local pressure,  $\sigma$  viscous stress tensor for Newtonian fluid,  $\vec{f}$  is a body force per unit volume and may arise from gravitational,  $\mu$  is the first coefficient of viscosity, and  $\lambda$  is the second viscosity coefficient, related to the bulk viscosity  $\mu' = \lambda + 2/3 \mu$ . Therefore, bulk viscosity is taken as zero,  $\lambda = -2/3 \mu$ .

The energy equation is obtained from the principle of conservation of energy, as applied to a differential fluid element:

$$\frac{\partial(\rho I)}{\partial t} + \nabla \cdot (\rho \vec{v} I) = -p \nabla \cdot \vec{v} + \nabla \cdot [-k \nabla T - \rho D \sum_k h_k \nabla c_k] + \sigma : \nabla \vec{v} + S \quad (4)$$

Where  $I$  is the specific internal energy, exclusive chemical energy,  $k$  is thermal conductivity,  $T$  absolute temperature of fluid,  $D$  diffusion coefficient of mixture,  $h_k$  enthalpy diffusion of species  $k$ , and  $c_k$  concentration of species  $k$ .  $(-k \nabla T)$  is the contribution due to heat conduction.  $S$  the thermal source per unit volume and time,  $(-p \nabla \cdot \vec{v})$  the pressure work, and  $(\sigma : \nabla \vec{v})$  the viscous dissipation effects that representing the irreversible part of energy transfer due to viscous forces.

The equation set must still be completed. In compressible case however, the density must be determined. We obtain the density from the pressure by means of thermal equation of state.

$$p = \frac{\rho \bar{R} T}{M} \quad (5)$$

This equation now contain the temperature in addition, which by means of caloric equation of state:

$$I = \int_{T_0}^T c_{v(T)} dT + I_0 \quad (6)$$

That is linked with internal energy where  $I_0$  is the internal energy at reference state.

The transport equation for the concentration  $c_k$  of the particular species must be formulated,

$$\frac{\partial(\rho c_k)}{\partial t} + \nabla \cdot (\rho c_k \vec{v}) = \nabla \cdot (D_k \rho \nabla c_k) + s_k \quad (7)$$

With:

$$\sum c_k = 1 \quad (8)$$

Also, an additional diffusion term appears in the energy equation (4). If we have  $N$  species, only  $N-1$  equations of (7) should be solved. The last one obtains from equation (8).

#### 4.1.2 Turbulence models

Turbulent flows, which are of great practical importance, are three dimensional and time-dependent. Computer methods of solving the differential equations of fluid dynamics are well advanced even for three-dimensional time-dependent flows. For most engineering purposes it is unnecessary to resolve the detail of turbulent fluctuations. Only the effects of the turbulence on mean flow are usually sought. For turbulence model to be useful in general CFD code, it must have wide applicability, be accurate, simple and economical to

run. A turbulence model is a semi-empirical equation relating the fluctuating correlation to mean flow variables with various constants. When this equation is expressed as an algebraic equation, it is referred to as the zero equation models. When partial differential equations are used, they are referred to as one equation or two equation models depending on the number of PDE which utilized. These consist of sets of differential equations, and associated algebraic equations and constants, the solutions of which, in conjunction with those of the Navier-Stokes equations, closely simulate the behavior of real turbulent fluids. Two equation models of turbulence determine both the turbulence kinetic energy  $k$  and the characteristic length  $L$  from transport equations. Fortunately, the increased computational complexity is compensated by a dramatic increase in universality. The best known and most extensively tested two equation model is the so called  $\kappa$ - $\epsilon$  model, although several turbulence models have been developed for solving various turbulent flows. Among these,  $\kappa$ - $\epsilon$  model, where  $\kappa$  is the turbulent kinetic energy and  $\epsilon$  the rate of dissipation of turbulent energy, widely used. Both of these quantities are calculated from their governing differential equations. (Lander & Spalding, 1974) discussed this turbulence model. The equations for  $\kappa$  and  $\epsilon$  are solved simultaneously with the continuity, momentum, and energy equations to yield the flow and temperature distributions. For the further details see (Launder & Spalding ,1974). This model has been used by many investigator and compared with experimental data, and continues to be a popular choice for the study of turbulent flows of practical interest. At high Reynolds numbers, the transport equation for  $\kappa$  turbulent kinetic energy may be expressed:

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho u k) = -\frac{2}{3} \rho k \nabla \cdot u + \sigma : \nabla u + \nabla \cdot \left[ \left( \frac{\mu}{Pr_k} \right) \nabla k \right] - \rho \epsilon \tag{9}$$

with a similar one for the dissipation rate,  $\epsilon$ :

$$\frac{\partial \rho \epsilon}{\partial t} + \nabla \cdot (\rho u \epsilon) = -\left( \frac{2}{3} c_{\epsilon 1} - c_{\epsilon 3} \right) \rho \epsilon \nabla \cdot u + \nabla \cdot \left[ \left( \frac{\mu}{Pr_\epsilon} \right) \nabla \epsilon \right] + \frac{\epsilon}{k} [c_{\epsilon 1} \sigma : \nabla u - c_{\epsilon 2} \rho \epsilon^s] \tag{10}$$

