

## THE APPLICATION OF COMPUTATIONAL FLUID DYNAMICS (CFD) SIMULATIONS FOR NASAL DRUG DELIVERY

Marit KLEVEN<sup>1</sup>, Morten C. MELAAEN<sup>1</sup>, Martin REIMERS<sup>2</sup>, Jan S. RØTNES<sup>3,4</sup>, Lars AURDAL<sup>4,5</sup>, Per G. DJUPESLAND<sup>6</sup> and Colin SHELDRAKE<sup>7</sup>

<sup>1</sup> Telemark University College (HiT-TF) and Telemark R&D Centre (Tel-Tek), 3918 Porsgrunn, NORWAY

<sup>2</sup> CMA, University of Oslo, 0316 Oslo, NORWAY

<sup>3</sup> SimSurgery AS, 0855 Oslo, NORWAY

<sup>4</sup> Interventional Centre at Rikshospitalet, 0027 Oslo, NORWAY

<sup>5</sup> Norwegian Computing Center, 0314 Oslo, NORWAY

<sup>6</sup> OptiNose AS, 0349 Oslo, NORWAY

<sup>7</sup> OptiNose UK Ltd, Wiltshire SN3 4TG, UK

### ABSTRACT

The Norwegian company OptiNose AS is developing a novel concept that challenges traditional nasal delivery systems. The patented bi-directional delivery system takes advantage of the posterior connection between the nasal passages persisting when the soft palate automatically closes during oral exhalation. The exhalation into the delivery device triggers the release of particles into an airflow, which enters one nostril via a sealing nozzle and exits through the other nostril.

During the development process for their drug delivery concept, OptiNose is using Computational Fluid Dynamics (CFD). The simulations are used to visualize and demonstrate the basic features of the bi-directional technique and discuss how its design and functionality could be further optimised. CFD computations thus increase the efficiency of device development and reduce the need for expensive and time consuming laboratory experiments and clinical tests. The simulations are carried out by use of Fluent, using a volume grid generated by Tgrid.

To perform successful CFD calculations on the nose, construction of a proper surface grid of the nasal cavity, is mandatory. A high resolution CT-scan was the first part of this process, where after the 3-dimensional surface geometry was created by a pipeline of image segmentation and geometric modification steps. The process of building the surface grid is presented in the paper.

Essential for the CFD computations is thorough validation of the simulation results. This paper presents comparisons between experimental and numerical simulations. Simple numerical models are used to verify critical parameters, e.g. airflow and particle size, in the design process, together with *in vitro* experiments in a human nasal airway model.

### NOMENCLATURE

$A$  projected area of particle

$c_i$  cost for a triangle amounted to an exponential function of its skewness  $s$   
 $C_D$  drag coefficient  
 $F_i$  source term  
 $g$  gravity  
 $m_p$  particle mass  
 $n_i$  number of vertices in geometry  
 $p$  pressure  
 $s$  skewness of a cell  
 $t$  time  
 $v$  velocity of the gas phase  
 $v_i$  vertex no.  $i$   
 $v_p$  particle velocity  
 $V_p$  volume of particle  
 $x_p$  particle position  
 $\mu$  viscosity of the gas phase  
 $\rho$  density of the gas phase  
 $\rho_p$  particle density  
 $\tau_{ij}$  the stress tensor

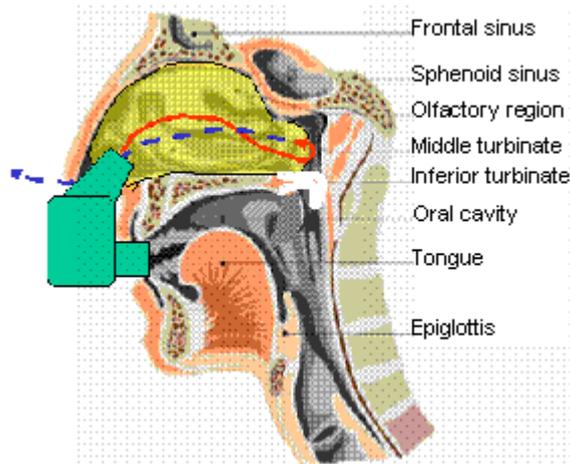
### INTRODUCTION

#### Description of the nasal geometry

The apparent external nose surrounds the nostrils and one-third of the nasal cavity, which in its entirety consists of a 5 cm high and 10 cm long chamber. The nose consists of two nasal passages, separated by the nasal septum, each of which opens independently into the upper part of the throat. The internal walls opposite the nasal septum have ridges formed of bone, i.e. turbinate bones. Several of the bones, the frontal, sphenoid, ethmoid and maxilla, contain hollow air-filled spaces, i.e. the sinuses. The olfactory mucosa is situated in the roof of the nasal cavity, with correspondence to the brain. A prerequisite for an odour sensation is that molecules can be brought in direct contact with the olfactory receptor cells in this area. Approximately 2-3 cm from the nares is the narrowest and most resistive portion of the entire airway, called the nasal valve. After passing through this narrow region, the airflow enters the nasal cavity, where the cross sectional area increases considerably.

## Background and motivation of this work

Nowadays, most drugs are being administered by the oral route or by injection, and vaccines are mainly injected, but nasal delivery is considered for an increasing number of existing and new drugs and vaccines. The easy access to the highly vascularized nasal mucosa rich in immunologically active dendritic cells and organized lymphatic tissues makes the nose especially attractive. Current nasal delivery devices, though, have major disadvantages and there is a considerable potential for improvement. The Norwegian company OptiNose AS is developing a novel concept that challenges traditional delivery systems, see Djupesland *et al.* (2004). The patented bi-directional delivery system improves drug and vaccine distribution to the nasal mucous membrane while at the same time preventing lung deposition. It takes advantage of a posterior connection between the nasal passages persisting when the velum automatically closes during oral exhalation. It is the exhalation into the delivery device that triggers the release of particles into an airflow, which enters one nostril via a sealing nozzle, passes throughout the communication posterior to the nasal septum, and exits through the other nostril. The opposite directed airflow in the exit nostril will cause particles to be deposited on the posterior surface of the nasal structures. The bi-directional delivery technique is illustrated in Figure 1.



