

Rapid optimization of blast wave mitigation strategies using Quiet Direct Simulation and Genetic Algorithm

Matthew R. Smith^a, Fang-An Kuo^a, Chih-Wei Hsieh^a, Jen-Perng Yu^b, Jong-Shinn Wu^{c,*}, Alex Ferguson^d

^a National Center for High-Performance Computing, Hsinchu, Taiwan

^b Department of Information Management, Ming Chuan University, Taipei, Taiwan

^c Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan

^d State Operations, Kellogg Brown & Root Pty. Ltd., Australia

ARTICLE INFO

Article history:

Received 8 August 2009

Received in revised form 30 October 2009

Accepted 15 February 2010

Available online 19 February 2010

Keywords:

Blast wave

Euler equation

Quiet direct simulation

Genetic algorithm

ABSTRACT

Presented is a rapid calculation tool for the optimization of blast wave related mitigation strategies. The motion of gas resulting from a blast wave (specified by the user) is solved by the Quiet Direct Simulation (QDS) method – a rapid kinetic theory-based finite volume method. The optimization routine employed is a newly developed Genetic Algorithm (GA) which is demonstrated to be similar to a Differential Evolution (DE) scheme with several modifications. In any Genetic Algorithm, individuals contain genetic information which is passed on to newly created individuals in successive generations. The results from unsteady QDS simulations are used to determine the individual's "genetic fitness" which is employed by the proposed Genetic Algorithm during the reproduction process. The combined QDS/GA algorithm is applied to various test cases and finally the optimization of a non-trivial blast wave mitigation strategy. Both QDS and the proposed GA are demonstrated to perform with minimal computational expense while accurately solving the optimization problems presented.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Recent rapid increases in urban populations, the increased frequency of dense groups of city buildings, and increased use of high powered explosives by terrorists, mean that the risk and consequence of blast waves in urban regions are now more significant than ever before. Blast wave reflection, channelling through urban corridors and propagation around buildings are all situations where analytical solutions are limited in usefulness [1]. In such environments, the predicted path and nature of blast waves is impossible to predict without the aid of experimentation or computer simulation. Previously, significant efforts have been made towards the prediction of blast waves in urban environments through the use of conventional CFD [2–8]. Such simulations are often used to aid city planners in better protecting against the threat of blast waves. However, the expense associated with these simulations limits their application to hypothetical predictions under ideal conditions. Very few numerical routines are computationally efficient enough to be applied in a general predictive manner. The time required for mesh generation, user input, general flow solution and post-processing means that even a simple analysis performed using CFD can easily take hours on a single CPU. Supercomputing facilities have become the expensive fallback when blast flow sim-

ulations of a realistic size are needed; however their use often remains unfeasible in practice.

A common focus for blast wave (CFD) simulations within city environments has been its application to studies in city planning, building design and materials safety [1,4]. In these instances, a large number of unknown factors are assumed by a designer and optimal design is obtained through the development of parametric studies and their outcomes [9,10]. In such studies, the outcome of each simulation is measured through the comparison of desired quantities (such as pressure or impulse) resulting from simulations with different parameters. Such parametric studies, which can only investigate a relatively small number of combinations of parameters, can lead to misleading conclusions and inappropriate designs for blast wave mitigation.

Due to the large number of variables in such parametric studies, the use of numerical optimization algorithms is very appealing [11–14]. A common choice is the application of Genetic Algorithms (GAs) to determine ideal configurations for any desired measure of fitness. In a Genetic Algorithm, the various parameters factored into the optimal design (such as structure or support placement, size, etc.) are referred to chromosomes. Each different combination of these chromosomes results in a unique genetic genome. Through the careful selection of superior genetic properties and reproduction, the general level of fitness (i.e. measure of quality, related to the desired outcome) is seen to generally increase from generation to generation. The primary drawback to such algorithms can

* Corresponding author.

E-mail address: chongsin@faculty.nctu.edu.tw (J.-S. Wu).

be the large number of generations required before the genome holding the ideal/optimal genetic information is determined.

In the instance of CFD simulations of blast waves through an urban environment (or otherwise), the measure of fitness (or quality) of each genetic code is determined by applying the complete steady/unsteady flow development and solution for each unique genetic code (or set of structure design parameters) [11]. Thus, the application of conventional CFD simulations to Genetic Algorithms is typically limited by their computational expense [11]. In addition, the unique nature of each genetic code means that each simulation may require its own unique computational grid and accordingly possess varying degrees of accuracy. Therefore, application of Genetic Algorithms to optimization where CFD is involved is restricted to the use of computationally efficient and accurate gas-flow solvers. Alternatives to Genetic Algorithms are conventional gradient based optimization methods, which have been applied to CFD related problems [13,14].

The Quiet Direct Simulation (QDS) method for rapid simulation of ideal gas flows was first introduced as the Quiet Direct Simulation Monte Carlo (QDSMC) by Albright et al. [15,16] and later extended to higher-order accuracy and renamed as QDS by Smith et al. [5]. The QDS algorithm is based on flux calculations from a numerical approximation of the Maxwell-Boltzmann equilibrium particle velocity distribution function. QDS fluxes are true directional; it is possible, in any given time step, for fluxes of mass, momentum and energy to travel from any source region to a destination region, regardless of whether an adjacent interface exists [5,17]. This feature has been shown to increase the accuracy of flows unaligned with the computational grid [17]. This accuracy and speed of QDS flux calculation makes it ideal for combination with a Genetic Algorithm for high speed optimization solutions.

2. Quiet Direct Simulation (QDS)

2.1. Introduction to QDS

The QDS algorithm was originally introduced by Albright et al. [15,16] as QDSMC, or Quiet Direct Simulation Monte Carlo. Following this, the QDS algorithm and its extension to second-order, multi-dimensional flux-based schemes were developed by Smith et al. [5] and Cave et al. The complete details are covered in [5,15,16] and it is briefly described in the following. The normal random variable $N(0, 1)$ is defined by the probability density:

$$p(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}. \quad (1)$$

By way of Gaussian quadrature approximation, the integral of Eq. (1) over its limits can be approximated by:

$$\int_{-\infty}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} f(x) dx \approx \sum_{j=1}^N w_j f(q_j), \quad (2)$$

where w_j and q_j are the weights and abscissas of the Gaussian quadrature (also known as the Gauss-Hermite parameters) and N is the number of terms. The abscissas are the roots of the Hermite polynomials which can be defined by the recurrence equation:

$$H_{n+1}(q) = 2qH_n - 2nH_{n-1}, \quad (3)$$

where $H_{-1} = 0$ and $H_0 = 1$. The weights can be determined from:

$$w_j = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(q_i)]^2}. \quad (4)$$

Eq. (2) becomes exact when the function $f(x)$ is a linear combination of the $2N - 1$ polynomials $x^0, x^1, \dots, x^{2N-1}$. Thus, in QDS

the Maxwellian distribution of velocities can be approximated with a number of velocity “buckets” in each spatial dimension having velocities determined of the abscissas of the quadrature, and the fraction of cell mass carried in each bucket determined by the weights.

2.2. QDS flux calculation

Using the weights and abscissas derived in Section 2.1, the total flux of mass, momentum and energy from a one-dimensional source into a one-dimensional destination is given by the sum of individual flux contributions from each “particle”:

$$F_{MASS} = \sum_{J=1}^N F_{MASS}^J, \quad F_{MOM} = \sum_{J=1}^N F_{MOM}^J, \quad (5)$$

$$F_{ENG} = \sum_{J=1}^N F_{ENG}^J,$$

where F_{MASS}^J , F_{MOM}^J and F_{ENG}^J are the individual mass, momentum and energy fluxes respectively from particle J . Each of the individual contributions (with first-order spatial accuracy) can be described by the expressions:

$$F_{MASS}^J = \frac{v_J \Delta t}{\Delta x} m_J, \quad F_{MOM}^J = \frac{v_J \Delta t}{\Delta x} m_J v_J,$$

$$F_{ENG}^J = \frac{v_J \Delta t}{\Delta x} m_J \left[\frac{1}{2} v_J^2 + \varepsilon_J \right], \quad (6)$$

where the particle mass m_J , particle velocity v_J and particle internal energy ε_J are:

$$m_J = \frac{\rho \Delta x w_J}{\sqrt{\pi}}, \quad v_J = u + \sqrt{2} \sigma q_J, \quad \varepsilon_J = \frac{(\xi - 1) \sigma^2}{2} \quad (7)$$

and ρ is the density, u is the bulk (or mean) flow velocity and $\sigma = (RT)^{1/2}$ in a given source cell. The total number of degrees of freedom ξ is defined as $\xi = 2(\gamma - 1)^{-1}$. Thus energy which should be transported by unsimulated translational degrees of freedom (in this case, the y - and z -directions) is accounted for as particular internal energy. The motion of fluxes of each particle J is demonstrated in Fig. 1. The flux F_{MASS} given by Eq. (5) is the total flux of mass from the source region to the destination region. This is analogous to the True Direction Equilibrium Flux Method (TDEFM) flux [17], which represents the analytical solution to the QDS flux as the number of terms (N) approaches infinity [5].