The standard values of  $\kappa$ - $\epsilon$  turbulence model constants  $c_{\epsilon 1}$ ,  $c_{\epsilon 2}$ ,  $c_{\epsilon 3}$ ,  $Pr_k$ , and  $Pr_\epsilon$  are constants whose values are determined from experimental and some theoretical considerations. The standard values of these constants are often used in engine calculation is given in below table:

$$c_{\epsilon 1} = 1.44 \qquad c_{\epsilon 2} = 1.92 \qquad c_{\epsilon 3} = -1.0 \qquad Pr_k = 1.0 \qquad Pr_\epsilon = 1.3$$

So far we have considered convective process related to laminar flows, in which the flow is well ordered in which fluctuations and disturbances are small compared to the mean flow. Most of the convective processes that occur in industry especially, internal combustion engines, are not laminar and characterized by disturbances of large magnitude. Such flows are termed turbulent, and governing equations, may be obtained by taking the velocity, pressure, and temperature as fluctuating components superimposed on the mean values, e.g.,  $u = \bar{u} + u'$ , where  $u$  is the instantaneous value of velocity component,  $\bar{u}$  is time-averaged value, and  $u'$  fluctuating quantity. The total instantaneous quantities are substituted in the governing equations. Due to complexity with the determination of these turbulent fluctuations near solid boundaries, several simple models have been developed for the study of turbulent transport mechanism.

### 4.1.3 The turbulent law of the wall

However, there is still a problem concerning the boundary conditions. On the walls, a boundary layer is formed, in which the velocity decreasing because of friction to zero, i.e. the flow, becomes laminar. Thus in a turbulent flow the boundary layer consist of typically of laminar sub layer and a turbulent zone. The  $\kappa$ - $\epsilon$  model cannot be applied across the entire boundary layer. Moreover, the boundary layers are often so thin (especially in engines) that they are hardly numerically solvable anyway. Then usual way to overcome this problem is by deriving the so-called law of the wall. We now need a turbulent law of the wall i.e. an analytical boundary layer model, in order to calculate shear stress from the local velocities in the cell closest to the wall logarithmic law of the wall.

In engine calculations, ordinary uses turbulent law of the wall velocity condition with fixed temperature walls. For turbulent law of the wall conditions the tangential components are determined by matching to logarithmic profile:

$$\frac{v}{u^*} = \begin{cases} \frac{1}{\kappa} \ln(c_{tw} \zeta^{7/8}) + B & \zeta > R_c \\ \zeta^{1/2} & \zeta < R_c \end{cases} \quad (11)$$

Where  $\zeta = \frac{\rho y v}{\mu_{air}(T)}$  is the Reynolds number based on the gas velocity relative to the wall

which is evaluated at distance  $y$  from wall, and  $u^*$  is the shear speed, it is assumed that  $y$  is small enough to be in the laminar sub layer region of the turbulent boundary layer. The  $Re_c$  defines the boundary between these two regions. The constant  $\kappa$ ,  $c_{tw}$ ,  $R_c$  and  $B$  are related to  $\kappa$ - $\epsilon$  model constant by :

$B=5.5$ ,  $c_{tw}=0.14$ ,  $\kappa=0.437$ ,  $R_c=114$  that commonly accepted values of the  $\kappa$ - $\epsilon$  constants.

For the numerical solution, the node nearest to the wall must remain within the boundary layer.

$$T - T_w = \frac{q_w Pr}{\kappa c_p \rho v_t} (\ln y^+ + const) \quad (12)$$

A law of the wall can be derived quite analogy for the temperature. For fixed temperature walls using the turbulent law of the wall condition,  $J_w$  is determined from the modified Reynolds analogy formula:

$$\frac{J_w}{\rho u^* c_p (T - T_w)} = \begin{cases} 1 / (Pr_\epsilon \frac{v}{u^*}) & \zeta \leq R_c \\ 1 / \{Pr[\frac{v}{u^*} + (\frac{Pr_l}{Pr} - 1)R_c^{1/2}]\} & \zeta > R_c \end{cases} \quad (13)$$

$T_w$  is the wall temperature and  $Pr_l$  is the Prandtl number of the laminar fluid.