**Figure 1:** OptiNose's bi-directional technique.

Industrial models based on Computational Fluid Dynamics (CFD) have often been used to study the flow of particles and droplets in gases. Since CFD has a wide application when analysing process equipment, it seems reasonable also to use this tool in the study of nasal aerodynamics. The literature give the impression that the establishment of a branch of science linking the two large fields of medical science and engineering science, is progressing. However, there is only one published study on CFD simulations of bi-directional nasal delivery (Kleven *et al.*, 2004). A bi-directional nasal airflow is completely separated from the oral cavity and lower airways, hence it is possible to adjust parameters such as particle size and airflow, to optimize distribution in the nasal mucosa. Kleven *et al.* (2004) describe briefly how CFD can be applied as a tool to visualize and demonstrate the basic features of the technique. It is important to demonstrate how CFD can be used to further optimize design and functionality of novel delivery devices based

on the bi-directional concept, yielding an optimized deposition pattern. Reducing the number of parameters evaluated in expensive and time consuming clinical trials, and hence increase the efficiency of device development is of great importance.

To perform successful CFD calculations on the nose, construction of a proper surface grid of the nasal cavity is important. The process of building the surface grid was presented in Kleven *et al.* (2004), but will to some degree be repeated here. The surface grid was next imported into the grid generator Tgrid for volume grid generation, and finally the simulations were carried out by use of the CFD code Fluent. This code provides comprehensive modelling capabilities for a wide range of fluid flow problems, where a useful group of models are the multiphase flow models. According to Fluent User's Guide (2003), there are currently two approaches to the numerical calculation of multiphase flows; Euler-Lagrange approach and Euler-Euler approach. In this work, i.e. an analysis of a gas-particle flow, the discrete phase model (DPM) is used. The Lagrangian DPM in Fluent follows the Euler-Lagrange approach. The fluid phase is treated as a continuum by solving the time-averaged Navier-Stokes equations, i.e. the conservation equations, while the dispersed phase is solved by tracking a large number of particles through the calculated flow field. The dispersed phase can exchange momentum, mass, and energy with the fluid phase. A fundamental assumption made in this model is that the dispersed second phase occupies a low volume fraction. The particle or droplet trajectories are computed individually at specified intervals during the fluid phase calculation. This makes the model appropriate for any application where the volume fraction of the second phase is negligible.

Thorough validation against physical experiments is essential for the reliability and value of the CFD computations. The present paper presents comparison results from the initial design process, where simple numerical models (using the computational code Matlab version 6.5, 2002) are used to verify critical parameters, for instance airflow and particle size. The mathematical model is built in Matlab to verify results from a similar model in Fluent. *In vitro* experiments in a human nasal airway model are used for comparison against results from numerical simulations in Fluent. This paper briefly presents some of the verification process.

## MODEL DESCRIPTION

### Initial design process – Matlab vs. Fluent

Initially, a simple mathematical model was established in Matlab. The same geometry was also drawn in the grid generator Gambit, with the ability to be read in Fluent. The geometry was divided into two main parts; a simple pipe, and a U-bend. These two features correspond to, respectively, the inlet and exit nostrils, and the curvature posterior in the nose, where the air flow turns due to the closed passage further down the throat. The two geometries are illustrated in Figure 2 and Figure 3.



Figure 2: Sketch of the simple pipe.

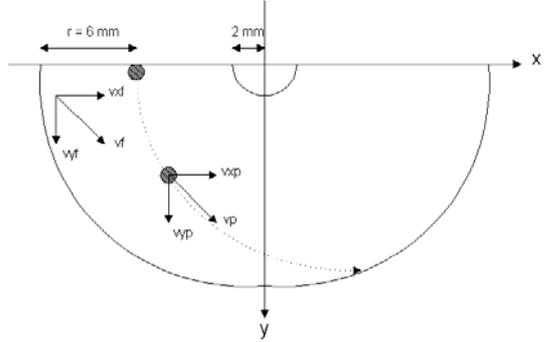


Figure 3: Sketch of the U-bend.

In this work, the gas phase is treated as a continuum, while the dispersed phase is solved by performing trajectory calculations. The equation describing the velocity of an individual particle in the two models, i.e. in Matlab and Fluent, is found by demanding a force balance for a particle:

$$\frac{d(m_p \vec{v}_p)}{dt} = \frac{1}{2} \rho C_D A |\vec{v} - \vec{v}_p| (\vec{v} - \vec{v}_p) + \vec{g} V_p (\rho_p - \rho) \quad (1)$$

The first term on the right-hand-side of Eq. (1) represents the drag force, and the next terms represent the gravity force. In addition, we could have added other forces, like the lift force, virtual mass force and Basset force. During the simulations, only the drag force is included in the model, which results in

$$\frac{d\vec{v}_p}{dt} = \frac{1}{2m_p} \rho C_D A |\vec{v} - \vec{v}_p| (\vec{v} - \vec{v}_p) \quad (2)$$

Integrating Eq. (2) with respect to time yields the particle velocity. The time is the time from particle release. The drag coefficient is taken from

$$C_D = a_1 + \frac{a_2}{\text{Re}} + \frac{a_3}{\text{Re}^2}, \quad (3)$$

where the constants  $a_1$ ,  $a_2$  and  $a_3$  apply for smooth spherical particles over several ranges of Re given by Morsi and Alexander (1972). The gravity force will depend on the position of the head, and since it usually is not well-defined how a person is positioned when using the medication device, this force was neglected, however easy to include. Other forces are not of importance in this study. The position of the particle is calculated from

$$\frac{d\vec{x}_p}{dt} = \vec{v}_p \quad (4)$$

In the Matlab simulations, only one particle was tracked in the pipe and the U-bend from a release point on the surface representing the inlet. The release point was placed in the middle of the pipe for both Matlab and

Fluent models. Single injections based on this release point were defined for different particle sizes. In this work, it is assumed that as soon as a particle hits the wall, it is deposited, and hence the condition *trap* in Fluent seems reasonable. This condition prevents a particle from flowing farther into the nose after colliding with the wall. It was assumed that the density of the particles is equal to that of water. A particle size distribution was not included.

The effect of air flow rates was also tested. In modelling the continuous air flow field, the Fluent model deviates from the model built in Matlab. In the latter model, a constant and uniform tangential velocity field was used, while the velocity field in Fluent is solved by general conservation equations, i.e. the Navier-Stokes equations. The equation for conservation of mass for the continuous phase, can be written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = 0 \quad (5)$$

The right-hand-side of Eq. (5), the source term, is set to zero, since we assume no mass is added to the continuous phase from the dispersed second phase. Conservation of momentum in the  $i$ th direction in a non-accelerating reference frame is described by

$$\frac{\partial (\rho v_i)}{\partial t} + \frac{\partial}{\partial x_j} (\rho v_i v_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + F_i \quad (6)$$

The last term on the right-hand-side of Eq. (6),  $F_i$ , represents the contributions from the dispersed phase. The stress tensor  $\tau_{ij}$  is given by

$$\tau_{ij} = \left[ \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] - \frac{2}{3} \mu \frac{\partial v_l}{\partial x_l} \delta_{ij} \quad (7)$$

We assume steady state for the continuum, hence, the transient terms in Eqs. (5) and (6) can be neglected. In this work, the equations are solved by the segregated solver in Fluent, i.e. the equations for the conservation of mass and momentum are solved sequentially, iterating until convergence.