2.3. Extension of QDS to higher spatial accuracy

To assist in removal of excessive numerical dissipation the original QDS algorithm was modified to include an extension to higher spatial accuracy. This is performed using the spatial reconstruction scheme proposed by Smith et al. [5]. The full details of the implementation can be found in [5]; the implementation is only discussed briefly here. The heart of the technique involves the reconstruction of primitive variables over each source cell assuming a linear variation, i.e. $u(x) = u_{AVG} + x(du/dx)_{EFF}$ where subscript AVG is the average value of $u(x)$ taken over the cell width, x is the distance from the cell center, and $(du/dx)_{EFF}$ is an effective gradient of quantity u calculated using a slope limiting function. In this study, the selected limiter is the Monotonized Central Difference (MC) limiter [24]:

$$\left(\frac{dQ}{dx} \right)_{EFF} = \text{MINMOD} \left[\left(\frac{dQ}{dx} \right)_C, \right. \\ \left. \alpha \cdot \text{MINMOD} \left[\left(\frac{dQ}{dx} \right)_F, \left(\frac{dQ}{dx} \right)_B \right] \right], \quad (8)$$

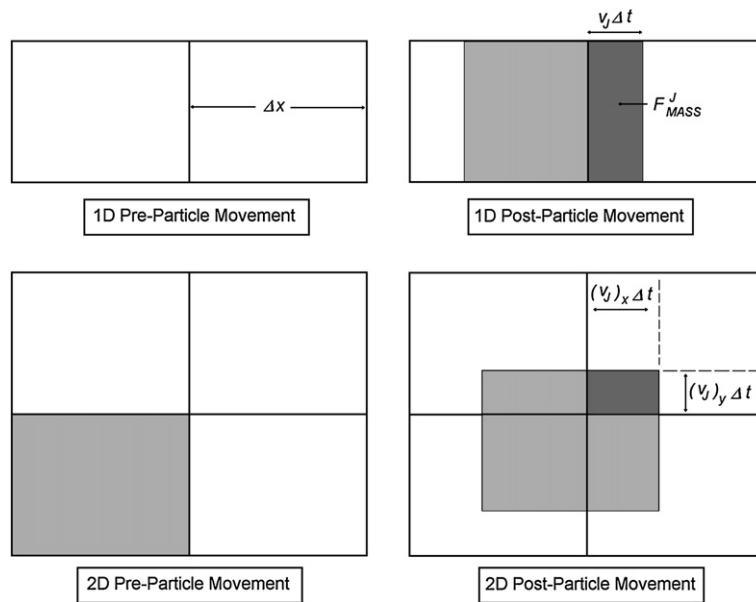


Fig. 1. One- (1) and two- (2) dimensional first-order accurate, true direction flux transfer for a QDS gas given a particular QDS particle and velocity. In a given time step, gas from a source cell is able to flux to any neighbor, regardless of whether or not the respective cells share an interface [17,5]. Fluxes are carried by particles travelling with velocities $(V_J)_x$ and $(V_J)_y$, where J represents the current QDS particle ($J = 0, 1, 2, \dots, N$) and N is the number of terms. In spatially first-order accurate, two-dimensional flux calculations the fraction of mass from each particle to successfully reach a given destination region is a ratio of the area $(V_J)_x(v_J)_y\Delta t^2/\Delta x\Delta y$. Details on extension to second-order spatial accuracy can be found in [15,18].

where the subscripts *EFF* indicate an effective gradient, while *C*, *F* and *B* indicate calculation of the slope of *Q* using central, forward and backward finite differences respectively. The parameter α is used to control numerical dissipation and has values between ($2 > \alpha > 0$) [24]. Since QDS and all related kinetic theory based numerical schemes (such as TDFEM and EFM) possess a large degree of numerical dissipation [18], the value is set to the maximum value of two (2).

3. Genetic Algorithm implementation

A large number of Genetic Algorithm (GA) methods have been developed [12]. These algorithms are part of a larger family known as Evolutionary Algorithms (EA) which also includes Genetic Programming and Evolutionary Strategies [19]. Included within the family of Evolutionary Algorithms are sub-families of Differential Evolution (DE) algorithms. There are many minor differences between various methods: however, the general approach is universal and is traditionally described in four steps:

1. **Initialization** – Create a population with desired (often randomly assigned) characteristics. If required, one may *seed* this population with a known result, which may increase the rate of convergence towards an optimal solution.
2. **Evaluation** – Regardless of approach, at some stage the fitness of each member of the population is evaluated. This indicator is typically defined by the user and is dependent on the nature of the optimization problem. Fitness may be described by a single variable or characteristic of each member of the population, or may depend upon multiple characteristics.
3. **Selection and reproduction:**
 - a. **Selection** – The population of later generations is controlled through the selection of parents and the reproductive algorithm applied. Parents are often randomly selected from members of the previous generation to ensure sufficient “genetic diversity” – selection of only the fittest members from each generation may lead to an incorrect evolutionary

state (i.e. convergence towards a local minimum). A larger diversity helps ensure the predicted optimal state is the globally optimal state, though typically increases the number of generations required to obtain a solution.

- b. **Reproduction** – The characteristics (genomes) of the selected parents are used to define the genomes of their offspring. The offspring’s genomes are calculated in two steps: (i) through a genetic crossover, where characteristics of both parents are combined in some fashion, and (ii) through mutation, where the child’s genes are modified in some manner (often a random process) to ensure that sufficient diversity exists that the solver does not become trapped in a local minima.
4. **Termination** – After a large number of generations either (i) the average measure of fitness meets a user defined level, or (ii) the variance of genomes is very low and the simulation has reached an evolutionary “dead end”. In these cases the generation cycle is ceased and the results are examined.

In the cases where information is required as a real, continuous value, the families of Differential Evolution (DE) solvers [20] are very useful. This family of solvers has been shown to be effective on a large range of optimization problems and has been shown to be more accurate than many other GAs, simulated annealing and controlled random search algorithms [20–22]. DE based solvers have also been applied to optimization problems involved CFD [11].

3.1. Conventional DE algorithm

The reproduction phase (combination and mutation) employed by a DE algorithm is different to the conventional crossover method applied to binary genetic forms used in existing approaches [19–21]. In a typical DE algorithm, a trial child *v*, intended to replace a child *c* of the previous population, is created using genetic information *X* from three separate parents (P_1 , P_2 and P_3) such that:

$$X_v = X_{P_1,g} + F(X_{P_2,g} - X_{P_3,g}), \quad (9)$$

where P_1 , P_2 and P_3 are unique ($P_1 \neq P_2 \neq P_3 \neq c$) and randomly selected from the population of generation g . The mutation factor F is used to increase genetic diversity [19] (and has values between 0 and 2) while the difference term ($X_{P_2,g} - X_{P_3,g}$) is proportional to the two-term standard deviation as seen by child v . Following this, a Monte Carlo based rejection–acceptance procedure is used to modify X_v :

$$X_v = \begin{cases} X_v, & R_f \leq CR, \\ X_c, & R_f > CR, \end{cases} \quad (10)$$

where CR is the probability of acceptance (varying from 0 to 1). Finally, if the measured fitness of v is found to be an improvement over the previous member of population it will replace, the new child is assigned the trial value X_v :

$$X_{c,g+1} = \begin{cases} X_v, & f(X_v) > f(X_{c,g}), \\ X_{c,g}, & f(X_v) \leq f(X_{c,g}). \end{cases} \quad (11)$$

This conventional DE algorithm has the advantage that the population's average fitness level from generation to generation must always increase. The mutation factor F and acceptance probability CR affect the speed and stability of the numerical scheme. Care must be taken to ensure a balance between the rate of convergence on the optimal solution and the scheme stability.

3.2. Proposed DE algorithm

The proposed algorithm aims to modify the original DE algorithm based on the assumption of normal (i.e. Gaussian) distribution of genetic properties in the population. This concept is modelled after the simulation particle creation process in a direct molecular simulation under equilibrium conditions (such as that employed by EPSM [18]). To demonstrate: particles in equilibrium are assigned velocities $V = U + R_N \sigma^{1/2}$ where V is the vector of molecular velocities, U is a vector of mean velocities, σ is the variance of the sampled population and R_N is a normally distributed random number with a mean of 0 and variance of 1.