## 5. Foundation of reaction

### 5.1 Chemical equilibrium

A general chemical equilibrium reaction with  $v'_{i,s}$  and  $v''_{i,s}$  representing the stoichiometric coefficients of reaction and products for the chemical species  $M_i$



Since every chemical reaction can be in principle run both forwards as well as backwards, the reaction arrow in (14) can be replaced with an equal sign. We hence obtain the general form of conservation of species:

$$\sum_i i (v''_{i,s} - v'_{i,s}) M_i = 0 \quad (15)$$

hence the stoichiometric coefficients are negative for all primitive and positive for all products. State of equilibrium can be interpreted as a situation, in which both the forward as well as reverse reactions progress with identical speed.

$$K_p(T) = \prod_i \left( \frac{p_i}{p_0} \right)^{(v''_{i,s} - v'_{i,s})} \quad (16)$$

The equilibrium constant  $K_p$  now contains the information about the equilibrium material composition in terms of partial pressure  $p_i$  of the various species  $i$ .

## 5.2 Reaction kinetics

A one step chemical reaction of arbitrary complexity can be represented by the following stoichiometric equation:



Where  $v'_i$  are the stoichiometric coefficient of reactants and  $v''_i$  representing the stoichiometric coefficient of products,  $M_i$  molecular weight of  $i$ th chemical species, and  $N$  total number of component involved. For the chemical reaction species concentration, e.g. for the  $[M_i]$ , can be given with empirical formulation:

$$\frac{d[M_i]}{dt} = (k_f [A_a]^{v_a} [A_b]^{v_b} - k_r [A_c]^{v_c} [A_d]^{v_d}) \quad (18)$$

hence the first term on the right side describes the reaction rate of the forward direction and the second term the rate of the reverse reaction.  $k_f$  and  $k_r$  are thereby the so-called rate coefficient of the forward and reverse reactions. They must be determined experimentally for every particular chemical reaction. They are customarily represented with an Arrhenius formulation form:

$$K = AT^b \exp(-E_A / RT) \quad (19)$$

The constant  $A$  and the exponent  $b$  as well as the so-called activation energy  $E_A$  are summarized for many chemical reactions in extensive table.

## 6. Boundary conditions

Any numerical simulation can consider only a part of the real physical domain or system. The truncation of the domain leads to artificial boundaries, where we have to prescribe

values of certain physical quantities. Furthermore, walls which are exposed to the flow represent natural boundaries of the physical domain. The numerical treatment of the boundary conditions requires a particular care. An improper implementation can result in inaccurate simulation of the real system. Additionally, the stability and the convergence speed of the solution scheme can be negatively influenced. The following types of boundary conditions are in general encountered in the numerical solution of the Navier-Stokes equations:

1. Solid wall
2. Symmetry
3. Coordinate cut and periodic boundary
4. Boundary between blocks
5. Inflow or out flow
6. Pressure inlet or pressure outlet
7. Axes

## 7. Numeric

After selecting the mathematical model, suitable discretisation method should be chosen, i.e. a method for approximation of differential equations by a system of algebraic equations, for the variables at some set of discrete locations in space and time. The most important approaches are finite difference (FD), finite volume (FV) and finite element (FE) methods.

In the following, some basic concepts of numerical fluid mechanics will be explained, in order to become acquainted with its most essential concepts, which are also of importance for practical work. The description that follows refers to a so-called finite volume procedure, that is proven to be unconditionally stable, as accurate as the accuracy of current state of physical modeling permits, robust and efficient.

The starting point of the analysis is the set of three-dimensional, partial differential equations that govern the phenomena of interest here. This set consists, in general, of the following equations: the continuity equation; the three momentum equations that govern the conservation of momentum per unit mass (the velocity) in each of the three space directions (the Navier-Stokes equations); the equation for conservation of energy and species concentration, and the equations for a "turbulence model". We shall consider them in their most general form.

It is generally accepted that the turbulence model flow field in an IC engine strongly influences its combustion process, thermal efficiency and emissions. Traditionally, knowledge of flow field has been extracted from experimental investigations which tend to be costly, lengthy and inflexible. The most widely used model of turbulence in IC engine research is the  $\kappa$ - $\epsilon$  model.

### 7.1 The finite volume method

Customarily, CFD codes work with the finite volume method. This approach guarantees the numerical preservation of conservative quantities for the incompressible flows. The FV method uses the integral form of the conservation equations. The solution domain is subdivided into a finite number of control volumes, and the conservation equations are applied to each control volume. At the centroid of each CV lie computational nodes at which the variable values are to be calculated. As a result, an algebraic equation for each CV is obtained. The FV method can accommodate any type of grid, so it is suitable for complex geometries.