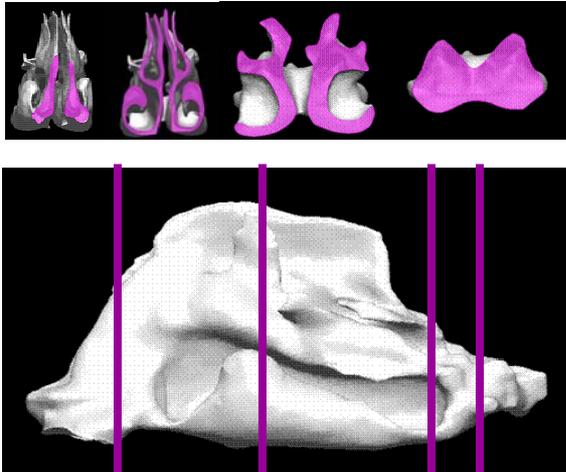
To visualize the collision of the particle against the pipe wall from the Matlab plots, the x- and y-coordinates for the inner and outer pipe wall are plotted together with the particle path and the position of collision.

## Design process – CFD-model in Fluent

### Surface grid generation

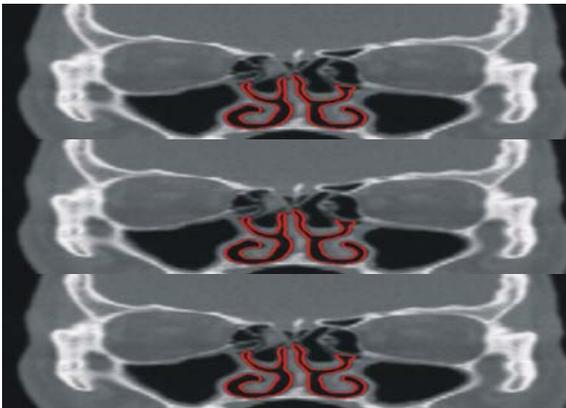
The first step of the modelling process was to acquire a high resolution CT-scan (Computer Tomography) of the human nose. During the imaging process, a tube was inserted in the left nostril to represent the device. CT is a radiologic technique based upon computer assessment of the radiation-absorbing characteristics of the various tissues of the body. The sensitivity of CT makes it possible to identify hard and soft tissues in multiple, sequential, radiographic slices, i.e. tomograms. The location, shape, and size of these tissues can be evaluated. Montgomery *et al.* (1979) concludes that CT can be used to get an accurate volume measurement of the nasal

airway, and that the cross-sectional areas along the airway can be determined. The CT-imaging of one of the authors was executed by Rikshospitalet (National Hospital of Norway), generating 109 slices of the nasal geometry with 1 mm between each slide.



**Figure 4:** Illustrates cross-section slices along the nasal geometry from an anterior view, representing from left to right; the nasal valve, sinus ostia, the posterior region with the inferior turbinate, and the bi-directional connection.

Prior to further processing, these data were segmented, that is, each voxel in the data was given a unique label indicating whether it belongs to the airway or not. This is a classical problem in image processing, and many possible methods exist. We used a Markov Random field based method, see Derin and Elliott (1987), as these produce particularly smooth interfaces between the different groups of voxels, see Figure 5 for an example.



**Figure 5:** The surface of the nasal cavity is identified as part of the segmentation process.

The next step in the process was to produce from the segmented image data a surface triangular mesh that modelled the surface of the nasal cavity. The mesh was later used as input to the volume grid generator and was to comply with a set of constraints and quality measures dictated by the grid generator and the CFD software. These measures can be divided in two categories; geometric quality and mesh quality. We wanted on the one hand a mesh that closely modelled the geometry of the surface of the nasal cavity and on the other hand a mesh configuration that gave high quality volume grids of

the cavity itself. To summarize, the most important requirements were that the mesh:

- Closely approximated the underlying geometry.
- Had the correct topology and was a closed surface without singularities and self intersections.
- Was geometrically smooth whenever the underlying model was smooth.
- Had well shaped triangles, i.e. triangles should not be too far from equilateral.
- Had a suitable number of triangles due to the efficiency of the CFD software.
- Had triangle density proportional to the local *thickness* of the geometry.

Our overall strategy was to first generate an initial mesh directly from the segmented image. This mesh was typically unsuitable for the grid generation step, with far too many triangles, topological and geometric artefacts and suboptimal mesh structure. We therefore took a number of steps such as decimation, smoothing, refinement and mesh optimization to enhance its quality. Due to the often competing requirements, the nature of the enhancement process was iterative.

Our first step was to employ the Marching Cubes (MC) algorithm of Lorensen and Cline (1987), which in this setting can be seen as to extract the surface of the segmented voxels in a consistent and efficient manner. We experienced that the *Generalized Marching Cubes* algorithm used in AMIRA (Amira 3.0 User's Guide, 2002) produced non-manifold meshes, requiring tedious manual fixing. However, the original *Marching Cubes* algorithm did not suffer from the same problems and was found better in the present context. The resulting mesh revealed artefacts such as topological holes stemming from inaccuracies in the segmented data. One of the requirements to the mesh was that it had the correct topology and was a closed surface without singularities and self intersections. We found it convenient in this situation to go back to the 3D image and remove the artefacts in the images before generating an updated mesh with the MC algorithm. After that, the mesh was pruned for irrelevant parts and holes where filled by manually editing the mesh. Having successfully obtained a surface mesh with the right topology we could begin enhancing the geometry of the model.

The next step was to smooth the initial mesh, getting rid of jagged geometry. One important requirement during the surface mesh production was that the mesh should be geometrically smooth whenever the underlying model was smooth. We used variants of Laplacian Smoothing, see Taubin (1995), as it is a straightforward and common smoothing method. It is based on mesh topology alone and thus does not respect the local geometry of the mesh very well, especially where neighbouring triangles differ significantly in size. A more sophisticated method is presented in Desbrun *et al.* (1999), where the intrinsic geometry of the surface is taken into account, giving a better geometric smoothing. We also found a rather effective smoothing method that we called *barycentre* smoothing as it amounts to moving each vertex to the average of the positions of its mesh neighbours, i.e.

$$v_i = \frac{1}{n_i} \sum_{j=1}^{n_i} v_{ij}$$

The first two methods were typically iterated until we got a visually pleasing mesh. The barycentre method on the other hand is very effective with one or two steps sufficient. A side effect that we learned to appreciate was that it also improved triangle shape significantly with many triangles being smoothed to be nearly equilateral. The smoothing methods mentioned had the tendency to shrink the volume enclosed by the surface. We countered this problem with a common solution: scaling the vertex positions such that the volume remained the same after each smoothing step.