The proposed algorithm aims to build on the traditional DE algorithm with this concept in mind. Instead of molecular velocities, genetic traits X are assigned to children based on some mean and variance of genetic traits from parents P . The proposed algorithm is outlined below:

1. Three parents are selected from the population (P_1 , P_2 and P_3). These three parents are unique ($P_1 \neq P_2 \neq P_3$), but only P_2 and P_3 are selected randomly. The parent P_1 is always selected as the fittest member of generation g .
2. The sample variance S^2 is estimated from Parents 2 and 3 as $S^2 = (X_{P_2,g} - X_{P_3,g})^2$. The employed genetic variance can be adjusted using σ such that $S^2 = \sigma^2(X_{P_2,g} - X_{P_3,g})^2$. Only two parents are selected for computational efficiency and must be randomly selected (i.e. cannot involve parent P_1) to, on average, obtain an accurate estimate.
3. To ensure bias towards the stronger parent, the mean genetic trait is estimated as:

$$\bar{X}_g = G_f X_{P_1,g} + (1 - G_f) X_{P_2,g}, \quad (12)$$

where G_f is a biased genetic fraction $G_f = F \cdot \text{MAX}[R_f, 1 - R_f]$. Here, R_f is a uniformly distributed random number and F is a constant used to further control the amount of bias given towards the fitter parent.

4. Using this information, the genetic properties of a new child $X_{c,g+1}$ are given by:

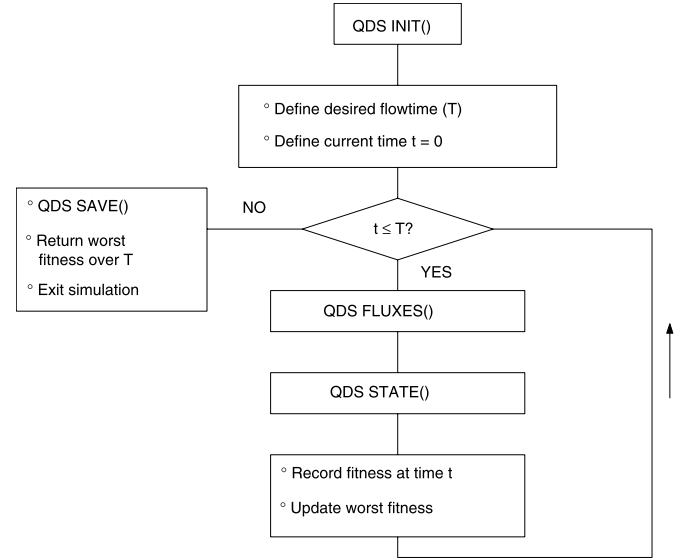


Fig. 2. Flowchart showing the general QDS unsteady flow solution algorithm. Boundary definition, initialization and mesh generation relevant to the conditions provided (contained within the genetic information of the governing child) are applied within the INIT() function. The flow solution is then advanced through time until a desired flow time T is reached. A typical true direction, finite volume approach is employed [16,15]. Fluxes are calculated from each cell to their respective neighbors in the FLUXES() function. The fluxes are later distributed and the conserved quantity vector updated within the STATE() function. For evaluation of fitness within the GA algorithm, the unsteady fitness parameter is recorded at each time interval dt . The worse level of fitness over the duration of time T is returned to the GA algorithm for use.

$$\begin{aligned} X_{c,g+1} &= \bar{X}_g + R_N S \\ &= G_f X_{P_1,g} + (1 - G_f) X_{P_2,g} + R_N \sigma (X_{P_2,g} - X_{P_3,g}). \end{aligned} \quad (13)$$

5. Following the traditional DE scheme, the genetic properties of the new child $X_{c,g+1}$ are only accepted if the new genetic properties result in a “fitter” configuration:

$$X_{c,g+1} = \begin{cases} X_{c,g+1}, & f(X_{c,g+1}) > f(X_{c,g}), \\ X_{c,g}, & f(X_{c,g+1}) \leq f(X_{c,g}). \end{cases} \quad (14)$$

3.2.1. Initialization

During initialization, the characteristics of each child c ($X_{c,0}$) are randomly selected over the permitted range of values. It is important that there is enough genetic diversity in the initial population to ensure sufficient scope to capture to globally optimized solution. The process of seeding, wherein one or more of the children are given characteristics known to be close to ideal, is not investigated here.

3.2.2. Evaluation

The evaluation of a child's fitness involves the QDS simulation of unsteady flow over the specified barrier geometry. The general procedure for an unsteady QDS simulation is presented in Fig. 2 with the integration of QDS and the proposed GA algorithm shown in Fig. 3. The fitness of each child c is measured by the maximum recorded pressure over the duration of the blast at either (i) a single location, or (ii) averaged over a series of specified locations. The unsteady flow simulation is ceased after allowing sufficient time for the pressure at the given region to have decreased well past maximum. The worse measure of fitness recorded over flow time F is returned to the GA algorithm for use in the selection and reproduction phases.

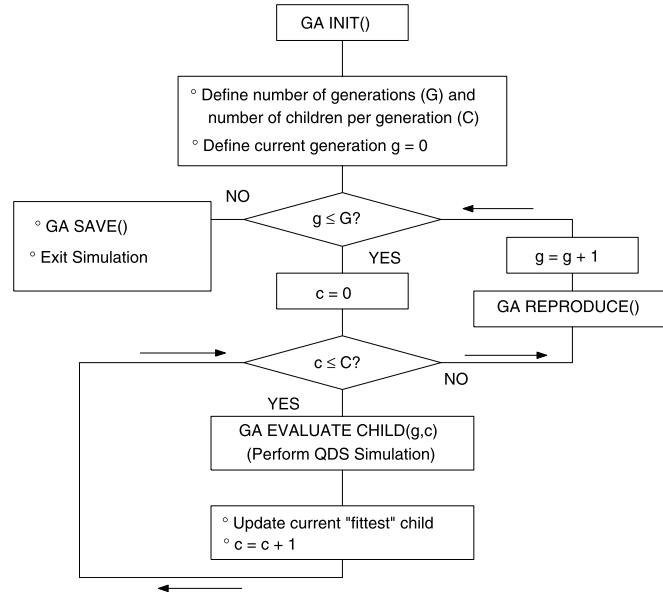


Fig. 3. Flowchart showing the general GA procedure. After initialization (within the GA_INIT function), each child c within generation g is tested for fitness within the GA_EVALUATE_CHILD function via an unsteady QDS flow simulation. The fittest member of generation g is used together with two other randomly selected and unique members of generation g to create the new children of each following generation $g + 1$ (GA_REPRODUCE).

3.2.3. Selection and reproduction

In the proposed scheme, there is always a slight bias towards the fitter parent P_1 . The variable F further controls the bias towards the fitter parent and performs a similar function to the variable F employed in existing traditional DE schemes. The variable σ is used to artificially create additional genetic diversity (than the amount sampled) through mutation and plays a similar role to that of CR employed in the traditional DE scheme. Increasing F while decreasing σ will lead to solutions with more focus on the fittest member of each generation, leading to rapid conversion at the potential cost of robustness (i.e. locating false maxima/minima in the solution range). Decreasing F and increasing σ leads to a more diverse population created during reproduction, slowing convergence towards the optimal solution while increasing robustness. Using this procedure, it can be shown that the average level of fitness from generation to generation increases towards an optimal level of fitness in a manner faster, on average, than the demonstrated DE algorithm.

4. Results and discussion

4.1. Genetic Algorithm optimization – Ackley's function

Ackley's function is a continuous, multimodal test function obtained by modulating an exponential function with a cosine wave of moderate amplitude [22,23]. The minimization of Ackley's function is a common test of robustness due to the large number of local minima [19,22]. The function in its two-dimensional form is:

$$f(x, y) = 20 + \exp(1) - 20 \exp\left(-0.2 \sqrt{\frac{x^2 + y^2}{2}}\right) - \exp\left(\frac{1}{2}(\cos(2\pi x) + \cos(2\pi y))\right). \quad (15)$$

A surface plot and contour of Ackley's function between $[(-5 < x < 5), (-5 < y < 5)]$ are shown in Fig. 4. The convergence towards the optimal solution, as calculated using both conventional DE algorithms [22] and proposed algorithms is also shown compared in Fig. 5. To aid in the reduction of statistical scatter re-

sulting from the various stochastic processes involved, the mean best fitness is shown over 1000 unique simulations with each using 20 children per generation over 50 generations. The parameters F , CR and σ can be adjusted to obtain various outcomes for both solvers. In general, the speed of initial convergence often comes at the cost of robustness. For any given value of F , CR and σ there is an associated convergence limit where the solver is unable to further refine the solution – this can be seen in Fig. 5 in both the proposed and conventional DE algorithms. For example, when using the proposed algorithm with $F = 1.5$ and $\sigma = 1.0$, there is insufficient mutation in later generations for the solver to move away from a certain point. However, with additional genetic variation ($\sigma = 1.5$), the solver is able to continue its convergence (at least over the range of generations shown) at the sacrifice of the initial rate of convergence. For this problem, optimal performance for the proposed algorithm is given when $F = 1.0$ and $\sigma = 1.0$ which provides results with faster convergence and accuracy after 50 generations than any conventional DE algorithm with parameter values in the range $(0.25 \leq F \leq 2.0)$ and $(0.2 \leq CR \leq 1.0)$.