However, the computational mesh ideally, be built hexahedratically. The conservation law for transport of a scalar in an unsteady flow has the general form:

$$\frac{\partial}{\partial t}(\rho \Phi) + \nabla \cdot (\rho u \Phi) = \nabla \cdot (\Gamma \nabla \Phi) + S_{\phi} \tag{20}$$

$(\rho u \Phi)$  designates convection,  $(\Gamma \nabla \Phi)$  diffusion flows of and  $S_{\phi}$  the corresponding local source. With the help of Gaussian law follows and by replacing the volume integrals of convective and diffusion term we obtain:

$$\begin{aligned} & \int_{CV} \left( \int_t^{t+\Delta t} \frac{\partial}{\partial t}(\rho \Phi) dt \right) dV + \int_t^{t+\Delta t} \left( \int_A n \cdot (\rho u \Phi) dA \right) dt \\ &= \int_t^{t+\Delta t} \left( \int_A n \cdot (\Gamma \nabla \Phi) dA \right) dt + \int_t^{t+\Delta t} \left( \int_{CV} S_{\phi} dV \right) dt \end{aligned} \tag{21}$$

**7.2 The finite volume equations formulation**

Finite volume equations are derived by the integration of above differential equations over finite control volumes that taken together fully cover the entire domain of interest (Fig 3). These control volume are called "cells" Say P, for which the fluid-property value, are regarded as representative of the whole cell. It is surrounded by neighboring nodes which we shall denote by N, S, E, W, B and T. cells and nodes for velocity components are "staggered" relative to those for all other variables.

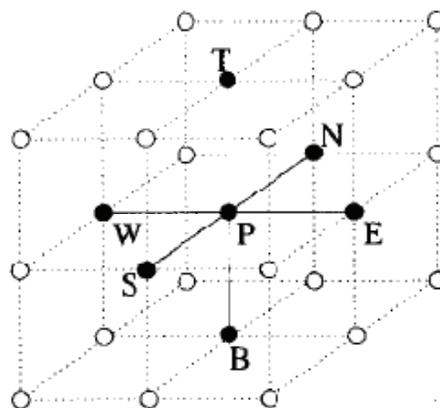


Fig. 3. Computational molecule in 3D domain(Patankar, 2002)

The integration involved is different to the usual Taylor-series expansion used in classical finite-difference technique, and results in different coefficients of algebraic equations that are finally obtained. The description that follows refers to a so called finite volume procedure, that is proven be unconditionally stable, as accurate as the accuracy of current state of physical modeling permits, robust and efficient.

**7.3 Discretisation and numerical solution of the momentum equation**

Finally, the momentum equation for the calculation of velocity and pressure by use of continuity equation should be considered. For numerical reasons, it is recommendable to resort to so called staggered grid, i.e. pressure and velocity are calculated on computational grids shifted to each other, the pressure for example in the cells and the velocity on the

nodes. The calculations of velocity commonly take place iteratively, for which several algorithms are known (e.g. SIMPLE, PISO, SIMPLER...). In final analysis, all have the fact in common that in the first step the momentum equation is solved for the velocities of momentums kept constant. In the second step, pressure corrections are then calculated with the help of a Poisson equation. For pressure with these pressure corrections, new velocities are then calculated again, and thus again, until a pre-given break off threshold for the convergence is reached.

### 7.3.1 Discretisation of transient convection diffusion equation

Transient three dimensional convection diffusion of a general property  $\Phi$  in a velocity field that  $\vec{v}$  govern by equation (20). The fully implicit discretisation equation is:

$$a_p \Phi_p = a_w \Phi_w + a_E \Phi_E + a_s \Phi_s + a_N \Phi_N + a_B \Phi_B + a_T \Phi_T + a_p^\circ \Phi_p^\circ + S_u \quad (22)$$

where:

$$a_p = a_w + a_E + a_s + a_N + a_B + a_T + a_p^\circ + \Delta F - S_p \quad (23)$$

with  $a_p^\circ = \frac{\rho_p \Delta V}{\Delta t}$  and  $\bar{S} \Delta V = S_u + S_p \Phi_p$

the neighbor coefficients of this equation for hybrid differencing scheme as follows:

$a_w$	$\max [F_w, (D_w + F_w/2), 0]$
$a_E$	$\max [-F_e, (D_e - F_e/2), 0]$
$a_s$	$\max [F_s, (D_s + F_s/2), 0]$
$a_N$	$\max [-F_n, (D_n - F_n/2), 0]$
$a_B$	$\max [F_b, (D_b + F_b/2), 0]$
$a_T$	$\max [-F_t, (D_t - F_t/2), 0]$
$\Delta F$	$F_e - F_w + F_n - F_s + F_t - F_b$

Table 1. The neighbor coefficients of this equation for hybrid differencing scheme (Patankar, 1980)

In the above expressions the values of F and D are calculated with the following formulae:

Face	W	E	S	N	B	t
F	$(\rho u)_w A_w$	$(\rho u)_e A_e$	$(\rho v)_s A_s$	$(\rho v)_n A_n$	$(\rho w)_b A_b$	$(\rho w)_t A_t$
D	$\frac{\Gamma_w}{\delta x_{WP}} A_w$	$\frac{\Gamma_e}{\delta x_{PE}} A_e$	$\frac{\Gamma_s}{\delta y_{SP}} A_s$	$\frac{\Gamma_n}{\delta y_{PN}} A_n$	$\frac{\Gamma_b}{\delta z_{BP}} A_b$	$\frac{\Gamma_t}{\delta z_{PT}} A_t$