The number of triangles was reduced to a size that was manageable for the grid generator, typically 100 000 triangles. Mesh decimation algorithms such as the one in Garland and Heckbert (1997) are typically based on a primitive mesh decimation operator such as the half edge collapse; a vertex is removed by identifying it with a neighbouring vertex, reducing the number of vertices by one and the number of triangles with at least two. In addition a norm measuring the quality of the mesh is used. Based on this, a standard *greedy* strategy can be employed, where one in each step perform the decimation operation that reduce the geometric quality the least. In other words, we reduce the size of the mesh while keeping the most significant geometric information. In addition a set of constraints can be built in, such that e.g. operations that produce badly shaped triangles are not allowed. In principle we could have used the criterions listed in the section above to guide the decimation process, but we found that it was not necessary to maintain high mesh quality during the decimation process. We thus employed the algorithm described in Garland and Heckbert (1997), where the most important optimization criterion is geometric approximation quality. The other quality criterions were taken into account at a later stage. We reduced the number of triangles from more than a million to around 75 000 which amounted to 75% of our triangle budget. This gave us good enough approximations while leaving enough room for later refinement if necessary.

The triangle shapes and distribution was then improved. One could think of this as the problem of redistributing the triangles so that they obtained better shape, while still representing the same geometry. The input mesh to the grid generator must have nicely shaped triangles. Our objective was to minimize the overall skewness and avoid triangles with skewness above 0.75.

The methods we used to optimize over skewness fall into two categories, optimizing over either vertex positions or vertex connectivity. We found the Laplacian Smoothing type of techniques described above particularly effective. The reason for the effectiveness was that these methods are based on averaging the geometry and thus the resulting mesh stays close to the geometry of the initial mesh. One step of barycentre smoothing was very effective in that respect. The other type of method we used was based on keeping the vertices of the mesh fixed while optimizing the vertex connectivity. The basic operation in the former type of method is the edge flip illustrated in Figure 6.

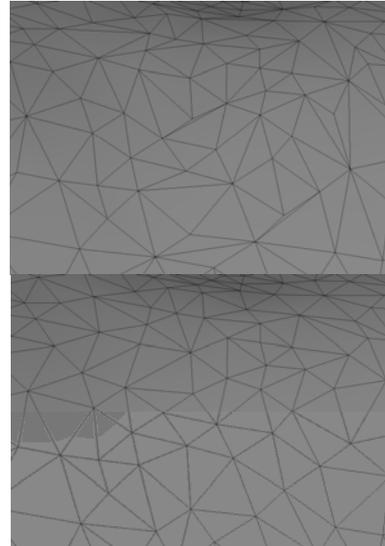


**Figure 6:** Edge flips improve the overall skewness of the triangle mesh.

We used the greedy local optimization (LOP) technique (Lawson, 1972, and Dyn *et al.*, 2001), with the cost for a triangle amounted to an exponential function of its skewness  $s$ , on the form

$$c_i = Cs^p,$$

with  $C = 10$  and  $p = 5$ . This ensured that triangles with high skewness were punished hard. The cost function in the LOP algorithm was the sum of this expression over all triangles. We also tested more advanced optimization techniques, but concluded that the simple LOP algorithm was equally effective and produced meshes with few skew triangles. If there were still triangles with high skewness we improved their quality manually. See Figure 7 for an example of mesh optimization with respect to skewness.



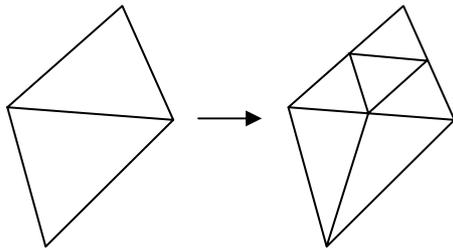
**Figure 7:** Part of the mesh before and after swapping with the LOP algorithm.

The grid generation process produces volumetric grids with element size proportional to the size of the surface triangles. In fact, a surface triangle in the input mesh ends up as a facet of a tetrahedron in the volumetric mesh. Therefore it is of vital importance that the density of the surface triangles is adapted to the thickness of the geometry. In other words, the size of the surface triangles is required to be proportional to the thickness of the geometry, with small triangles in thin regions.

Thickness of the geometry could be defined in different ways. We define the thickness in a point on the surface to be the interior distance to other parts of the surface. In practice, we estimated the local thickness at a triangle by measuring the distance to the nearest surface triangle in the normal direction. Due to the instable nature of these calculations, we employed a median filter to obtain consistent thickness values throughout the mesh. We found the results of this method satisfactory and sufficient for our purposes. Defining also the size of a triangle to be the length of its longest edge, we required the triangle size

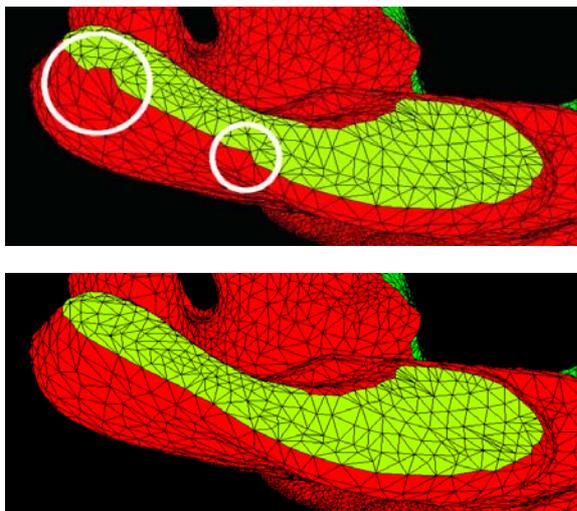
to be a factor 0.2 of the local thickness. We allowed in other words the volume to be at approximately five elements thick. In practice we adapted the local triangle density by adaptively refining the mesh in thin areas, splitting edges in two and dividing the triangles in smaller triangles. Special care had to be taken for triangles neighbouring triangles that were meeting the thickness criterion, see Figure 8. We repeated the following steps until our thickness criterion was met:

1. Associate thickness with each triangle.
2. Split triangles with too high triangle size.
3. Optimize the split triangles with respect to triangle shape.