4.2. Genetic Algorithm optimization – Peak's function

Another function employed as a benchmark test is the Peak's function from MATLAB [25], which is:

$$f(x, y) = 3(1 - x)^2 \exp(-x^2 - (y + 1)^2) - 10\left(\frac{x}{5} - x^3 - y^5\right) \exp(-x^2 - y^2) + \frac{1}{3} \exp(-(x + 1)^2 - y^2). \quad (16)$$

This function is also shown in Fig. 4 and is used here to demonstrate the proposed DE algorithm's flexibility as a general optimization tool. Over the range $(-3 < x < 3)$, $(-3 < y < 3)$ the global minima is at $(x, y) \approx (0.23, -1.62)$ ($f(x, y) \approx -6.551$) with another local minima at $(x, y) \approx (-1.36, 0.2)$ ($f(x, y) \approx -3.05$). The difference between mean best fitness and global minima is shown in Fig. 5 for various parameters in both the proposed and

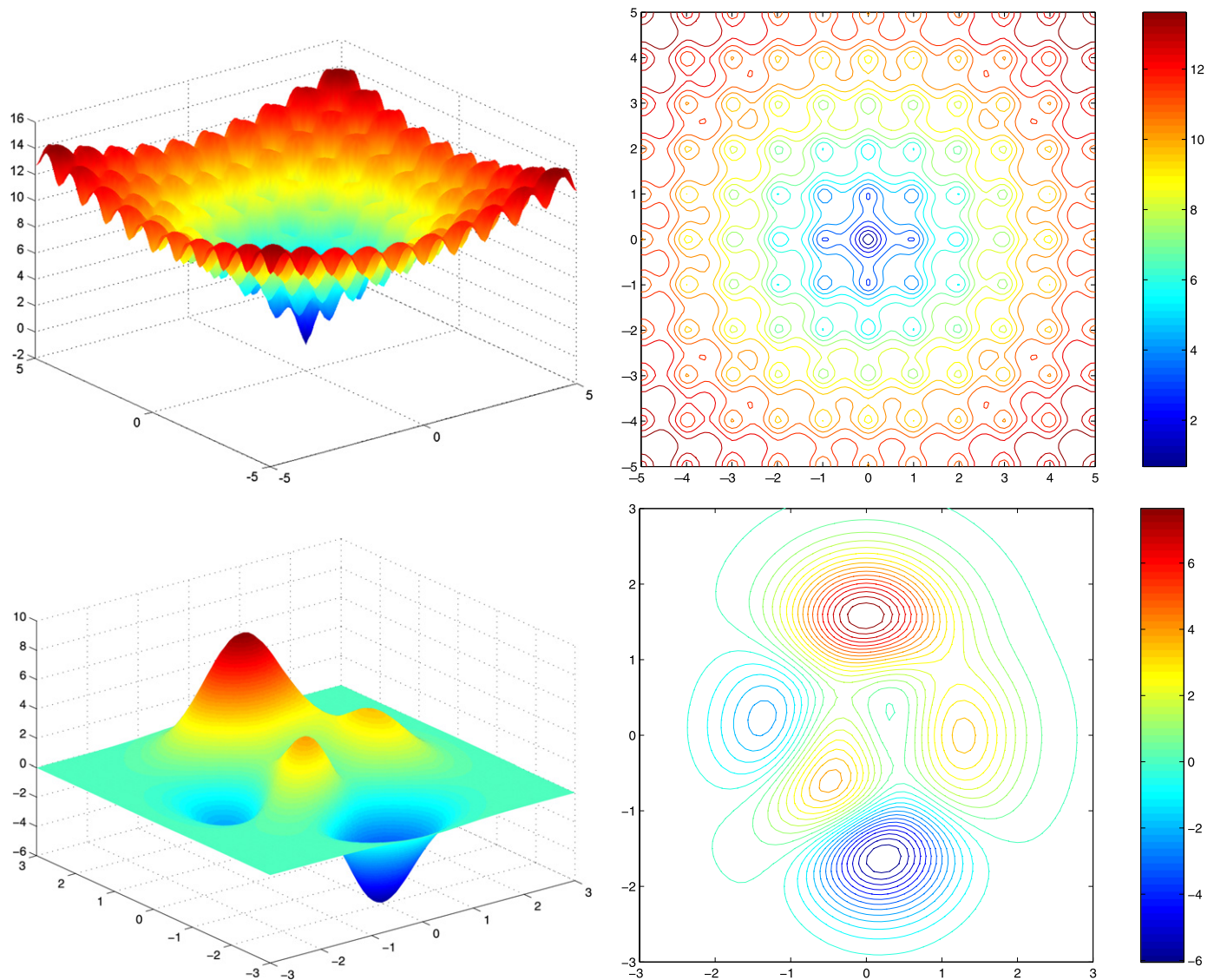


Fig. 4. (Top) Ackley's function (in two-dimensional form) plotted over range $([-5 < x < 5], [-5 < Y < 5])$. The global minima exists at $(x^*, y^*) = (0, 0)$ with $f(x^*, y^*) = 0$. The large number of local minima over the specified range makes the function useful for testing the robustness of optimization schemes [23]. (Bottom) MATLAB's Peak function [25] plotted over range $([-3 < x < 3], [-3 < Y < 3])$. The global minima exists at $(x^*, y^*) \approx (0.23, -1.62)$ with $f(x^*, y^*) \approx -6.551$.

conventional DE algorithms. The same trade-off between initial rate of convergence and robustness is demonstrated for various values of F , CR and σ . Insufficiently large values of σ result in an increasing fraction of solutions becoming trapped in the local minima at $(x, y) \approx (-1.36, 0.2)$. The same trend is (generally) presented in the conventional DE solver. Over 50 generations, the proposed algorithm provides the best result when $F = 1.0$ and $\sigma = 2.0$ and can be shown to provide better performance when compared to traditional DE algorithms with parameter values in the range $(0.25 \leq F \leq 2.0)$ and $(0.2 \leq CR \leq 1.0)$.

4.3. Simulation and optimization verification – barrier design

In this study, blast wave mitigation is investigated through (i) barrier design and (ii) barrier placement under various restrictions. Here we investigate two barrier designs, each with a different number and type of variable.

4.3.1. Barrier Design A

Barrier Design A holds four (4) geometric configuration variables. This is demonstrated in Fig. 6. These variables make up the

genetic characteristics X_1 – X_4 of each child and are (i) the primary arm length L_1 , (ii) the secondary arm length L_2 , (iii) the primary arm angle α_1 , and (iv) the secondary arm angle α_2 . The barrier stem height, L_3 , is fixed in its location and restricted by the condition that each design uses the approximately the same total length of material, or $L_1 + L_2 + L_3 = \text{constant}$.

4.3.2. Barrier Design B

Barrier Design B has two (2) variables and is a general simplification of Barrier Design A. This is also demonstrated in Fig. 6. In this instance, the variables that make up the genetic characteristics of each child are (i) the primary arm length L_1 , and (ii) the secondary arm length L_2 . The previous angles α_1 and α_2 are fixed at 0 and 180 degrees forming a T-shaped cross section. As with Barrier Design A, the height of the stem of the barrier L_3 is restricted ($L_1 + L_2 + L_3 = \text{constant}$). In both designs, blasts are modelled through the inclusion of a small, high temperature region at the location $L_5 = 0.5L$ where L is the length of the entire simulation region. The radius of the high temperature region was $0.02L$ with a temperature one hundred times greater than ambient conditions, resulting in a very large, strong blast wave

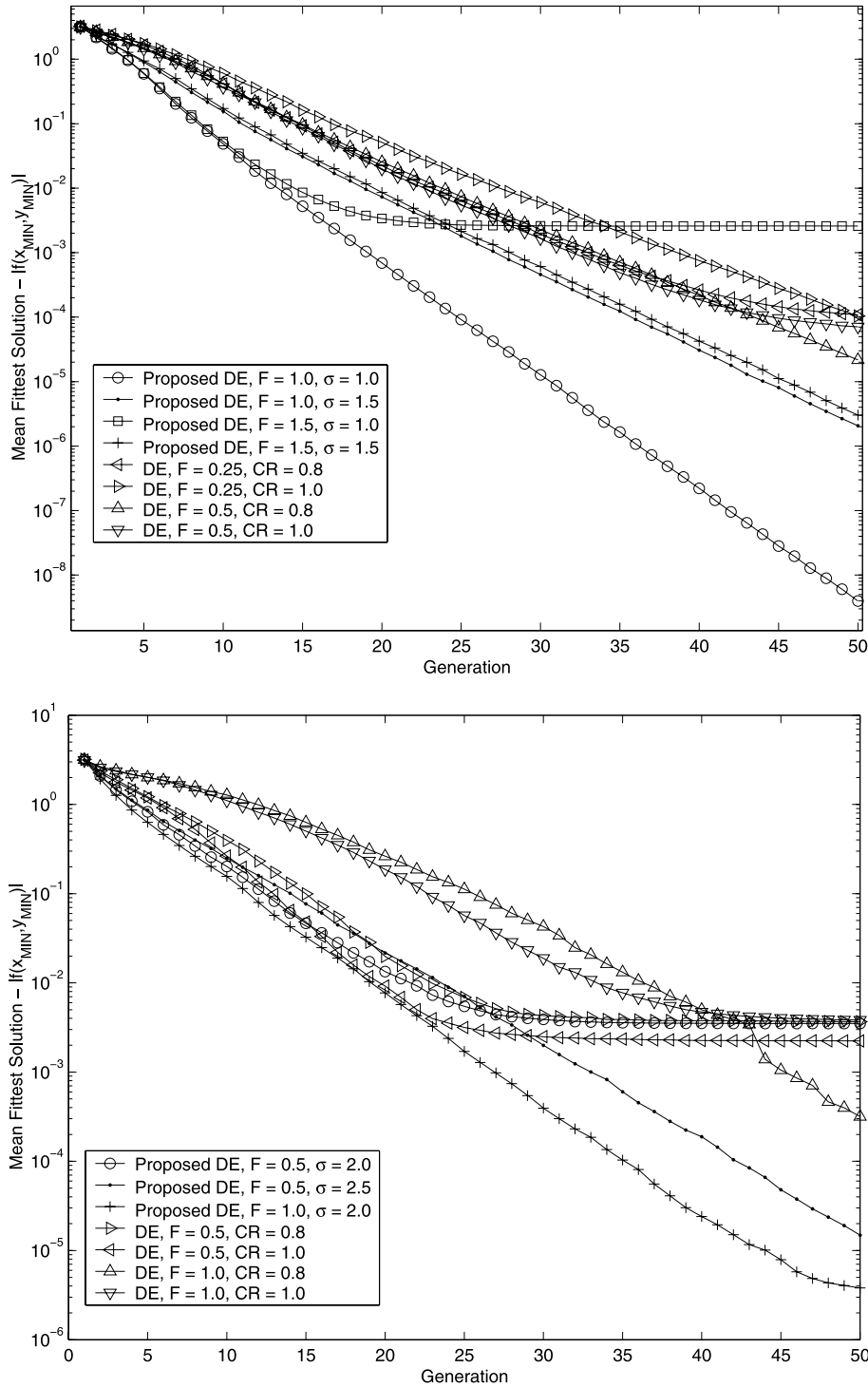


Fig. 5. Convergence towards the global minima for (top) Ackley's function, and (bottom) MATLAB's Peak function. Both the conventional DE algorithm and the proposed algorithm are presented for various values of F , CR and σ with the average taken over 1000 unique runs of 50 generations with each containing 20 children.

propagating through the surrounding regions. All simulations are two-dimensional simulations of height $H = 0.5L$. Each simulation employed a Cartesian grid with 20,000 (200 by 100) rectangular cells. The maximum length of material permitted in each case was $L_1 + L_2 + L_3 = 0.3H$. The barrier was placed at location $L_B = 0.7L$ and pressure readings taken at location $L_P = 0.9L$. In all cases, the ambient ideal inviscid gas is at rest with temperature T and a specific gas constant of R . All simulations are advanced in time up to a maximum flow time F of $F(RT)^{0.5}L^{-1} = 0.4$. The fluxes from

each cell are distributed to the nearest surrounding 8 cells only through the restriction of the kinetic CFL number, given in each cell as:

$$CFL = \text{MAX} \left[\frac{\text{MAX}(v_j^x) \Delta t}{\Delta x}, \frac{\text{MAX}(v_j^y) \Delta t}{\Delta y} \right]. \tag{17}$$

The time step is adjusted to ensure that the largest CFL number at any point in the simulation region is $CFL = 0.75$. It is emphasized that this CFL is not for stability but rather to maintain physical

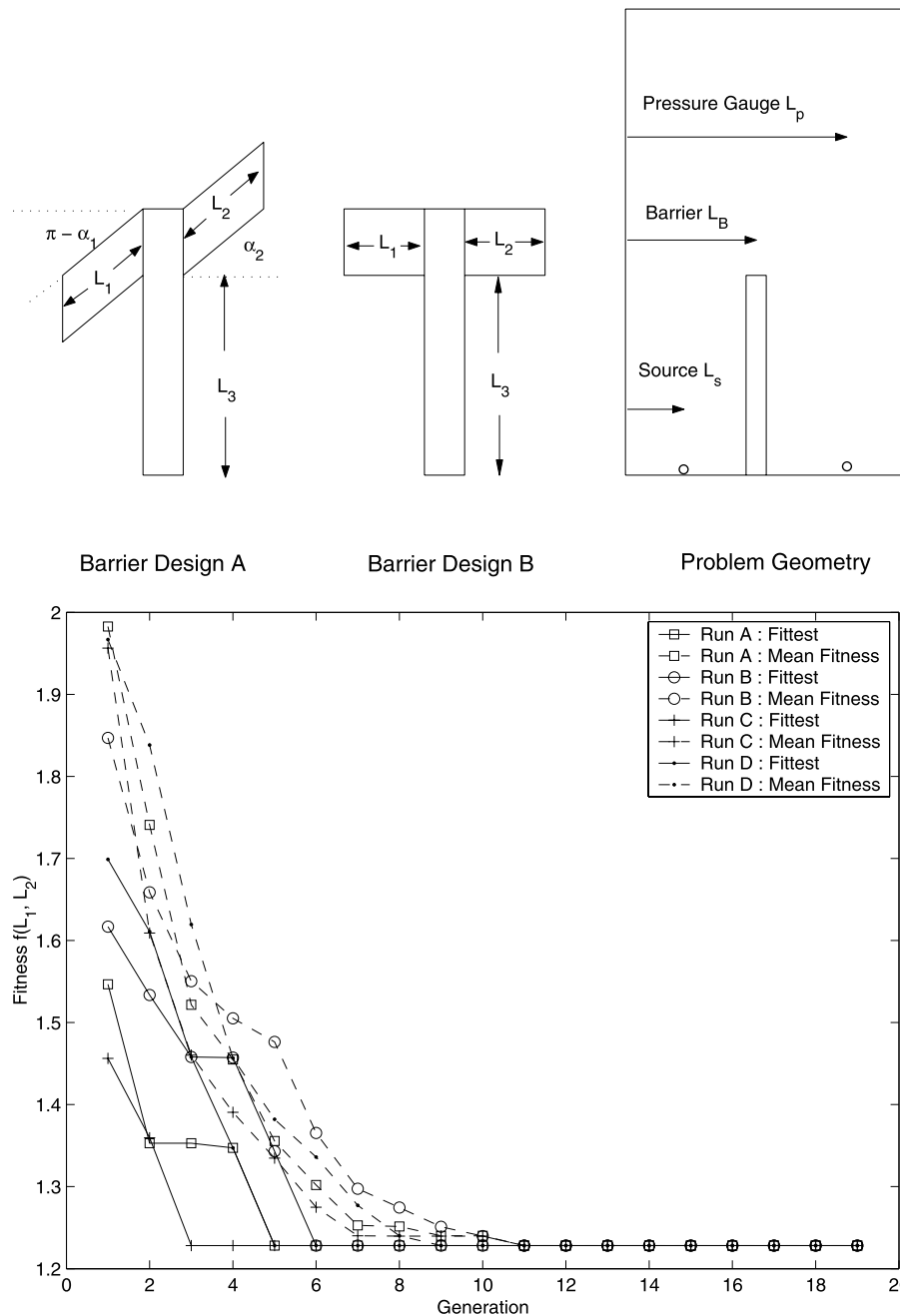


Fig. 6. Barrier Design A (top, left) and Barrier Design B (top, center) for simple optimization verification and simulation domain showing blast source, barrier and gauge placement (top, right). (Bottom) Convergence of the fittest and mean fittest solution predicted using the proposed DE algorithm with $F = 1.5$, $\sigma = 1.0$.

realism (accuracy) in the results [15,16,18]. Neumann boundary conditions are applied for the atmospheric boundaries while simple reflective conditions are applied to body surfaces.

For the purpose of demonstrating the flow solving capacity of QDS, sample unsteady flow contours for a randomly selected barrier configuration are presented in Fig. 7. As a result of the large temperature ratio across the blast source, a blast wave (shock wave) is seen to propagate outwards from the blast source in a radially symmetric manner. The interaction of the shock wave with the barrier results in a reflected shock. The reflected shock propagates back towards the source region and later recombines with the original blast wave.

The convergence towards the correct optimal wall design is shown in Fig. 6. Results are shown for several simulation runs

(runs A through D) with varying random initial conditions. Convergence is shown over 20 generations with 10 children in each using parameter values $F = 1.5$ and $\sigma = 1.0$. Pressure contours of the optimal barrier shape (i.e. a straight wall) are shown in Fig. 7 at various flow times. This result graphically confirms the convergence shown in Fig. 6. Despite being trivial in nature, the test was important in verifying the accurate nature of the combined QDS/GA method prior to a less trivial CFD optimization problem.

4.4. Blast wave mitigation optimization through barrier placement

The optimization of a blast wave mitigation strategy through optimal placement of several barriers is investigated. The genetic characteristics in this case are the locations of two walls, L_1 and

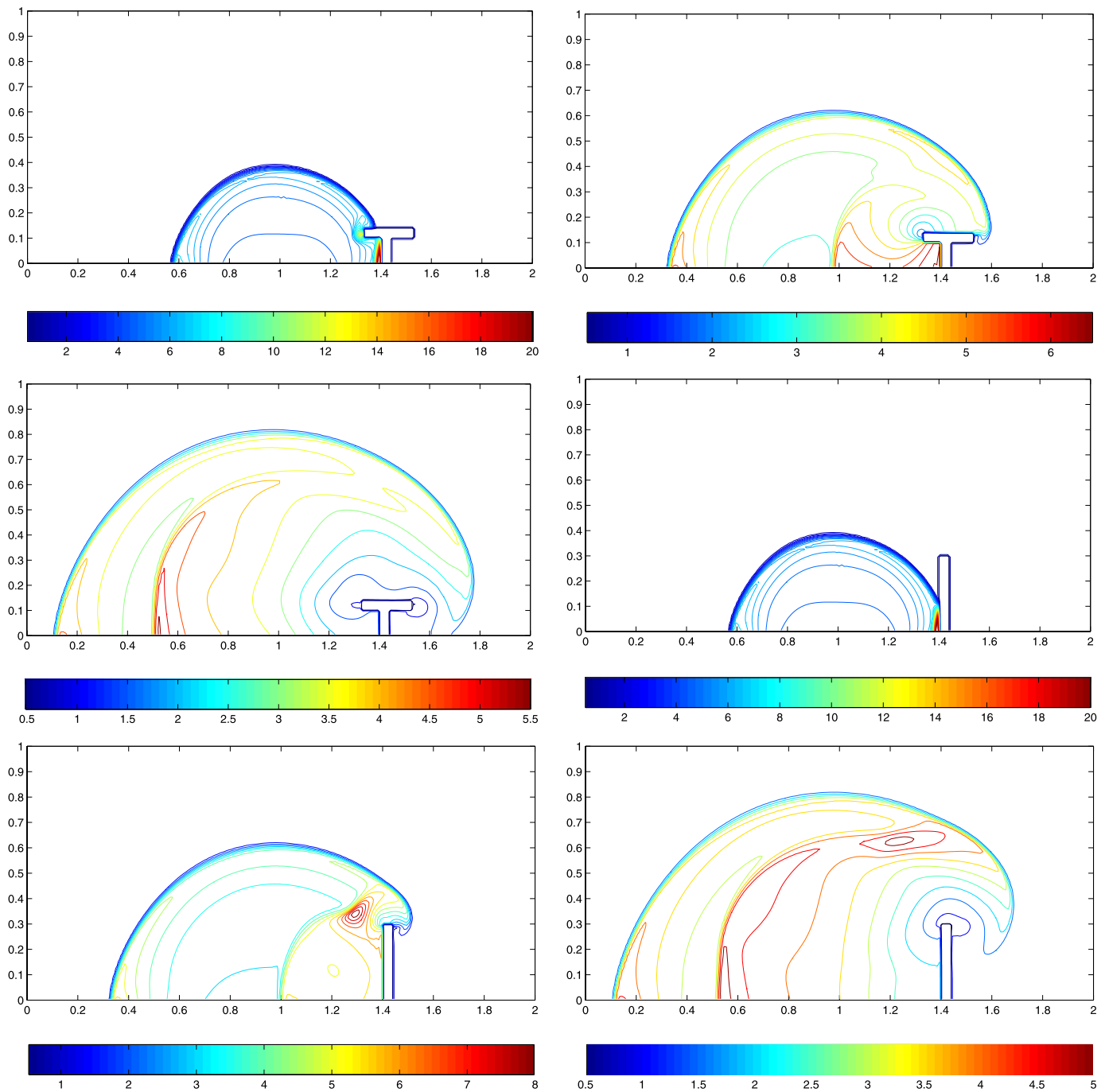


Fig. 7. (Top) Unsteady flow development over a barrier of genetic configuration ($L_1/L_T = 0.307$, $L_2/L_T = 0.225$, $L_3/L_T = 0.468$). (Bottom) Unsteady flow development for the resulting optimal design (a simple wall). Contours of pressure are shown at flow times of $f(RT)^{0.5}H^{-1} = 0.08, 0.16$ and 0.24 . Since the QDS solver is an Eulerian flow solver, a taller barrier between the source and the pressure gauge is (mathematically) equivalent to placing the pressure gauge at a location further from the source.

L_2 , as shown in Fig. 8. These variables are further governed by the rules that (i) each must fall within the limits $L_{MIN} = 0.6L$ and $L_{MAX} = 0.9L$, and (ii) they cannot geometrically interfere with each other (i.e. they must not coexist in the same region of space). Each barrier has a width of $W_{BARRIER} = 0.02L$ and height of $H_{BARRIER} = 0.3H$. The building surface is located at $L_{BLG} = 0.95L$ and the blast source is placed at $L_S = 0.5L$. The radius of the blast is identical to previous test cases ($0.02L$) with a temperature 100 times that of the surrounding region.

Determining the optimal location for blast barrier placement is guesswork when performed by human experts: the flow of blasted

gas in complex urban environments is very difficult to predict. An original barrier placement configuration was chosen arbitrarily under the assumption that (i) a barrier placed as close to the blast source as possible, and (ii) a barrier placed as close to the target structure as possible (i.e. $L_1 = 0.6L$, $L_2 = 0.9L$), would provide optimal protection. The two mobile barriers are identical, with heights $H_{BARRIER} = 0.3H$ and width $W_{BARRIER} = 0.02L$. The unsteady flow field development resulting from the initial configuration is shown in Fig. 9.

The optimal configuration, as calculated using the proposed Genetic Algorithm, placed the barrier locations at approximately

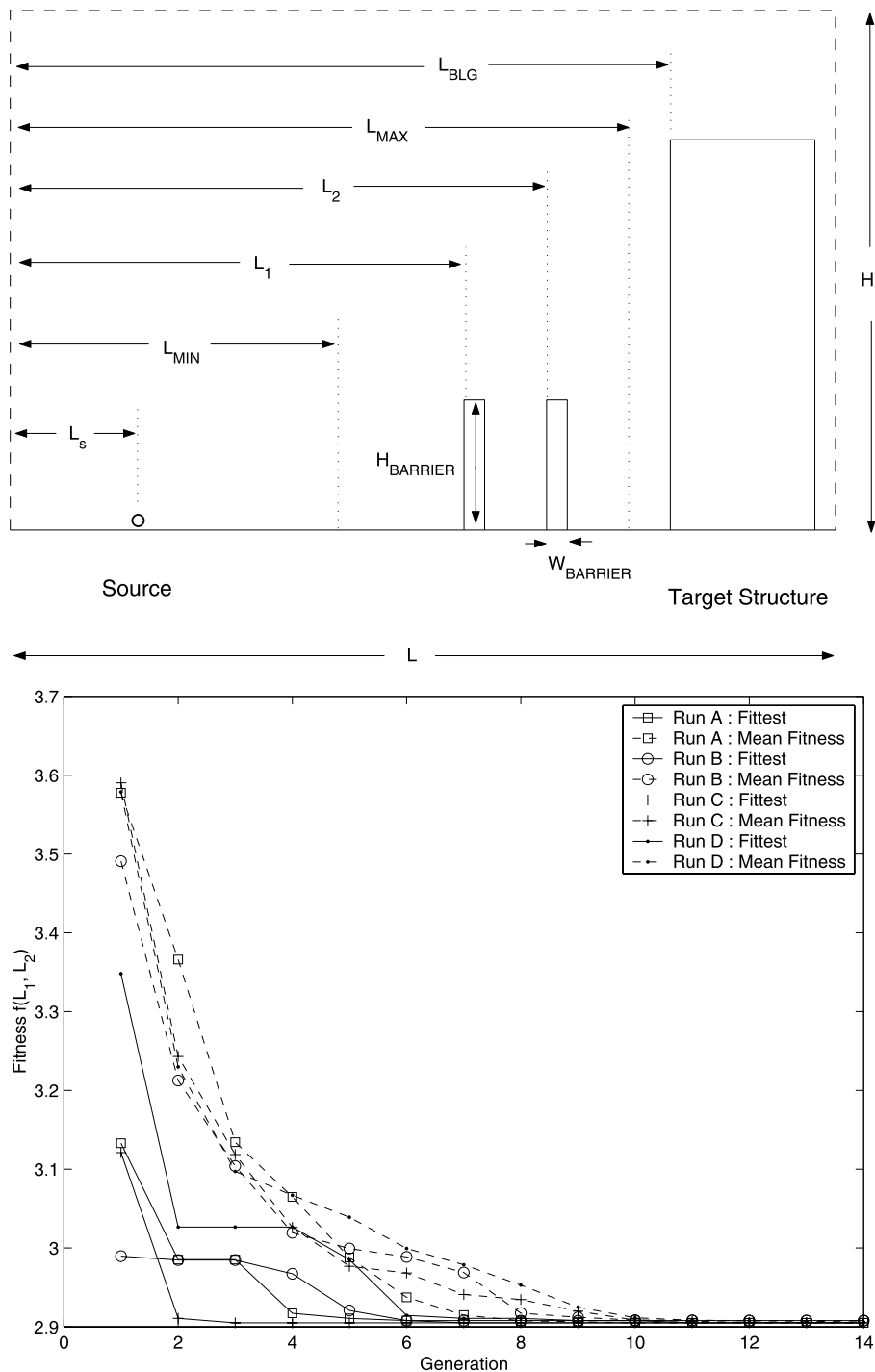


Fig. 8. Optimization for barrier location in a blast wave mitigation scenario. (Top) Computational domain for the mitigation scenario: the target structure is placed L_{SOURCE} from the blast source with barriers of height H_{BARRIER} and width W_{BARRIER} located between L_{MAX} and L_{MIN} in front of the target structure. (Bottom) Convergence of fittest and mean fitness solutions for four unique runs of the modified DE algorithm using $F = 1.5$ and $\sigma = 1.0$ with 10 children in each generation.

$L_1 = 0.69L$ and $L_2 = 0.8L$. The convergence towards the optimal result is also shown in Fig. 8 for several runs (runs A through D) using parameter values $F = 1.5$ and $\sigma = 1.0$. The pressure contours resulting from this configuration is shown in Fig. 9. The closer proximity of the barriers, combined with sufficient distance from the structure, results in lower instantaneous pressures acting on the pressure gauges P_1 – P_4 placed at $0.1H$, $0.3H$, $0.5H$ and $0.7H$ respectively. This trend is demonstrated in Fig. 10.

To confirm the correct identification of the optimal result, a parametric study was conducted. A large number (2500) of sim-

ulations with values of L_1 and L_2 ranging from $(0.75 < L_1 < 0.9)$ and $(0.65 < L_2 < 0.73)$ were conducted. For each configuration, the maximum recorded average pressure (over gauges P_1 – P_4) was recorded. The resulting data showed that the optimization routine has successfully predicted the best barrier placement configuration. The parametric study also revealed (i) the worst possible placement configuration is both barriers located in close proximity to the target structure, and (ii) configurations where a barrier is placed at $0.7L$ and the second barrier is placed anywhere in the space between results in a generally good outcome.

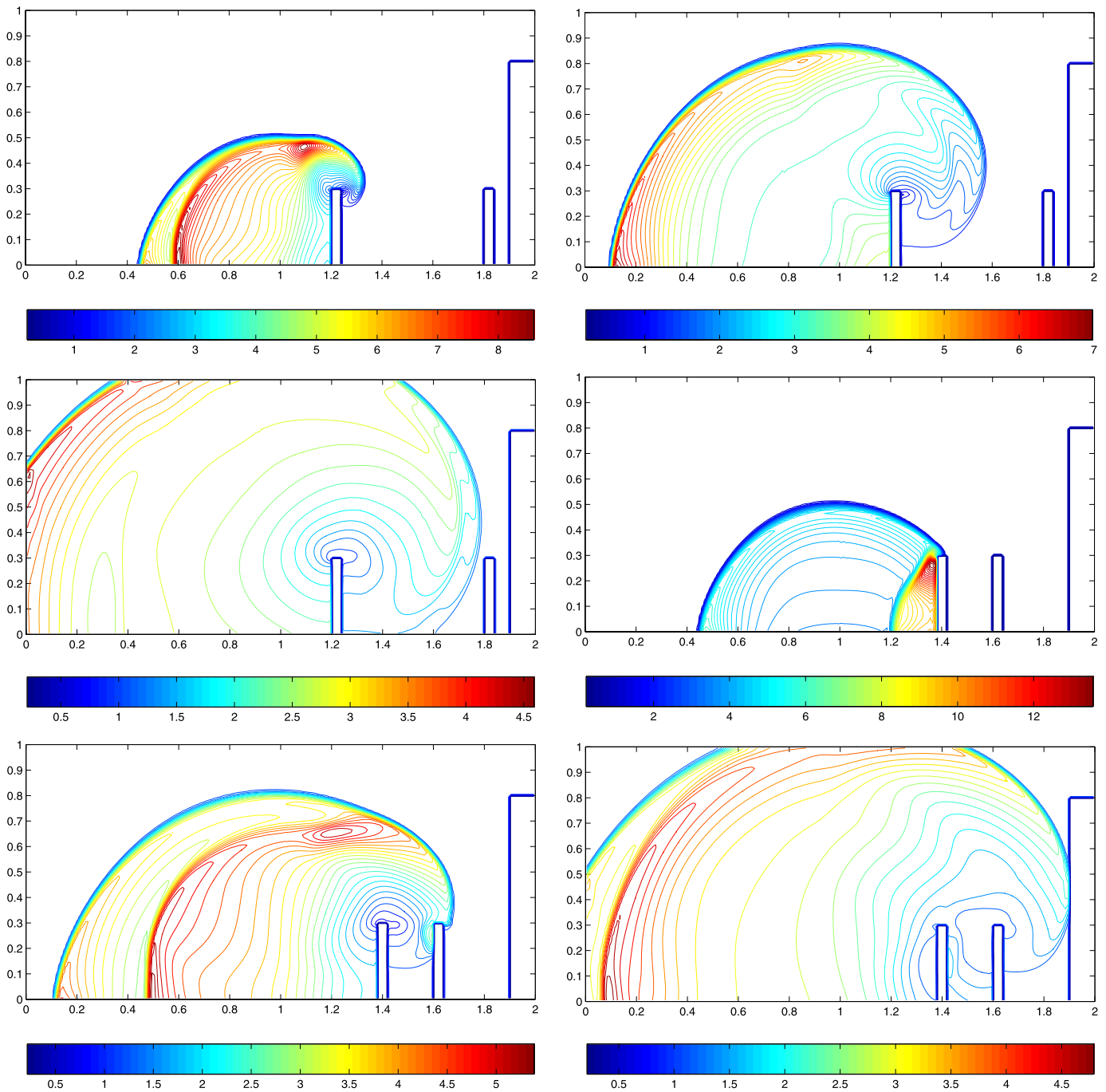


Fig. 9. Pressure contours showing unsteady flow development for the multi-barrier optimization problem at flow times of $T(RT)^{0.5}H^{-1} = 0.24, 0.36$ and 0.48 . (Left) Barrier placement configuration is $L_1 = 0.6L$ and $L_2 = 0.9L$. These placements were chosen by an assumption that (i) a barrier as close to the blast as possible, and (ii) a barrier as close to the building as possible, would provide maximum protection. (Right) Barrier placement configuration is the calculated optimal configuration of $L_1 = 0.69L$ and $L_2 = 0.8L$. The optimal configuration was based on the minimal instantaneous average pressure over the building surface measured by four (4) pressure gauges over the time frame $T(RT)^{0.5}H^{-1} = 0.6$.

5. Conclusion

Presented is a modified form of a Differential Evolution (DE) algorithm combined with the rapid Quiet Direct Simulation (QDS) flow solver. The proposed Genetic Algorithm is demonstrated in the solution of a standard optimization benchmark – minimization of Ackley's function – and then applied to various blast wave optimization scenarios. The optimal design of a barrier (with simple design parameters) under the constraint of constant material quantity is demonstrated. Finally, a blast wave mitigation optimization problem with several mobile barriers is solved with non-

trivial results. The optimal solution is confirmed through a parametric study, demonstrating the outcome of the combined QDS/GA method and providing additional information necessary for an effective barrier placement strategy.

The developed Genetic Algorithm rapidly converges towards optimal solutions while possessing sufficient mutation to prevent capture in local minima. The combined optimization/unsteady flow problems converge to an optimal result in approximately 1 hour on an ordinary single CPU (~ 3 GHz) with approximately 18 seconds required per unsteady flow simulation. Future efforts include the combination of more realistic three-dimensional blast simulations

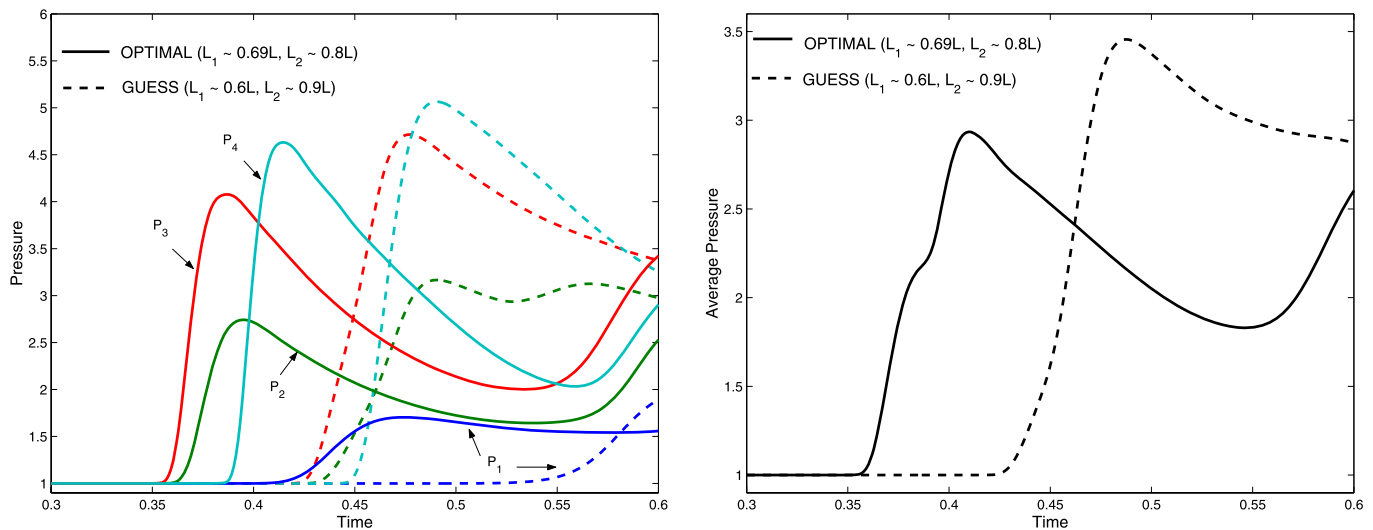


Fig. 10. (Left) Traces from pressure gauges P_1 – P_4 placed on the target structure surface at $y = 0.1H, 0.3H, 0.5H$ and $0.7H$ respectively. (Right) Trace of average pressure over the target structure. Results are shown for the optimal barrier placement configuration ($L_1 = 0.69L, L_2 = 0.8L$) and for the guessed optimal configuration ($L_1 = 0.69L, L_2 = 0.8L$).

and optimization, together with the application of Graphics Processor Units (GPU) to aid in the efficient and rapid solution of blast wave optimization problems.

Acknowledgements

Corresponding author, Prof. J.-S. Wu, would like to express his sincere thanks to the financial support of National Science Council of Taiwan through the grant No. NSC-96-2628-E-009-136-MY3.

References

- [1] R.C. Ripley, B. von Rosen, D.V. Ritzel, D.R. Whitehouse, Small-scale modeling of explosive blasts in urban scenarios, in: Proceedings of the 21st International Symposium on Ballistics, April 19–23, Adelaide, Australia, 2004.
- [2] P.D. Smith, T.A. Rose, Blast wave propagation in city streets: An overview, *Progress in Structural Engineering and Materials* 8 (1) (2006) 16–28.
- [3] C.E. Needham, Blast loads and propagation around and over a building, in: K. Hannemann, F. Seiler (Eds.), *Shock Waves: 26th International Symposium on Shock Waves, Proceedings*, vol. 2, Springer, Berlin/Heidelberg, 2007.
- [4] N.K. Birnbaum, R.A. Clegg, G.E. Fairlie, C.J. Hayhurst, N.J. Francis, Analysis of blast loads on buildings, in: ASME Pressure Vessels and Piping Conference, Structures Under Extreme Loading Conditions, 1996.
- [5] M.R. Smith, H.M. Cave, J.-S. Wu, M.C. Jermy, Y.-S. Chen, An improved Quiet Direct Simulation method for Eulerian fluids using a second-order scheme, *Journal of Computational Physics* 228 (6) (2009) 2213–2224.
- [6] T. Ngo, P. Mendis, A. Gupta, J. Ramsay, Blast loading and blast effects on structures – an overview, *Electronic Journal of Structural Engineering, Special Issue: Loading on Structures* (2007) 76–91.
- [7] J.K. Clutter, J.T. Mathis, M.W. Stahl, Modeling environmental effects in the simulation of explosion events, *International Journal of Impact Engineering* 34 (2007) 973–989.
- [8] J. Tang, CFD simulation of blast in an internal geometry using a Cartesian cell code, in: Proceedings of the 16th Australasian Fluid Mechanics Conference, December 2–7, Gold Coast, Australia, 2007.
- [9] T. Nozu, R. Tanaka, T. Ogawa, K. Hibi, Y. Sakai, Numerical simulation of hydrogen explosion tests with a barrier wall for blast mitigation, in: Proceedings of the 1st International Conference on Hydrogen Safety, September 8–10, Pisa, Italy, 2005.
- [10] A.G. Venetsanos, T. Huld, P. Adams, J.G. Bartzis, Source, dispersion and combustion modeling of an accidental release of hydrogen in an urban environment, *Journal of Hazardous Materials* 105 (2003) 1–25.
- [11] A.D. Belegundu, S.D. Rajan, Shape optimization of metal plates subject to blast pressure loading, in: Proceedings of the 2008 International Conference on Engineering Optimization, June 1–5, Rio de Janeiro, Brazil, 2008.
- [12] Z.-H. Ma, H.-Q. Chen, J. Periaux, A hybrid mesh/meshless CFD solver coupled with genetic algorithms for solving inverse design problems in aerodynamics, in: Proceedings of the 8th World Congress on Computational Mechanics, June 30–July 5, Venice, Italy, 2008.
- [13] O. Soto, R. Löhner, C. Yang, An adjoint-based design methodology for CFD problems, *International Journal for Numerical Methods in Heat and Fluid Flow* 14 (6) (2004) 734–759.
- [14] R. Löhner, O. Soto, C. Yang, An adjoint-based design methodology for CFD optimization problems, AIAA-03-0299, 2003.
- [15] B.J. Albright, W. Daughton, D.S. Lemons, D. Winske, M.E. Jones, Quiet direct simulation of plasmas, *Physics of Plasmas* 9 (2002) 1898.
- [16] B.J. Albright, D.S. Lemons, M.E. Jones, D. Winske, Quiet direct simulation of Eulerian fluids, *Physical Review E* 65 (055302) (2002) 1–4.
- [17] M.R. Smith, M.N. Macrossan, M.M. Abdel-jawad, Effects of direction decoupling in flux calculation in Euler solvers, *Journal of Computational Physics* 227 (8) (2008) 4142–4161.
- [18] D.I. Pullin, Direct simulation methods for compressible ideal gas flow, *Journal of Computational Physics* 34 (1980) 144–231.
- [19] K. Fleetwood, Introduction to differential evolution, in: Proceedings of Mathematics and Statistics of Complex Systems (MASCOS) One Day Symposium, 26th November, Brisbane, Australia, 2004. <http://www.maths.uq.edu.au/MASCOS/Multi-Agent.html> (Last visited: June 12th 2009).
- [20] R. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (1997) 341–359.
- [21] K.V. Price, An introduction to differential evolution, in: N. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, McGraw-Hill, London, 1999.
- [22] M. Gen, R.-W. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, ISBN 0471127418, 1997.
- [23] D.H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, 1987.
- [24] B. Van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, *Journal of Computational Physics* 32 (1979) 101.
- [25] The Mathworks Inc., MATLAB Product Documentation, <http://www.mathworks.com/access/helpdesk/help/techdoc/index.html> (Last viewed: October 19th, 2009).