### 7.3.2 Different approach

Employing the governing momentum equation and Poisson equation for the pressure, the numerical scheme may be based on the primitive variables. A stagger grid is employed, as shown in Fig.4 for two dimensional flows. The locations for the velocity components are placed at the faces of the control volume. If a uniform grid is used, the locations are exactly midway between the grid points. Therefore, the locations for the velocity components are agreed. Since the pressure difference between two adjacent grid points is the driving force for the velocity component. Located between these points, the finite difference

approximation is physically correct and will accept only reasonable velocity field. Also, the transport rates across the faces of control volumes can be computed without interpolation of velocity components. This approach has been adopted for several efficient schemes for flow computation, such as those discussed by (Patankar, 1980). If interest lies only in the steady state flow field and if a steady state is known to exist, the problem may be solved by iterative procedure, though time marching may also be employed to yield the steady state distribution at large time. The SIMPLER algorithm, discussed in detail by (Patankar, 1980), and outlined in the following section, for instance, employs a pressure-correction equation for correcting the velocities during iteration and a pressure equation for improving the pressure field.

### 7.3.3 The staggered grid

The finite volume method starts, as always, with discretization of the flow domain and of the relevant transport equations. First we need to decide where to store the velocities. It seems logical to define these at the same locations as the scalar variables such as pressure, Temperature, etc. However, if the velocity and pressure are both defined at the nodes of ordinary control volume, it is clear that, the influence of pressure is not properly represented in the discretization momentum equations, For more information (Blazek, 2001). A remedy for this problem is to use a staggered grid for velocity components (Harlow & Welch, 1965). The idea is to evaluate scalar variables, such as pressure, density, temperature etc., at ordinary nodal points but to calculate velocity components at staggered grids centered on the cell faces. The arrangement for a two dimensional flow calculation is shown in Fig. 4.

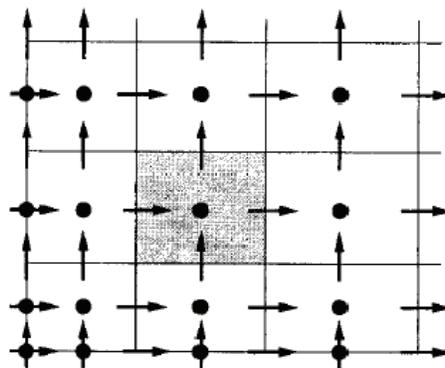


Fig. 4. Staggered grid (Patankar, 1980)

The scalar variables, including pressure, are stored at nodes marked ( $\bullet$ ). The velocities are defined at the cell faces in between the nodes and are indicated by arrows. Horizontal arrow ( $\rightarrow$ ) indicates the locations for  $u$ -velocities and vertical ( $\uparrow$ ) ones denote those for  $v$ -velocities. For the moment we continue to use the original E, W, N, S notation; the  $u$ -velocities are stored at scalar cell faces 'e' and 'w' and the  $v$ -velocities at faces 'n' and 's'. In a three dimensional flow the  $w$ -component is evaluated cell faces 't' and 'b'.

## 8. Simple algorithm

As discussed in the preceding section, the governing equation for the flow may be solved in terms of derived variables, or in term of primitive variables consisting of the velocity components and the pressure.

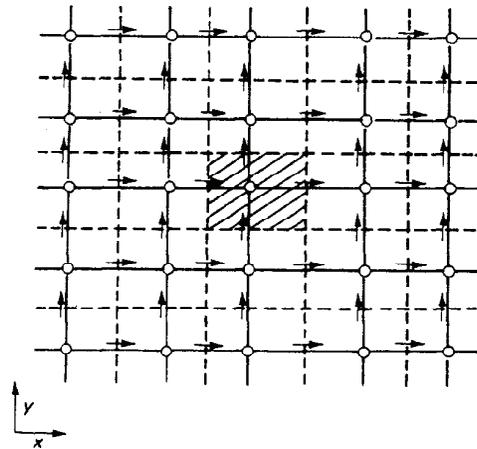


Fig. 5. Staggered location for the velocity components in a two dimensional flow (Patankar, 1980)

Still, pressure calculation has been a major hurdle in, attracting researchers to the stream function-vorticity approach even for three dimensional flows. However, in the advent of Simple (Semi Implicit Method for Pressure Linked Equations) algorithm, along with its revised version Simpler and the enhancement such as Simplec (Van Doormaal & Raithby, 1984), the solution of the equations using primitive variable approach has become very attractive. In fact, Simple and Simple like algorithms are extremely popular for the solution of problems involving convective flow and transport. The basic approach involves the control volume formulation, with the staggered grid, as outlined in the preceding section. This avoids the appearance of physically unrealistic wavy velocity fields in the solution to equations. The pressure difference between two adjacent points in the natural driving force for the velocity component located between these points and checkerboard pressure fields do not arise as possible solutions. The pressure at a chosen point is taken at arbitrary value and the pressures at other points are calculated as differences from the chosen pressure value.