**Figure 8:** Adaptively splitting triangles in thin regions.

After the above steps we would have a final surface mesh that met all our requirements with respect to geometric- and surface mesh quality. The final step was to divide the mesh into logical partitions each consisting of a set of connected triangles that would model inlet/outlet or other areas of special interest. We assigned each triangle to one unique partition and used this as part of the mesh description. The nature of the inlet/outlet partitions allowed us to use a simple region growing algorithm based on surface normals. We subsequently edited the partitions manually to obtain visually smooth partitions, see Figure 9 for example.

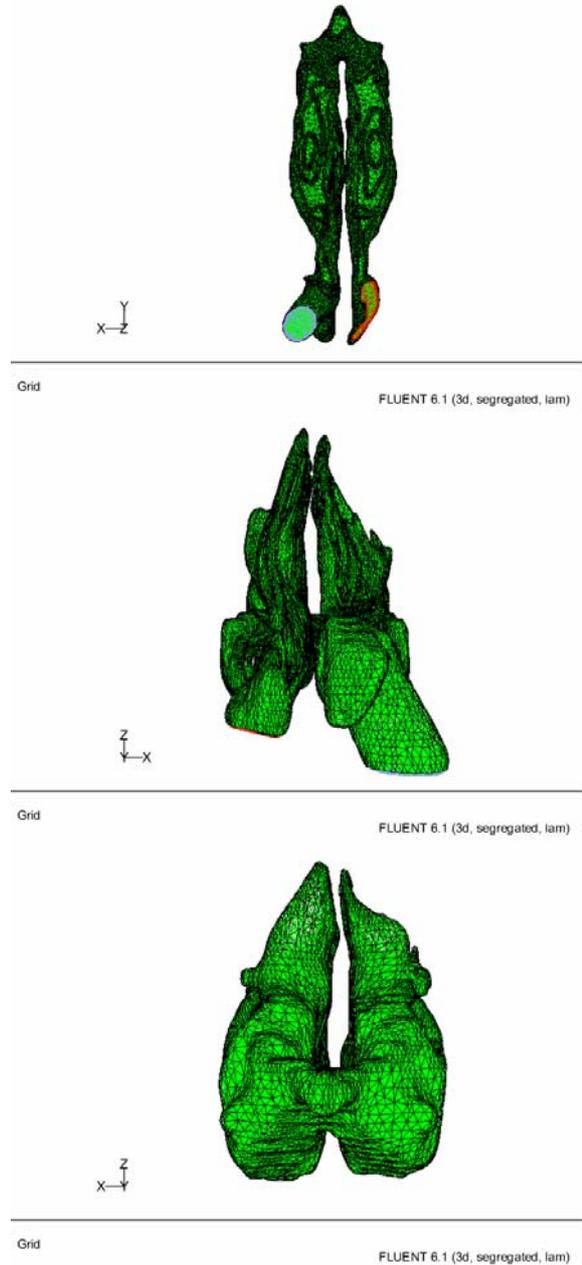


**Figure 9:** Creating visually smooth partitions was our final step.

#### **Volume grid generation in Tgrid**

By use of the default algorithm included in Tgrid, i.e. the highly automated Delaunay method (Tgrid 3 User's Guide, 1997), a tetrahedral mesh of the nasal cavity was generated. The cell quality was tested in Tgrid, to ensure a

proper grid for the simulations. This was done by investigating the degree of skewness, this time based on comparison of the cell's shape to an equilateral cell of equivalent volume. Again, the skewness range is between zero and one, where zero skewness is optimal and a skewness of one indicates a degenerate cell. The images from the grid in Fluent are shown in Figure 10. The final grid consists of approximately 500 000 cells.



**Figure 10:** Images from Fluent showing the grid. Upper image: grid seen from below, including the medication tube in the left nostril. Middle image: frontal grid view. Lower image: grid seen from behind.

#### **CFD simulations in Fluent**

The volumetric grid was subsequently imported into the commercial CFD code Fluent. The equations and conditions used are the same as described for the simple Fluent model used in the initial design process. The nasal geometry, though, represents major challenges, both seen

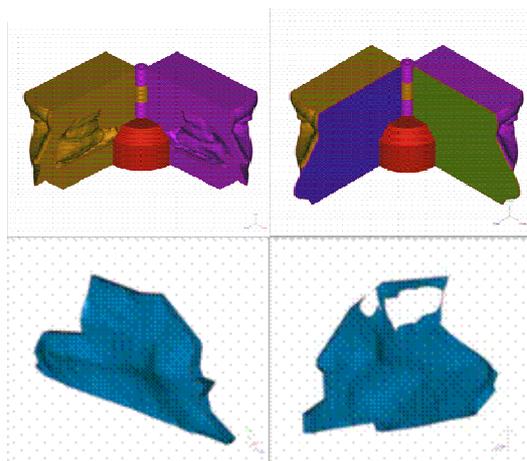
during the grid generation process, but also after the grid has been read into Fluent.

The fate of a particle in the air passage will be influenced by the dimensions of the space through which it passes and the nature of the airflow in that passage. The characteristics of importance of airflow are the velocity of the stream, bends in the passage, and the degree of turbulence of the flow. Other factors that influence the deposition and absorption of the medicaments from a nasal spray include particle size, particle size distribution, shape and density (Proctor and Andersen, 1982, and Cole, 1993). The angle of the spray nozzle piece in the nasal vestibule and particle velocity from the nozzle also contributes (Cheng *et al.*, 2001).

The flow in the nose is, however, neither turbulent nor laminar, but lies in the transitional regime (Cole, 1993). Modelling a flow in the transitional regime is seen as a difficult task. Since the turbulence affects the distribution and uptake of medications in the nasal cavity in a positive manner, we decided to simulate the airflow as a laminar flow in this study. This would rather under-estimate the degree of deposition than the opposite. The particles were released on the face of the left nostril medication unit, see Figure 10. The simulations were carried out with reasonable values for high and low air flows and different particle sizes.