Following (Patankar, 1980), if a guessed pressure field  $p^*$  is taken, the corresponding velocity field can be calculated from the discretised equations for the control volume shown in Fig. 6. These equations are of the form:

$$a_e u_e = \sum a_{nb} u_{nb}^* + b + (p_p^* - p_E^*) A_e \quad (24)$$

Where the asterisk on the velocity indicates the erroneous velocity field based on guessed pressure field. Here,  $a_{nb}$  is a coefficient that accounts for the combined convection-diffusion at the faces of the control volume, with  $nb$  referring to the neighbors  $e$  to the control volume,  $b$  includes the source terms except the pressure gradient, and  $A_e$  is the area on which pressure acts, being  $\Delta y^* \Delta z$  for 3D. The numbers of neighbor terms are 6 for three dimensional ones. Similar equations can be written for  $v_n^*$  and  $w_t^*$ , where  $t$  lies on the  $z$ -direction grid line between grid points P and T.  $p$  is the correct pressure and  $p'$  the pressure correction, we may write:

$$p = p^* + p', \quad u = u^* + u', \quad v = v^* + v', \quad w = w^* + w' \quad (25)$$

Where the prime indicates corrections needed to reach the correct values that satisfy the continuity equation. Omitting the correction terms due to the neighbors, an iterative

solution may be developed to solve for the pressure and the velocity field. Then, the velocity correction formula becomes:

$$u_e = u_e^* + \frac{A_e}{a_e} (p'_p - p'_E)$$

$$v_n = v_n^* + \frac{A_n}{a_e} (p'_p - p'_N)$$
(26)

And similarly for  $w_t$ . From the time dependent continuity equation, the pressure correction equation is then developed as:

$$a_p p'_p = a_E p'_e + a_w p'_w + a_N p'_N + a_s p'_s + a_T p'_T + a_B p'_B + b$$
(27)

Where  $b$  is a mass source which must be eliminated through pressure correction so that continuity is satisfied. Here, T and B are neighboring grid points on the z direction grid line.

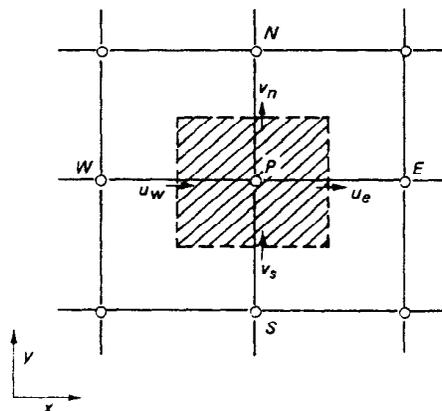


Fig. 6. Control volume for driving the pressure correction equation (Patankar, 1980)

The simple algorithm has the following main steps:

1. Guess the pressure field  $p^*$ .
2. Solve the momentum equation to obtain  $u^*, v^*$ , and  $w^*$ .
3. Solve the pressure correction equation to obtain  $p'$ .
4. Add  $p'$  to  $p^*$  to obtain the corrected pressure  $p$ .
5. Calculate  $u, v$  and  $w$  from  $u^*, v^*$  and  $w^*$  using velocity correction equations.
6. Treat the corrected pressure  $p$  as the new guess  $p^*$  and iterate the preceding procedure to convergence.

The revised version Simpler is quite similar to preceding algorithm and was developed mainly to improve the rate of convergence. In this case, the main steps are:

1. Guess the velocity field
2. Solve the pressure equation, which is similar to pressure correction equation, Eq. (26), to obtain the pressure distribution. In this equation  $p'$  is replaced by  $p$  and a different expression arise for  $b$ .
3. Treating the pressure field as  $p^*$ , solve the momentum equations to obtain  $u^*, v^*$  and  $w^*$ .
4. Solve the pressure correction equation to obtain  $p'$ .
5. Correct the velocity field but not the pressure.
6. Use the velocity field as the guessed distribution and iterate the preceding procedure to convergence.

The pressure at any arbitrary point in the computational domain is specified and pressure differentials from this value are computed. The boundary condition may be a given pressure, which makes  $p' = 0$ , or a given normal velocity which makes the velocity a known quantity at the boundary and not a quantity to be corrected so that  $p'$  at the boundary is not needed. For further details, (Ptankar, 1980) may be consulted.

## 10. Example

In the following simulation of SI internal combustion is discussed. We consider the intake, compression, combustion and exhaust process in an engine but only the results of mesh preparation and heat flux are discussed.

### 10.1 Grid generation

Prior to CFD simulation, computational mesh was generated for the engine using ICEM CFD scheme. The geometry of a mesh is composed of any arbitrary number of logical blocks that are patched together in a seamless fashion. Patching allows complex geometries to be created, block by block while minimizing the number of deactivated zones that surround the final mesh. The movement of piston/flow domain was resolved using the boundary motion feature of available code. During the solution process, as the piston moves, the internal mesh structure deforms automatically to optimize the mesh. The distortion of each individual cell occurs when the generated cell distortion reaches a certain level, and the solution is re-zoned onto new mesh. Mesh size ranged from about 90000 at BDC to about 46000 at TDC for computational studies.