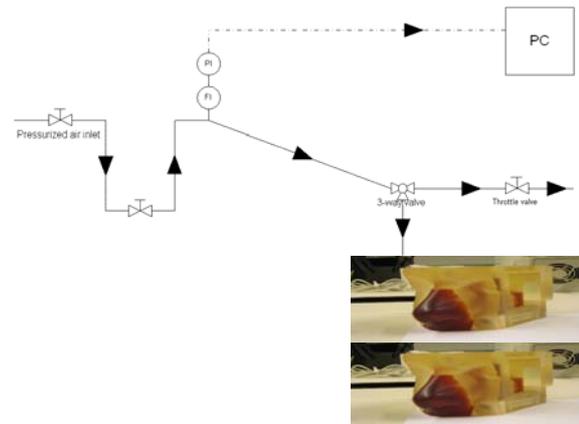
#### Verification of CFD simulations

Essential for the CFD computations is thorough validation of the results. Without comparison against physical experiments, the simulations are of limited value. For this purpose, comparison data is produced by use of a nasal cast model developed by the Norwegian company VINN Design AS. An ultra violet light source hardens thin layers of an epoxy resin in a stereolithography instrument (SLA), to build a very accurate model from digitally imaged data. According to Bache (2004), the thickness of the layers is 0.075 mm. The digitally imaged data used to build the nasal cast model are the same data, that is the CT-data that was used to construct the surface grid for the model in Fluent. In other words, the same nasal geometry is used to build both models.



**Figure 11:** Illustrates the nasal cast for visualization and deposition studies. Upper images: Open model without the inserts (left) and with plates and inserts in position (right). Below images: Septal inserts.

The test rig used in the laboratory is shown in Figure 12. The pressurized air used is regulated to achieve the required air flow in the experiment. The air flow, together with the pressure, are measured, and logged. Hence, trends from the different tests can be saved and viewed in the computer. The 3-way valve is used to obtain two switchable paths for the air flow. During the time needed to prepare the nasal cast model for an injection, the flow passes a bypass throttle valve on its way out of the system. This valve ensures the same pressure loss in each path, through adjusting the throttle valve to the same pressure drop as the path going via a tube and throughout the nose model.

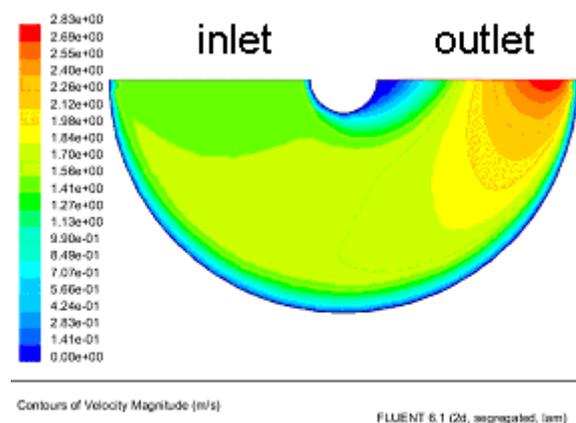


**Figure 12:** Sketch of the test rig used in the laboratory.

## RESULTS

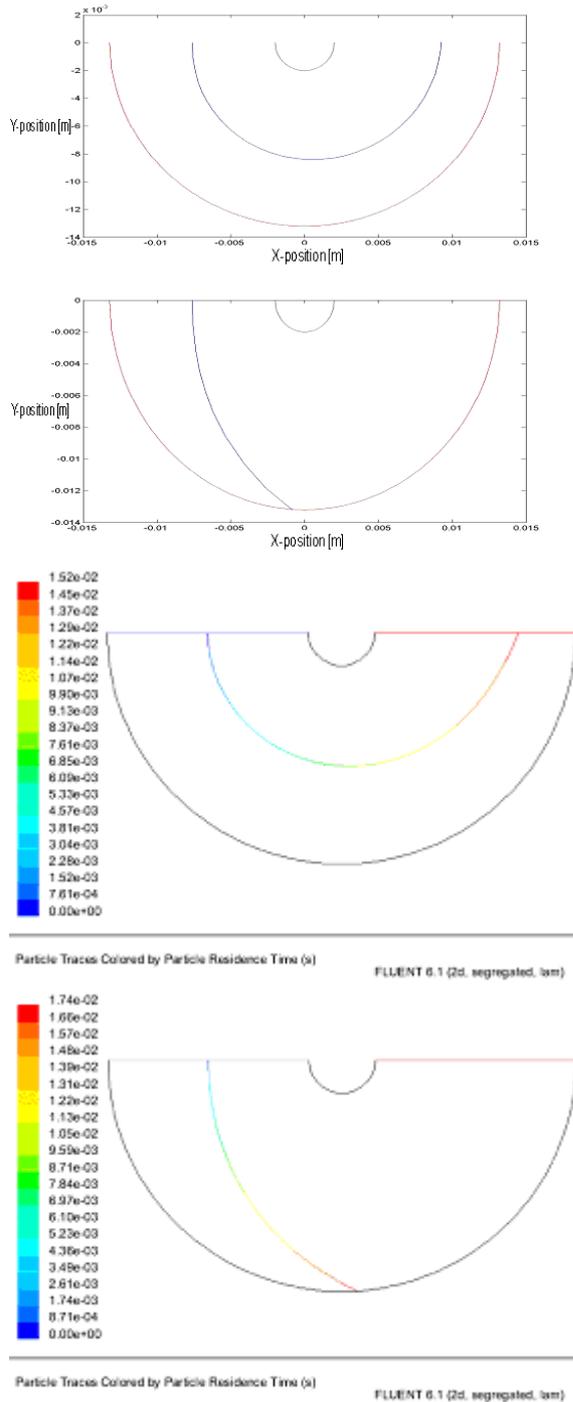
### Initial design process – Matlab vs. Fluent

Comparison between the elapsed times of the Fluent simulations and the Matlab simulations shows that both the simulation tools give logical results and follow the same trends, when changing the air flow, and/or the particle size. The elapsed times indicate how long a particle spends in the domain of calculation (Fluent User's Guide, 2003). The residence times from the Matlab simulations are, however, a bit higher than those from the Fluent simulations. This is because in Matlab, the velocity field is constant throughout the simulation, while it changes along the pipe in Fluent. The more realistic velocity field is the one in Fluent, illustrated in Figure 13.



**Figure 13:** Contour-plot of the velocity field throughout the U-bend.

The air flow in both these figures is 10 l/min, which corresponds to an inlet air velocity of 1.5 m/s.



**Figure 14:** Particle traces from, respectively, Matlab (upper images) and Fluent (lower images). Air flow is 10 l/min, and the two particle sizes showed are 10 and 50 micron, respectively upper and lower image in the two groups of images from Matlab and Fluent.