### 10.2 Initial and boundary conditions

Computation starts at TDC. Initial charge densities were calculated based on ambient pressure of 0.85 bar and temperature of 300K at inlet valves opening. The standard  $\kappa - \varepsilon$  turbulence model in the code was used with an initial value of turbulent kinetic energy  $k$ , assumed 10 percent of the total kinetic energy based on the mean piston speed that supposed to be uniform. Radial velocity is initialized assuming a swirl ratio 0. Temperature was taken as 485K for liner, 600K for cylinder head and piston, 550K for intake valves, and 800K for exhaust valves.

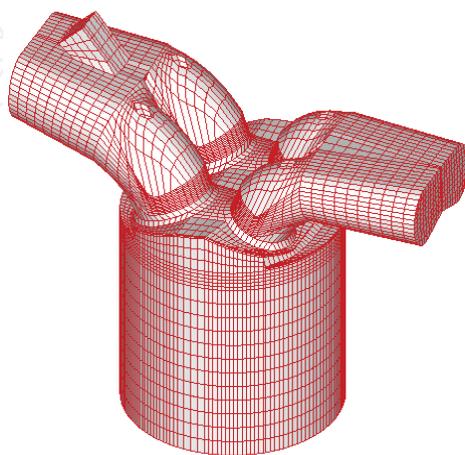


Fig. 7. Computational mesh at bottom dead center (Mohammadi et.al, 2009)

### 10.3 Discussion

In Figs. 8-a and b variations of heat flux and heat transfer coefficient on the cylinder head, liner, piston, intake and exhaust valves versus crank angle are illustrated. Heat flux has very low value in the intake and most part of compression process relative to the value of heat flux after combustion. Such quantities of heat transfer are usually negligible in experimental measurements. After combustion and release of chemical energy, heat flux rises rapidly when the flame arrives at each location, and it has maximum value at about 20 ATDC for all parts of the cylinder. Maximum heat flux and heat transfer coefficient are at intake valves that reach to 5.4 MW/m<sup>2</sup> and 3.82 kW/m<sup>2</sup>.K and the minimum value is on the cylinder wall with 1.18 MW/m<sup>2</sup> and 0.96 kW/m<sup>2</sup>.K respectively. Heat flux through combustion chamber walls is mainly due to gas-phase convection, fuel film conduction and chemical reactions.

From Fig. 8-a it can be seen that the peak heat flux take places on the intake valves because the gradient of temperature between gas and the valves surface are higher than other locations. Heat flux on the cylinder head is more than piston because flame initiate near head (spark plug is between inlet and exhaust valves) and arrives sooner to the surface of piston. Liner heat flux has lower value than other places because its location is further from flame than other locations and temperature of the wall is lower than other locations and flame is quenched near it.

Fig. 8-b shows that heat transfer coefficient on the intake and exhaust valves is almost equal and is the highest value, because the gas velocity (or Reynolds number) in this region is highest. Heat transfer coefficient on the cylinder head is more than piston and on the liner it has the lowest value where the temperature gradient and gas velocity are lowest.

After 60 degree, ATDC expansion cools the burned gases and heat flux decay to relatively low level.