The corresponding paths for, respectively, particle size 10 micron and 50 micron, both for the Matlab model (upper images) and the Fluent model (lower images) are shown in Figure 14. As the particle size increases, the particle hits the wall earlier due to inertia forces. In the Matlab script, the above image is a result of the defined plot that shows

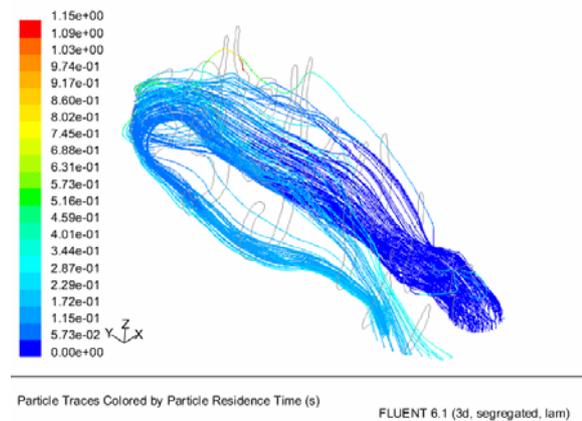
the particle movement from start and to the step after collision. In the script, commands are given, to print the crash position element in the Matlab command window. In addition, the vectors for the step before and after collision, that means the  $x$ - and  $y$ -positions, and the  $x$ - and  $y$ -velocities of the particle, are printed. This is tested in case of large computational steps in the collision area.

It appears that increasing the air flow results in a higher deposition rate, i.e. the particles collide earlier with the wall.

From the plots, one can conclude that intuitive physical phenomena are reflected in both the Matlab and Fluent models. It is important to note that Fluent yields the most *correctly* velocity field, due to use of the Navier-Stokes equations. For the U-bend, the constant and uniform velocity field in Matlab does not consider the recirculating zone seen at the outlet side in Figure 13.

### Design process – CFD-model in Fluent

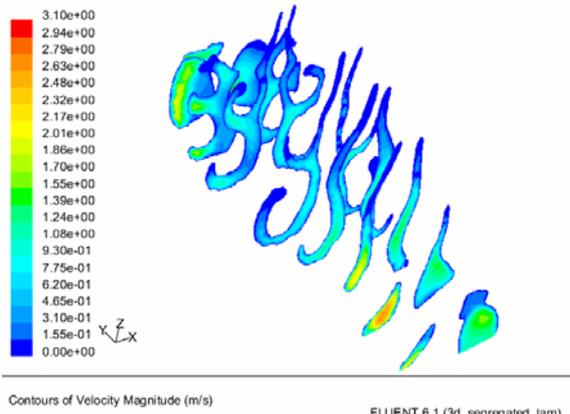
Fluent manages complex geometries and complex fluid flows. A typical result from the simulations of the bi-directional flow pattern in Fluent is shown in Figure 15, where the nose geometry is included and a number of particles are released on the face of the left nostril medication unit. The figure shows the path of particles from the introduction into the left nostril and to the back of the nasal cavity, where the particles turn and follow the air stream out of the right nostril. The darker lines show cross-sections of the nasal cavity. The particle traces are coloured by the particle residence time. The colour scale in all the simulation results goes from red representing high values, to blue representing low values.



**Figure 15:** Flow of particles in the nasal cavity with 6 l/min air and a uniform particle size of 3.5  $\mu$ m. The darker lines show the cross-sections.

From Figure 15, it can be seen that the number of particles in the left nostril is considerably higher than in the right nostril. The difference in particles entering and exiting the nostrils is assumed to be the amount of particles that deposit in the nasal cavity. In this particular simulation, 54.6% of the particles by-passes the bi-directional nasal passage.

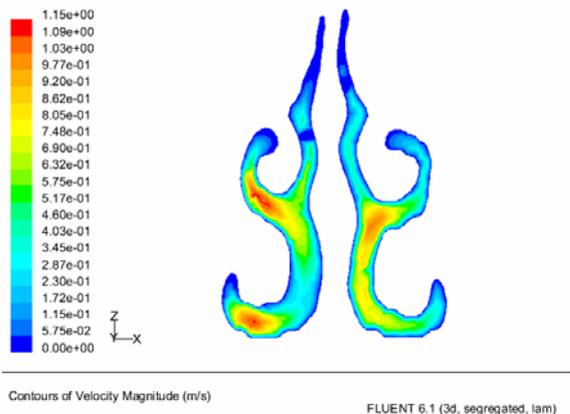
We can also focus on the cross-sections of the nasal cavity, illustrated by the darker lines in Figure 15, and study the gas velocity distribution with the air flow of 6 l/min.



**Figure 16:** The velocity distribution in the cross-sections of the nasal cavity with an air flow of 6 l/min.

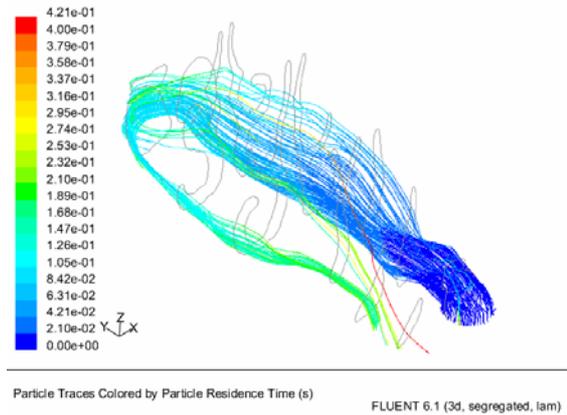
The air velocity is highest where the airway passage is the narrowest; where the air flow turns around the nasal septum, and by the nasal valve in the exit nostril, see Figure 16. Due to the placement of the delivery device, and hence an expanding of the nasal valve, in the left nostril, the same air velocity increment is not seen around the nasal valve on this side.

We can also pick one of the cross-sections of the nasal cavity, illustrated by the darker lines in Figure 15, and study the velocity distribution from an orthogonally view. The air velocity is highest in the exiting nasal passage, however, the mass flow of air in the two nasal passages is equal, as shown in Figure 17.



**Figure 17:** The velocity distribution in one of the cross-sections of the nasal cavity with a 6 l/min air flow.

An equal number of 10  $\mu\text{m}$  and 3.5  $\mu\text{m}$  particles are released from the delivery device, see Figure 15 and Figure 18. By increasing the particle size, but keeping the air velocity constant, less particles follow the air stream around the nasal septum and out the right nostril. This is due to inertia forces. In this simulation, where the uniform particle size is set to 10  $\mu\text{m}$ , 30.8% of the particles bypasses the bi-directional nasal passage, compared to 54.6% for 3.5  $\mu\text{m}$  sized particles.



Particle Traces Colored by Particle Residence Time (s) FLUENT 6.1 (3d, segregated, iam)

**Figure 18:** Flow of particles in the nasal cavity with 6 l/min air and a uniform particle size of 10  $\mu\text{m}$ . The darker lines show the cross-sections.

The deposition rate changes with a change in the air flow as described by Kleven *et al.* (2004), who concluded that the particle deposition rates increased with both increased air velocity and increased particle size, seen in simulations carried out by the same procedure as used in the present paper.

#### Experimental results comparison with CFD simulations

Quantitative comparison of the simulations with *in vitro* results from tests performed with the experimental model are at an early stage of development. An example of a typical particle deposition map is shown in Figure 19. In this experiment a dry red powder was injected into the model so that the particle deposition on the inner surfaces of the model could be readily visualized.



**Figure 19:** The open nasal cast model photographed after injection of a red-coloured powder.

The powder had a slightly higher mean particle size than the particles tracked in both Figure 15 and Figure 18, which corresponds to the low degree of deposition in the exit nostril (Figure 19). In addition, the air flow in this exact experiment was also higher than 6 l/min.

The powder used in the *in vitro* experiments has a relatively broad particle size distribution, contrary to the mono-sized particles used in the Fluent simulations. Nevertheless, a similar deposition pattern can be seen in the Fluent simulations. Clearly, it will be necessary to see the effect on deposition of including a particle size distribution in future numerical simulations.

As described earlier in this paper, the flow in the nose, and in the nasal cast model, will vary between the laminar and turbulent regimes. In areas where turbulence occurs, more particles are likely to be deposited. So far the Fluent simulations have assumed laminar flow in the nose. Analyzing the flow pattern more thoroughly and testing different turbulence models for low Reynolds number flow will be performed in the future studies. Perhaps the use of a turbulence model will improve the results further.

## CONCLUSION

Some simple simulations on a model of nasal geometries have been performed by the use of Matlab and Fluent. The intuitive physical phenomena are reflected well in both the models. For instance, for a given flow smaller particles follow changes in the fluid velocity more faithfully than larger particles. This implies that for the bi-directional delivery relatively small particles will predominantly pass the single nasal cavity at the posterior of the nose, whereas larger particles are more likely to be deposited further upstream as their inertia will prevent them following the flow as it changes direction.

This work has also shown that by starting with a CT-scan, one can establish a nasal geometry surface mesh, a volume mesh, and then run simulations on the nasal cavity, yielding simulation results. We believe that CFD simulations have the potential to keep the research development in medical companies like OptiNose on a high level. It is important to note that such simulations are of minor use without thorough validation of the results. It is important to focus on establishing experimental series of data from physical tests that can provide a platform for validation. The *in vitro* experimental analysis described is at an early stage of development but the initial results are promising. By developing this initial qualitative method of comparing data to a quantitative simulation, it will eventually be possible to quantify the Fluent model prediction. It is thus possible that CFD computations can increase the efficiency of device development process by reducing the number of expensive and time consuming laboratory experiments and clinical tests.

Improvements to the Fluent model and the grid used may be important in refining the simulation prediction. For instance, it might be that grid refinement will be necessary. Analyzing the need for use of turbulence models, and introducing a particle size distribution in the simulations, are two obvious areas of improvement. Due to differences in nasal structures from person to person, it will be interesting to establish geometrical grids for several noses, to see the effect of anatomical differences.

## ACKNOWLEDGEMENTS

VINN Design AS is OptiNose AS's Norwegian engineering consultant for design development, and has developed the nasal cast models used in the verification process.

## REFERENCES

AMIRA 3.0 USER'S GUIDE, (2002), Mercury Computer Systems/3D Viz group, Merignac Cedex, France.

BACHE, S.N., (2004), "Nidarosdomen digitaliseres – Slik gjøres det", *Teknisk Ukeblad*, 151. årgang, nr. 4, 13. Februar.

CHENG, Y.S., HOLMES, T.D., GAO, J., GUILMETTE, R.A., LI, S., SURAKITBANHARN, Y. and ROWLINGS, C., (2001), "Characterization of Nasal Spray Pumps and Depositon Pattern in a Replica of the Human Nasal Airway", *J. Aerosol medicine*, **14**(2), 267-280.

COLE, P., (1993), "The respiratory role of the upper airways - A selective clinical and pathophysiological review", St. Louis, United States of America: Mosby-Year Book, Inc.

DERIN, H. and ELLIOTT, H., (1987), "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1** (9), 39-55, January.

DESBRUN, M., MEYER, M., SCHRÖDER, P. and BARR, A.H., (1999), "Implicit Fairing of Arbitrary Meshes using Diffusion and Curvature Flow", *Proc. ACM SIGGRAPH '99*.

DJUPESLAND, P.G., SKRETTING, A., WINDEREN, M. and HOLAND, T., (2004), "Bi-directional Nasal Delivery of Aerosols Can Prevent Lung Deposition", *J. Aerosol medicine*, **17**(3), 249-259.

DYN, N., HORMANN, K., KIM, S.J. and LEVIN, D., (2001), "Optimizing 3d triangulations using discrete curvature analysis". In mathematical methods for curves and surfaces: Oslo 2000, Lyche T., Schumaker L.L. (eds.), Vanderbilt University Press, 135-146.

FLUENT USER'S GUIDE, (2003), Release 6.1, Fluent Inc., Centerra Resource Park, Lebanon, January.

GARLAND, M. and HECKBERT, P.S., (1997), "Surface Simplification Using Quadric Error Metric", *Proc. SIGGRAPH 97*.

KLEVEN, M., MELAAEN, M.C., REIMERS, M., RØTNES, J.S., AURDAL, L., DJUPESLAND, P.G., (2004), "Fluid flow and particle deposition simulations in the human nose", *Proc. SIMS 2004*, Copenhagen, Denmark, 373-380, September 23-24.

LAWSON, C., (1972), "Transforming Triangulations. Discrete Mathematics 3", 365-372.

LORENSEN, W.E. and CLINE, H.E., (1987), "Marching Cubes: a high resolution 3D surface reconstruction algorithm", *Proc. SIGGRAPH 87*.

MATLAB VERSION 6.5, (2002), Release 13, The MathWorks, Inc.

MONTGOMERY, W.M., VIG, P.S., STAAB, E.V. and MATTESON, S.R., (1979), "Computed tomography: A three-dimensional study of the nasal airway", *American Journal of Orthodontics*, **76**(4), 363-375, October.

MORSI, S.A. and ALEXANDER, A.J., (1972), "An Investigation of Particle Trajectories in Two-Phase Flow Systems", *J. Fluid Mech.*, **55**(2), 193-208, September 26.

PROCTOR, D.F. and ANDERSEN, I., (1982), "The nose – Upper airway physiology and the atmospheric environment".

TAUBIN, G., (1995), "A Signal Processing Approach to Fair Surface Design", *Computer Graphics*, **29**, 351-358.

TGRID 3 USER'S GUIDE, (1997), First Edition, Fluent Inc., Centerra Resource Park, Lebanon, June.