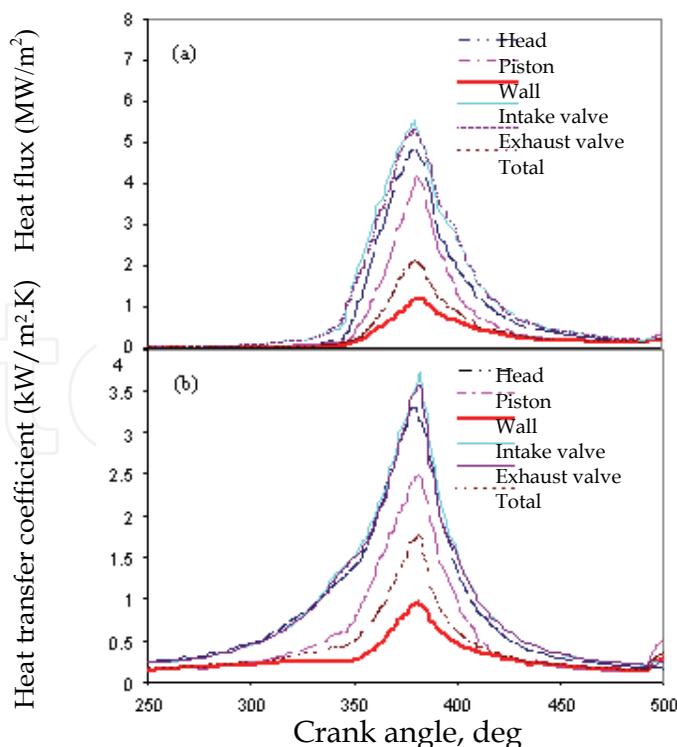


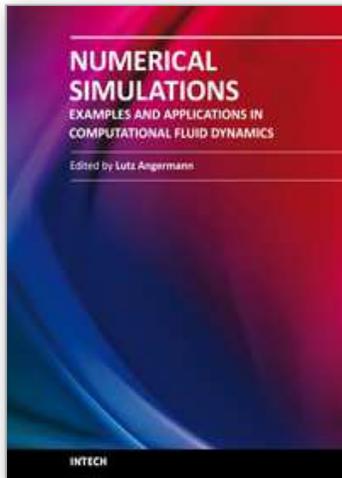
Fig. 8. Variation of: a) surface heat flux at various locations with crank angle, b) heat transfer coefficient at various locations with crank angle (Mohammadi et.al, 2009)

## 11. Conclusion

In this chapter CFD simulation of spark ignition engine was discussed. Fundamental of structured mesh generation, governing equation for Fluid mechanics, include, mass, momentum and energy equations, turbulence model and law of the wall, principle of equilibrium and kinetics reaction equation for combustion simulation were discussed. For numerical simulation of system of equations finite volume method for discretization of the equations was used, simple algorithm for solving the equations were discussed and in final one example of numerical simulation of heat flux and heat transfer coefficient in a spark ignition was discussed.

## 12. References

- Blazek, J. (2001). *Computational Fluid Dynamics : Principles and Applications*, Elsevier Science Publishing Co, ISBN 0080430090, UK.
- Chung, T. J. (2002). *Computational Fluid Dynamics*, Cambridge University Press, ISBN 0-521-59416-2, UK.
- Ferziger, J. H. & Peric, M. (2002). *Computational Method for Fluid Dynamics*, Springer, ISBN 3-540-42074-6, Germany.
- Heywood, J. B. (1988). *Internal Combustion Engine Fundamentals*, McGraw-Hill, ISBN 0-07-02863-X, USA.
- Hoffmann, A. (1989). *Computational Fluid Dynamics for Engineers*, John Wiley Engineering Educational system, ISBN 0-9623731-4-1, USA.
- Kuo, K. K. (2005). *Principle of Combustion*, John Wiley & Sons, ISBN 0-471-04689-2, USA.
- Launder, B. E. & Spalding, P. B. (1974). The numerical computation of turbulent flows, *Computer Method in Applied Mechanics and Engineering*, Vol 3 3, pp. 269-289.
- Mohammadi, A. & Yaghoubi, M. (2010). Estimation of instantaneous local heat transfer coefficient in spark ignition engines, *International Journal of thermal science*, Vol. 49, pp. 1309-1317.
- Mohammadi, A.; Jazayeri, A. & Ziabasharhagh. M. (2008). Numerical simulation of convective heat transfer in a spark ignition engine, *Proceeding of the ASME international combustion engine*, Illinois, USA, April 27-30, Chicago.
- Mohammadi, A.; Yaghoubi, M. & Rashidi, M. (2008). Analysis of local convective heat transfer in a spark ignition engine, *Journal of International Communication heat and mass transfer*, Vol 35, pp. 215-224.
- Patankar. S. V. (1980). *Numerical Heat Transfer and Fluid flow*, McGraw-Hill, ISBN 0-07-048740-5, USA.
- Thompson, J. F.; Warsi, Z. U. A, & Wayne Mastin, C. (1985). *Numerical Grid Generation*, Elsevier Science Publishing Co,.
- Versteeg, H. K. & Malalasekera, W. (1995). *An Introduction to Computational Fluid Dynamics the Finite Volume Method*, Prentice Hall, Edinburg, ISBN 0-444-00985-X, England.



**Numerical Simulations - Examples and Applications in  
Computational Fluid Dynamics**

Edited by Prof. Lutz Angermann

ISBN 978-953-307-153-4

Hard cover, 440 pages

**Publisher** InTech

**Published online** 30, November, 2010

**Published in print edition** November, 2010

This book will interest researchers, scientists, engineers and graduate students in many disciplines, who make use of mathematical modeling and computer simulation. Although it represents only a small sample of the research activity on numerical simulations, the book will certainly serve as a valuable tool for researchers interested in getting involved in this multidisciplinary field. It will be useful to encourage further experimental and theoretical researches in the above mentioned areas of numerical simulation.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Arash Mohammadi (2010). Numerical Simulation of Spark Ignition Engines, Numerical Simulations - Examples and Applications in Computational Fluid Dynamics, Prof. Lutz Angermann (Ed.), ISBN: 978-953-307-153-4, InTech, Available from: <http://www.intechopen.com/books/numerical-simulations-examples-and-applications-in-computational-fluid-dynamics/numerical-simulation-of-internal-combustion-engines>

**INTECH**  
open science | open minds

**InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

**InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen