

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/273078591>

# Approximating Computational Fluid Dynamics for Generative Tall Building Design

**Article** in *International Journal of Architectural Computing* · June 2014

DOI: 10.1260/1475-472X.12.2.155

CITATIONS

9

READS

650

2 authors:



**Samuel Wilkinson**

Hoare Lea

**13** PUBLICATIONS **31** CITATIONS

[SEE PROFILE](#)



**Sean Hanna**

University College London

**100** PUBLICATIONS **448** CITATIONS

[SEE PROFILE](#)

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



ENFOLDing [View project](#)



Approximating CFD for Generative Design - UCL EngD [View project](#)

# Approximating Computational Fluid Dynamics for Generative Tall Building Design

Samuel Wilkinson and Sean Hanna



# Approximating Computational Fluid Dynamics for Generative Tall Building Design

Samuel Wilkinson and Sean Hanna

Background literature review, methodology, results, and analysis are presented for a novel approach to approximating wind pressure on tall buildings for the application of generative design exploration and optimisation. The predictions are approximations of time-averaged computational fluid dynamics (CFD) data with the aim of maintaining simulation accuracy but with improved speed. This is achieved through the use of a back-propagation artificial neural network (ANN) with vertex-based shape features as input and pressure as output. The training set consists of 600 procedurally generated tall building models, and the test set of 10 real building models; for all models in both sets, a feature vector is calculated for every vertex. Over the test set, mean absolute errors against the basis CFD are 1.99–4.44% ( $\sigma$ :2.10–5.09%) with an on-line process time of 14.72–809.98s (0.028s/sample). Studies are also included on feature sensitivity, training set size, and comparison of CFD against prediction times. Results indicate that prediction time is only dependent on the number of test model vertices, and is therefore invariant to basis CFD time.

## I. INTRODUCTION

Although computational fluid dynamics (CFD) has existed now for over 50 years and parametric CAD for over 30, both have seen an increased interest in architectural practice over the last decade. However, in computational design, especially in generative exploration or optimisation, CFD remains a challenging simulation tool to integrate. There are at least three reasons for this: firstly, the cost of expertise and software is high; secondly, the relationship between a design and its fluid environment is complex, often subtle, and esoteric; and thirdly, the time required to achieve accurate results is typically greater than that available, namely at early project stages when the guidance provided by the simulation is most valuable.

The third issue is fundamentally one of approximation, a trade-off common in simulation of time against accuracy. This relationship is marked by the two characteristic extremes of fast-inaccurate and slow-accurate, with a range of solutions existing along this spectrum. Responses to this problem can be categorised into two forms: i) type-one is solver approximation, including all conventional CFD methods which by one approach or another seek to emulate the full underlying physical fluid behaviour. Any approach of this type can only fit within the time-accuracy spectrum since the two properties remain dependent; ii) type-two is solution approximation, encompassing methods which aim to emulate simulation behaviour. Through model reduction and machine learning, a break from the established trade-off can enable a move towards fast-yet-accurate approaches.

Effects of the wind upon buildings are numerous: for pedestrian comfort in surrounding proximity; ventilation and therefore thermal comfort and indoor air quality; and structural performance. Wind loads, along with seismic, are the two primary external forces that increase with building height. Therefore tall buildings have been identified as a focal typology for this and a number of reasons. The aerodynamic shape has a primary impact on these forces and therefore subsequently on the overall structural, material, energy, and financial performance.

The trend has always been to build them as high as (contextually, economically and structurally) possible, necessitating cutting-edge design and construction technologies. With the quantity, height, and complexity of tall buildings still increasing, there is a greater need for early-stage form analysis and optimisation. The geometric complexity in the latest generation has broken away from the previously necessary extruded planform to more freeform shapes. This has been facilitated by the recent ubiquity of computation in design, analysis, fabrication, and construction. Tall buildings generally lend themselves well to parametric design since there is often a strong vertical repetition which can be expressed easily computationally. It also means it is suitable for generating procedural models that, with a

relatively small number of parameters, can represent a large number of potential designs.

### 1.1. Contribution

The aim of this paper is to demonstrate a new approach to approximating wind pressure on tall buildings for the application of generative design exploration and optimisation. The predictions are approximations of time-averaged computational fluid dynamics (CFD) data with the aim of maintaining simulation accuracy but with improved speed. This is achieved through the use of a back-propagation artificial neural network (ANN) with vertex-based shape features as input and pressure as output. Success of the approach is measured against the objective of being fast-yet-accurate; therefore the time and errors are quantified in the end discussion.

## 2. LITERATURE REVIEW

The background review of existing literature is divided into three parts: i) solver approximation; ii) solution approximation; and iii) shape features.

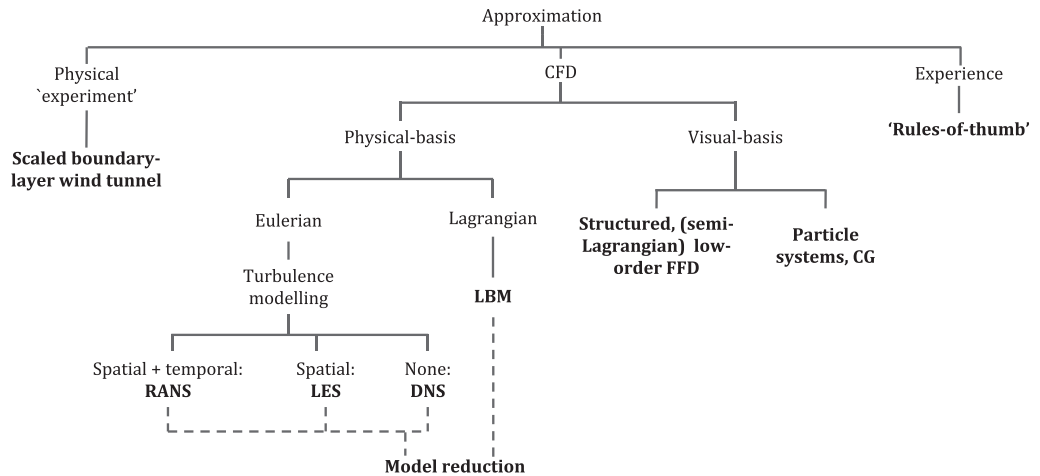
### 2.1. Solver approximation

Most approaches towards CFD approximation focus on simplification of the solver itself. For instance: simplified meshes (spatial discretisation); the use of lower-order equations; or the treatment of turbulence through modelling. These methods can be classed as *type-one, solver approximation* (Figure 2.1). For instance, RANS (Reynolds-Averaged Navier-Stokes), LES (Large Eddy Simulation), and DNS (Direct Numerical Simulation) all treat turbulence with different numerical approaches, i.e. temporally, spatially, and directly.

Another example is the 'Stable Fluids' fast fluid dynamics (FFD) solver developed by [1] for the computer graphics and games industries, which has subsequently been developed and tested for architectural applications [2, 3, 4, 5]. Development and application for architectural design was motivated by three factors: a validation study suggested it as suitable for purpose, even though it was limited to indoor, low Reynolds number flow regimes [6, 7]; the qualitative appearance of accuracy for turbulent flows; and its remarkable speed compared to traditional CFD methods like RANS (Reynolds-Averaged Navier-Stokes). [6] implemented the FFD with a zero-equation turbulence model but found that it performed worse since it was not designed or suited to the FFD approach. It should be noted, however, that with a lack of turbulence model, the solver relies on continuous interaction (such as game character movement) to compensate for numerical dissipation.

Although such recent developments aim to increase the speed of CFD, they do so at the direct expense of accuracy; the opposite of traditional CFD development where accuracy is increased at the expense of speed.

► Figure 2.1:  
CFD solver  
approximation  
taxonomy.



The benefit of solver over solution approximation is the availability of full spatial field data for all fluid properties, although in some cases such as the FFD, production of surface data is more difficult due to the structured mesh approximation (voxelisation).

## 2.2. Solution approximation

Another possible approach to this problem, type-two, is *solution approximation*. CFD originated in aeronautics and astronautics, as such there is a large quantity of work directed towards modelling and optimisation of aerofoils, fuselages, and turbine blades. An optimisation routine will often generate large data sets of simulation data, from which knowledge of the problem can be extracted. The following fall into the greater category of supervised machine learning approaches where the relationship between an input feature vector extracted from some geometry and the ground truth data output from full CFD simulation is learnt.

In one such approach, a large model set of turbine blades is used with a decision tree to analyse the relationship between point deformation of models and their change in surface pressure [8, 9]. Areas of high sensitivity can then be mapped onto a pre-defined base geometry and used to focus subsequent analysis. Extension of this work incorporates an evolutionary optimisation process, so as to use the information extracted from previous cases to create non-random initial populations of solutions and to guide the evolution.

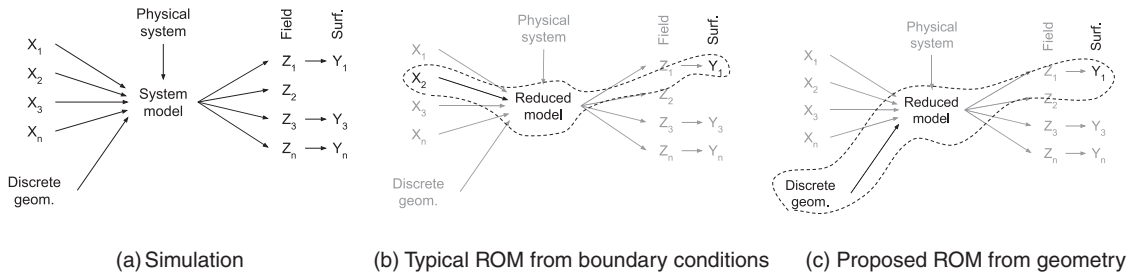
Analyses that are potentially obstructive to the design process may involve partial differential equations (PDEs), such as the Navier-Stokes equations of fluid flow and the Maxwell equations in electromagnetism [10]. Whilst simulation of these phenomena can give high-accuracy results, they are computationally expensive and cannot be computed in real-time. As a

result, a design process using high-accuracy techniques has inherently slow response times and loses any desired interactivity.

Significant efforts have been made to reduce the complexity of these systems in order to make them interactive or suitable for optimisation; this is generally referred to as *model reduction*. Reduced-order models (ROMs) approximate representations of system behaviours, namely for computational simulations with slow response times, with the aim to create a lower-dimensional system model whilst retaining predictive fidelity [11, 12]. They typically do so by restricting the input quantities to boundary conditions and outputs to those of interest (e.g. lift, drag, or a quantity at a single point).

In one example, [13] use spatial and behavioural parameters as input features to a radial basis function (RBF). The RBF is used to interpolate and merge CFD and wind-tunnel data on pressure coefficient values (lift and drag) for aerofoil analysis. They use an input feature vector  $C_p\{x, y, z, a, M, Re\}$ : where  $x, y, z$  is the spatial position;  $a$  the angle of attack;  $M$  the Mach number; and  $Re$  the Reynolds number.

▼ Figure 2.2: Reduced-order model schematic.



Whilst this method proved successful for linking behavioural characteristics ( $a$ ,  $M$ , and  $Re$ ) to data sources (CFD and wind-tunnel), it is limited to a single geometry, thus the use of explicit spatial positions ( $x$ ,  $y$ , and  $z$ ). For cases of differing geometry between training and testing, spatial positions become non-unique and can therefore not be used within the feature vector. This necessitates the use of either explicit global design parameters or implicit local shape description.

Using spatial positions (or mesh node numbers) for a feature vector is also proposed by [14]. In this case, an ANN is used to predict post-processed CFD data for rapid visualisation and interpolation of boundary conditions with an augmented reality user display system. The input feature vector,  $\mathbf{X}$ , and output response,  $\mathbf{Y}$ , are defined as:  $\mathbf{X}\{n, S, P\}$ ,  $\mathbf{Y}\{T, V\}$ , where  $T$  is the air temperature and  $V$  is the air speed at a node,  $n$  is the mesh node number (1224 nodes in the cubic room),  $S$  is the supply temperature, and  $P$  is the supply pressure. Again the proposal is strongly limited to not only a single geometry but to a single mesh by the use of spatial positions or node

numbers corresponding consistently with fixed locations. The limitation again is that for differing training/test geometry the positions are non-unique.

### 2.3. Shape features

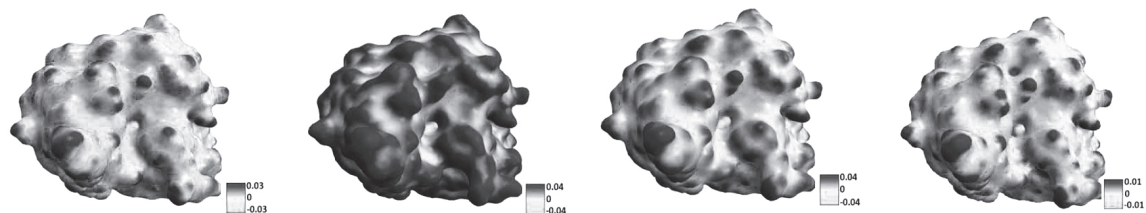
A component of the methodology involves a localised description of a sample point or vertex, constituting the input vector for the machine learning. These are predominantly shape or topological characteristics, although this can be extended to local fluid flow properties. With discretised surface representations (meshes) there is often a need to describe local shape features for a broad range of applications.

An exemplar case is for scale-invariant surface descriptors for the matching of molecular surface regions to identify potential chemical functionality, i.e. binding molecules often have locally complementary shapes [15, 16]. When calculating surface curvature, the distance or neighbourhood size must be included, shown in Figure 2.3. From left to right the features are: minimum curvature; maximum curvature; mean curvature; and Gaussian curvature.

There are a large number of similar studies on mesh curvature, edge detection, and invariant shape analysis, for example [17, 18, 19, 20, 21, 22, 23]. Applications range from chemistry, to rationalising and reconstructing 3-D scanned models, to identifying constant features in images for camera stabilisation.

Whilst [9] do not learn the function between local shape features and pressure to make predictions on new cases, the ability, however, to generate the sensitivity at a point from its deformation is an inspiration for two key elements of this work. Firstly, it highlights the importance of showing the pressure distribution over the entire model rather than calculating a global metric of a design's success, i.e. considering the problem locally rather than globally. Secondly, the use of top-down models (those found in parametric models with global variables) have an inherently limited flexibility. They can be adjusted infinitely within the bounds of the logic of the model parameters, however it is firstly difficult to alter these foundations at later stages and secondly each model will have a different logic, variables, or set of dependencies. It is therefore difficult to use these global variables (for example, a parametric tall building may have *Height* and *Taper Factor* as two of its defining variables) as input features to learning. If they are used, the

▼ Figure 2.3: Shape analysis for protein shape matching in biochemistry [16].





problem being learnt is restricted to that logic. [9], on the other hand, use a local mesh vertex deformation as input feature. Since all CFD simulations require a surface mesh of the geometry to be generated, it is a relatively simple generalisation to use the mesh data and its derivatives as input for the learning.

Further generalisation of the method is proposed, specifically for automobile design, in particular for the detection of design novelty or for characterising families of similar products [8]. The key similarity with this work is the use of unstructured surface meshes as the basis geometric representation, which in itself is a good foundation for generating training data due to the high acceptance in design practice. The distinction however is in the definition of the actual learning process and feature vector: [8] use a deformation metric from a base case; as opposed to the here proposed broader shape description.

Their proposed shape mining process focuses on the extraction of performance data from conventional analysis processes for compilation of a large database. From which a meta-representation, or reduced-order model, can be created and used for sensitivity analysis, concept retrieval, and interaction analysis. These data modelling and knowledge formation components link back holistically to knowledge utilisation and decision making processes (DMPs).

### 3. METHODOLOGY

The approach can be split into the following steps: i) training set procedural geometry generation; ii) training set CFD evaluation; iii) training set feature calculation; iv) ANN training; v) test set geometry generation; vi) test set CFD evaluation; and vii) prediction and assessment.

#### 3.1. Procedural training geometry

The parametric model was created in *GenerativeComponents* [24]. The goal was to create a generalised tower model, with the two properties of minimising the number of parameters used whilst maximising the design representation potential, i.e. the number of possible buildings it could create. This is important when considering optimisation or exploratory design space searches to avoid the curse of dimensionality. This means that as the number of variables increases, the design space increases exponentially by  $n^D$ , where  $n$  is the number of samples taken per parameter and  $D$  is the number of parameters, or dimensionality. There is therefore clearly a compromise to be made between model efficiency and representability.

The geometry for the training set was generated using a procedural tall building model with a select number of key parameters. There are in fact three separate topologies in the procedural model with their own parameters, since it is difficult to incorporate the entire design space with

one parametric logic [25, 26]. Using the unstructured triangulated surface mesh from these means we are not limited by a single parametric topology in the learning phase of the method [9]. Local surface-mesh shape characteristics are used as input features to the learning algorithm instead of the design parameters, avoiding reliance on any one parametric model definition.

Contemporary tall building design, as discussed previously, has been freed up by computational design tools, analysis, and construction methods. The iconicity of skyscrapers is also a driving force for unique, bespoke forms; as such, increasingly complex forms are being planned and constructed. This presents an interesting challenge to wind engineers who, for the new generation of tall buildings, typically struggle to find general behavioural rules akin to the top-down global learning approach, i.e. bespoke designs require bespoke analysis. In a similar way, a procedural tall building model is used to generate the reduced-order model and is tested on an extended set of 10 real buildings.

The geometry for the training set was generated using a procedural tall building model with a select number of key parameters. There are three separate topologies in the procedural model each with their own parameters, shown in Figure 3.1.

The three procedural models can be classified as one of three topologies: *Extrusion*; *Periodic*; and *Blocks*. The topology is initially selected randomly, and then each can be used to generate instances by randomly assigning parameter values from the ranges given in Table 3.1.

With these parameter sets and ranges, the maximum number of potential instances for the three topologies respectively is:  $2.56e^{16}$ ;  $2.34e^{21}$ ; and  $3.24e^{12}$ ; giving a sum of the three of  $2.3448e^{21}$ . Although this is the maximum number, certain combinations or regions of the parameter space lead to invalid instances which reduces the total. These are filtered out in the code simply with a `while(solid.Success==false)` statement. A training set of 600 instances is generated, giving a sampling of  $2.56e^{-17}$  % of the total parameter space.

### 3.2. CFD simulation

CFX 13.0 [27] is used for the steady-state time-averaged Reynolds-Averaged Navier-Stokes (RANS) simulations with a  $k-\epsilon$  turbulence model. Typically the models are meshed with roughly an equal number of cells (up to the maximum available computational resources), of around four million elements. Each simulation, depending on the complexity, requires on average 1914.470s ( $\sigma$ :629.808s) to converge on a 2.66GHz i7 4GB RAM.

For the ground, a no slip smooth wall is assigned (i.e. fluid velocity at wall boundary is zero); for the sides and top parallel to the flow, a free slip wall (i.e. zero shear stress from wall friction); and for the outlet, a zero relative pressure opening. For the inlet, the wind profile is applied as

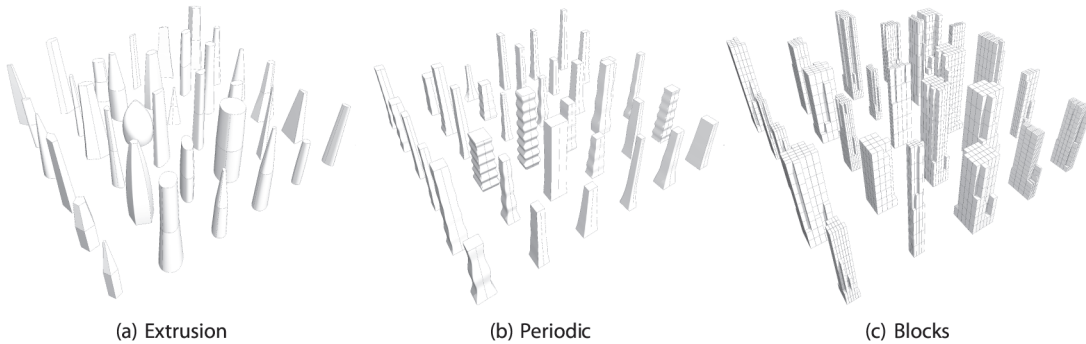
described below, with a medium intensity turbulence and eddy viscosity ratio [28].

Basic simulation parameters are: high-resolution advection and turbulence numerics; isothermal fluid at 25°C; a scalable wall function; and a convergence residual target of  $1.0e^{-6}$  RMS. The following meshing parameters are used: an unstructured tetrahedral domain mesh, with patch independence; a boundary surface element size of 5m; a model surface minimum size of 0.20m and maximum face size 0.25m; for prismatic expansion, a growth rate of 1.2, a transition ratio of 0.77, and a maximum of 3 layers.

▼ Table 3.1: Procedural training model parameter ranges.

	<i>N</i>	<i>W</i>	<i>D</i>	<i>H</i>	<i>mSF</i>	<i>tSF</i>	<i>F</i>	<i>o</i>	<i>R</i>	<i>A</i>	<i>D</i>	<i>f</i>	<i>fO</i>	<i>Nw</i>	<i>Nd</i>	<i>Nh</i>
<i>Extrusion</i>																
Min.	3	10	10	100	0.5	0.1	0.3	2	0	-	-	-	-	-	-	-
Max.	7	60	60	200	1.1	1.1	5.1	4	180	-	-	-	-	-	-	-
Inc.	1	0.1	0.1	0.1	0.1	0.1	0.1	1	0.1	-	-	-	-	-	-	-
<i>Periodic</i>																
Min.	-	10	10	100	-	-	0.5	-	0	0.1	100	0.1	-180	-	-	-
Max.	-	20	20	200	-	-	20	-	180	10	1000	2	180	-	-	-
Inc.	-	0.1	0.1	0.1	-	-	0.1	-	0.1	0.1	1	0.1	1	-	-	-
<i>Blocks</i>																
Min.	-	10	10	100	-	-	-	-	0	-	-	-	-	4	4	5
Max.	-	60	60	200	-	-	-	-	180	-	-	-	-	7	7	15
Inc.	-	0.1	0.1	0.1	-	-	-	-	0.1	-	-	-	-	1	1	1

**Note:** *N* no. edges; *W* width [m]; *D* depth [m]; *H* height [m]; *mSF* mid planform scale factor; *tSF* top planform scale factor; *F* fillet radius [m]; *o* planform curvature order; *R* orientation [°]; *A* amplitude; *D* decay; *f* frequency; *fO* frequency offset; *Nw* no. blocks in width; *Nd* no. blocks in depth; *Nh* no. blocks in height.



▲ Figure 3.1: Procedural training model sets.

The wind speed is applied at an upstream inlet with a reference speed ( $v_r$ ) of  $10m \cdot s^{-1}$  at a reference height ( $z_r$ ) of 10m. The most commonly used distribution of wind speed with height is the ‘power-law’ expression:

$$v_x = v_r \cdot (z_x/z_r)^\alpha \quad (1)$$

The exponent  $\alpha$  is an empirically derived coefficient that is dependent on the stability of the atmosphere. For neutral stability conditions it is

approximately 0.143, and is appropriate for open-surroundings such as open water or landscape [29].

Both the training (Figure 3.2) and test sets are evaluated under the same boundary conditions and with the same wind profile. The ‘static’ pressure [Pa or  $N \cdot m^{-2}$ ] is the force per unit area, taken at sample-points (vertices) over the model surfaces (mesh).

### 3.3. Shape feature vector

The basic concept is to define the pressure at a point or vertex on a model by its geometric characteristics, i.e. its relative position on the model, proximity to an edge, curvature, relative position on the vertical wind profile distribution, and direction of orientation. These features are calculated for every vertex  $\mathbf{V}$ , along with its pressure, to be used as a sample. The  $\mathbf{R}^{23}$  definition of the model is now:

$$f^{ANN} : (Z, \mathbf{n}_{x,y,z}, \mathbf{n}\sigma_{x,y,z}^{1-5}, \mathbf{T}_{x,y,z}) \rightarrow P \quad (2)$$

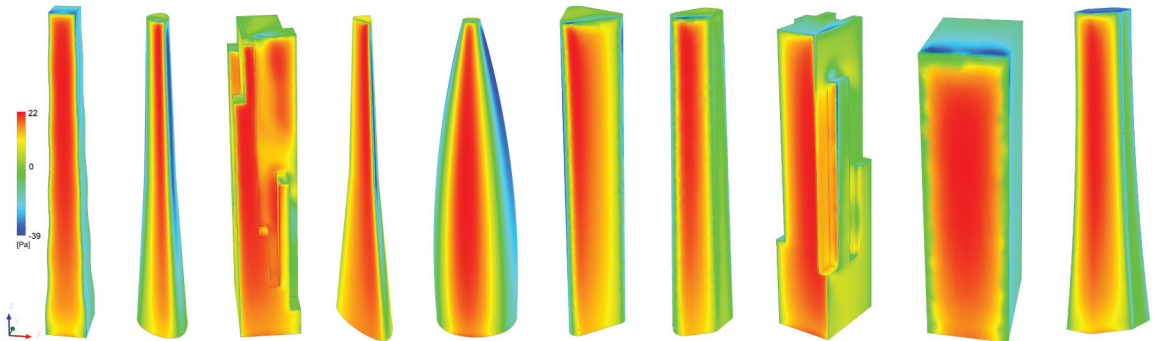
- HEIGHT:  $Z$  is the vertical position of  $\mathbf{V}$ , i.e.  $\mathbf{V}_z$  ;
- NORMAL:  $\mathbf{n}_{x,y,z}$  are the normal components of  $\mathbf{V}$ ;
- CURVATURE:  $\mathbf{n}\sigma_{x,y,z}^{1-5}$  is the standard deviation of the vertex normals in each independent ring, inversely weighted by the distance. This is also visualised in Figure 3.3.

$$\mathbf{n}\sigma^r = \left( \frac{1}{n-1} \sum_{i=1}^n \frac{(\mathbf{n}_i - \bar{\mathbf{n}})^2}{d} \right)^{\frac{1}{2}} \quad (3)$$

Where:  $r$  is the vertex neighbourhood ring;  $n$  is the number of vertices in  $r$ ;  $d$  is the distance between each vertex in  $n$  and the central feature vertex;  $\bar{\mathbf{n}}$  is the average of the normals in  $r$ ;  $\mathbf{n}_i$  are all the normals in each neighbourhood ring,  $r$ .

In this case, convex mesh curvature is given as a positive standard deviation and negative for concave regions. Figure 3.3 shows how this is calculated in 3-D. The basic procedure is as follows:

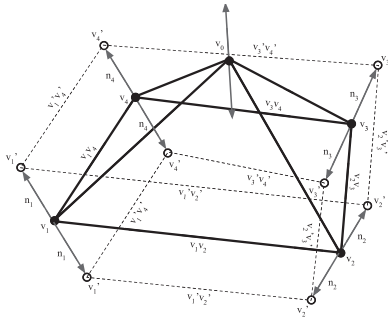
▼ Figure 3.2: Example set of evaluated procedural models.



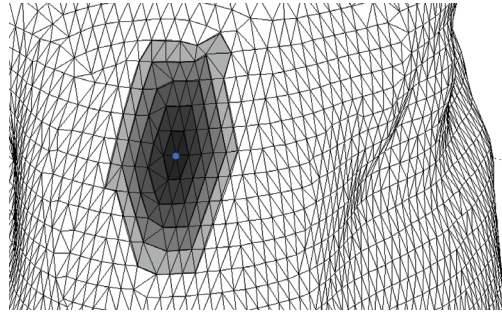
for vertex  $v_i$ , its offset is the vertex plus normal  $v'_i = v_i + n_i$   
 if  $(v'_1...v'_n > v_1...v_n)$  then  $v_1...v_n$  is convex,  $\sigma^+$   
 if  $(v'_1...v'_n < v_1...v_n)$  then  $v_1...v_n$  is concave,  $\sigma^-$   
 else  $v_1...v_n$  is planar,  $\sigma = 0$

The extension of the standard deviation to be positive for convex vertex neighbourhood and negative for concave vertex neighbourhoods allows for greater accuracy and applicability to a broader set of forms. The use of various neighbourhood scales (rings one through five) gives the local curvature over a range of scales, as can be seen in Figure 3.4. In these images the white regions are concave and black are convex. Due to the complexity of the meshes here though certain points may in fact be convex in one axis and concave in another. For this reason the  $x$ ,  $y$ , and  $z$  components are given for each scale. The images show the mean of all three components to give the primary indication of curvature direction.

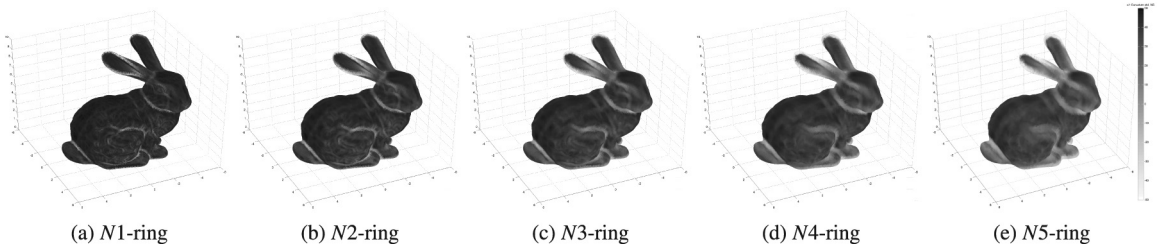
The convex-concave calculation is shown in Figure 3.4 on the Stanford Bunny [30], a standard test model used in computer graphics. The model consists of 69451 triangular polygons, and shows the extension of the curvature neighbourhood from the vertices' first to fifth neighbourhood rings.



(a) N1 convexity / concavity calculation.



(b) Vertex neighbourhoods N1 – 5.



▲ Figure 3.4: Convex-concave curvature analysis over N1 to N5 neighbourhoods.

- **POSITION:**  $\mathbf{T}_{x,y,z}$  is the normalised position of  $\mathbf{V}_i$  within the range of all model vertices  $\mathbf{V}$

$$\mathbf{T}_i = \frac{(\mathbf{V}_i - \mathbf{V}_{min})}{(\mathbf{V}_{max} - \mathbf{V}_{min})} \quad (4)$$

- **PRESSURE:**  $P$  is simply the pressure at  $\mathbf{V}$  as extracted from the simulation. This can quite easily be replaced with any dependent secondary metric, such as force or the pressure coefficient.

The feature generation time, for both training and test models, is currently 0.02784 s/sample. For example, a model with a mesh of 1000 vertices currently requires 27.84s. The following pseudocode is a

```

for each mesh {
  read each  $\mathbf{v}\{x,y,z\}$  {
    calculate minRange $\{x,y,z\}$ 
    calculate maxRange $\{x,y,z\}$ 
  }
  read each  $\mathbf{i}\{a,b,c\}$ 
  read each  $\mathbf{n}\{x,y,z\}$ 
  read each  $P$ 
  for each vertex {
    for each neighbourhood ring (0 to 5) {
      find indices of connected vertices, e.g. for vertex index a:  $\mathbf{r0}\{a\}$ ,  $\mathbf{r1}\{b,c,d,e,f\}$ , etc.
    }
    for each neighbourhood ring (1 to 5) {
      calculate standard deviation of vertex normals in  $\mathbf{r1-5}$ 
    }
    print vertex feature  $\mathbf{X}\{z,\mathbf{n},\mathbf{n}\sigma^{1-5},\mathbf{T}\}$  and  $Y\{P\}$ 
  }
}

```

The output of the calculation, per vertex, is simply a 23-dimensional vector. E.g.  $z\{0.52\}$ ,  $\mathbf{n}\{-0.68,0.72,-0.05\}$ ,  $\mathbf{n}\sigma^1\{-0.03,-0.03,-0.02\}$ ,  $\mathbf{n}\sigma^2\{-0.07,-0.07,-0.03\}$ ,  $\mathbf{n}\sigma^3\{-0.11,-0.10,-0.02\}$ ,  $\mathbf{n}\sigma^4\{-0.15,-0.14,-0.02\}$ ,  $\mathbf{n}\sigma^5\{-0.20,-0.18,-0.03\}$ ,  $\mathbf{T}\{0.79,0.13,0.04\}$ ,  $P\{-0.17\}$

From the training feature set, the reduced-order model is generated by a back-propagation artificial neural network (ANN), with a hyperbolic tangent sigmoid transfer function [31]:

$$\text{tansig}(x) = 2/(1 + \exp(-2 \cdot x)) - 1 \quad (5)$$

The ANN structure  $X:H:Y$  is 22:20:1, i.e. 22 input neurons, 20 hidden layer neurons, and 1 output. The sensitivity analysis on the number of neurons in the hidden layer, and the number of layers, is not included here; although 20 in a single layer has been seen to be sufficient. There is generally no rule-of-thumb or guidance to define either, necessitating sensitivity analysis for each problem.

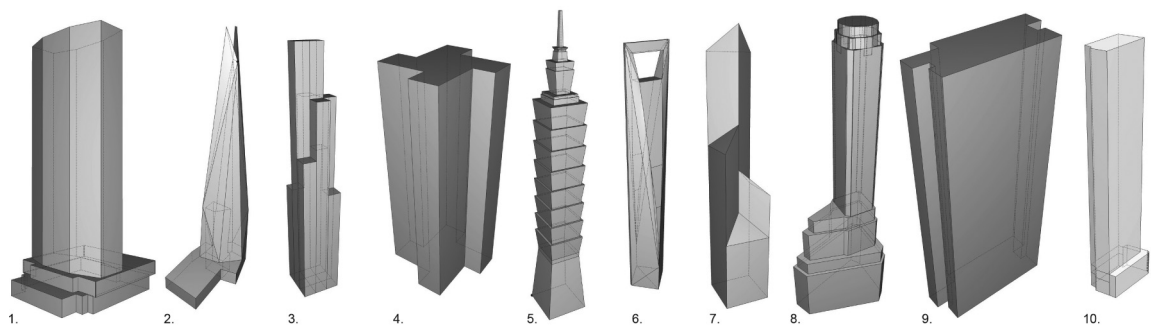
3.4. Test data set

The models are selected from *Google Earth*, rebuilt in *GC* as solids, and evaluated with *CFX*. They were selected relatively arbitrarily, except for the one criteria that each has a unique architectural design feature. For instance, the set contains features such as pedestals (1 and 2), tapering (2), stepping (3 and 7), concavities, corner filleting (5), voids (6), spires (5), etc.

Note that the heights range between 124 and 508m, whilst the procedural model used to generate the training set has a height range of 100 to 200m. The test models were therefore all scaled to 100m to reduce the amount of sampling required for the training set.

	Name	Location	Height [m]	Completion date
1	Met Life Building	New York City, US	246.3	1963
2	The Shard	London, UK	306.0	2013
3	Willis Tower (Sears)	Chicago, US	442.1	1974
4	Euston Tower	London, UK	124.0	1970
5	Taipei 101	Taipei, Taiwan	508.0	2004
6	Shanghai World Financial Centre	Shanghai, China	492.0	2008
7	Bank of China Tower	Hong Kong, China	367.4	1990
8	20 Exchange Place	New York City, US	225.9	1931
9	Frankfurter Buro Center	Frankfurt, Germany	142.4	1980
10	123 Washington Street	New York City, US	192.1	2010

◀ Table 3.2: Real building test set details.



▲ Figure 3.5: Real building test set models.

## 4. RESULTS

Firstly, sensitivity analyses are conducted on the feature vector components and the training set size using data only from the procedural models; followed by an assessment of the accuracy of the predictions on the real building test set.

The distinction is drawn between the simulation output response  $Y$  from CFD, and the prediction output response  $Y'$  from the reduced-order model. For a single vertex sample  $i$ , the difference between the  $Y$  and  $Y'$  is used to calculate the sample prediction error,  $\delta_i$  :

$$\delta_i = (Y'_i - Y_i) / (Y_{max.} - Y_{min.}) \quad (6)$$

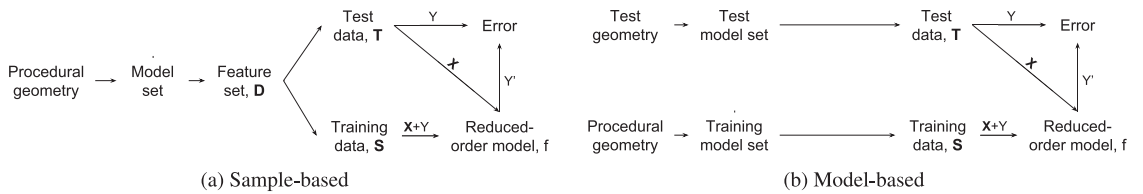
The descriptive statistics used for reporting the errors throughout are:

$\delta_{min.}$	real-valued minimum of the error range [%]
$\delta_{max.}$	real-valued maximum of the error range [%]
$ \bar{\delta} $	mean of the absolute error range [%] ( $ \bar{\delta}  \neq  \delta $ )
$\sigma_{ \delta }$	standard deviation of the absolute error range [%] ( $\sigma_{ \delta } \neq \sigma_{\delta}$ )

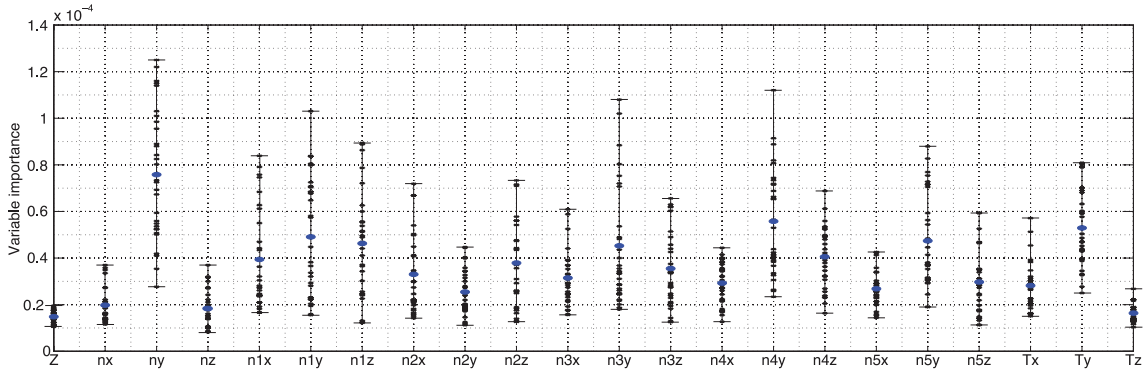
There are two types of test used here: sample-based (Figure 4.1a) or model-based (Figure 4.1b). In the sample-based assessment, test data set  $\mathbf{T}$  of size  $m$  and training data set  $\mathbf{S}$  of size  $n$  are drawn from the same set of available data  $D$ , meaning that  $m = D - n$ . Both  $\mathbf{T}$  and  $\mathbf{S}$  are randomised in this case, and are used to monitor error convergence during the ANN training. For model-based tests, a completely different test set is used so that  $\mathbf{T}$  and  $\mathbf{S}$  are independently generated, such as in the case where a procedural model is used for training and real models for testing.

Whilst the descriptive statistics for the sample- and model-based accuracy are a good indicator of the ROM's performance, a qualitative visual comparison of simulated and predicted surface pressure is also included at the end.

▼ Figure 4.1: Testing of sample- and model-based data sets.







▲ Figure 4.2: Feature importance for  $\mathbf{X}\{z, n, n\sigma^{1-5}, \mathbf{T}\}$ .

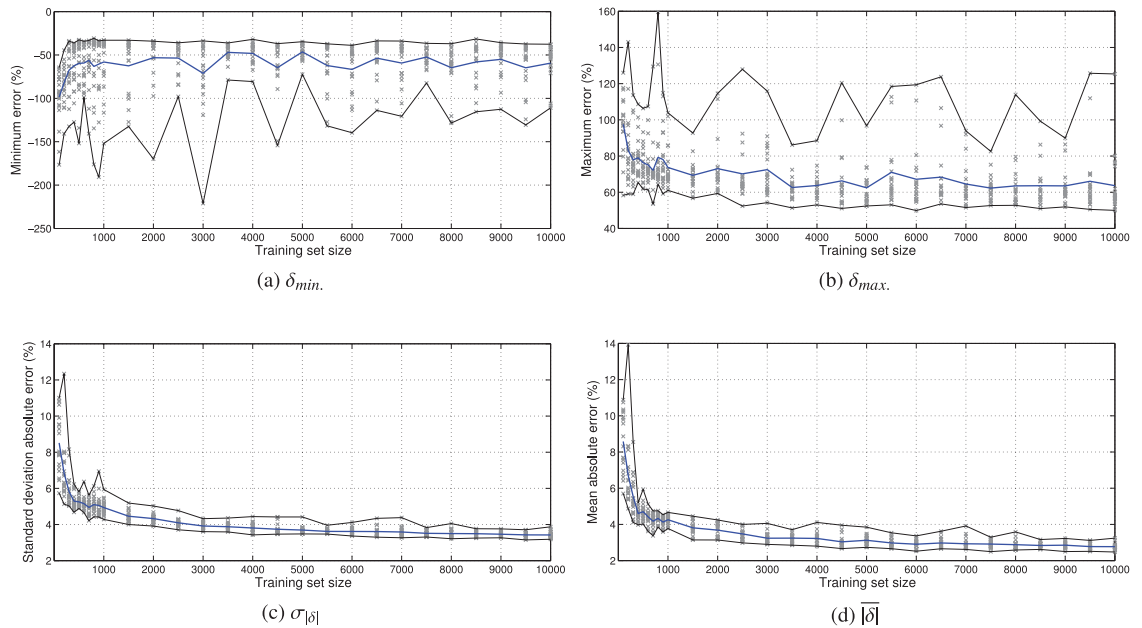
#### 4.1. Feature sensitivity analysis

Given the feature vector definition of  $\mathbf{X}\{z, n, n\sigma^{1-5}, \mathbf{T}\}$ , each of the 22 input components has a different significance, importance, or sensitivity to the output. Whilst this varies between problems and geometry, a measure of importance can be calculated during generation of the reduced-order model based solely on the training data set. The random forest method [32], specifically the *TreeBagger* [33] algorithm, intrinsically calculates the *OOBPermutedVarDeltaError*. A set size of 10000 randomly sampled from the full training set is used; 10 trees for the *TreeBagger* algorithm; and the process is re-run 30 times to take the mean and range.

The first observation is that primarily the  $y$ , and secondarily the  $x$ , components are typically the most significant part of vector. The  $x$  (across flow) and the  $y$  (stream-wise) direction components determine whether the point is facing into, perpendicular to, or away from the flow. Which in turn is the primary indicator of a positive or negative pressure. Also, note that the variability or distribution increases with feature importance: the standard deviation  $\sigma$  and the mean have an  $r^2 = 0.795$ .

#### 4.2. Training set size sensitivity analysis

Sample-based errors given are at the converged training set size (Figure 4.3) of  $n=10000$ . The test set size  $m$ , being the full data set  $D$  minus the training set  $n$ , is therefore  $5726831 - 10000 = 5716831$ . Where the full data set  $D$  has  $5.727e^6$  samples; an average of 9545 vertices per training model. These are randomly selected for each training run, which is repeated 20 times. The individual runs are shown as grey crosses, with the black lines showing the limits and the blue line the mean over the 20 re-runs. The training set size is increased incrementally, by an increment of 100 from 100 to 1000, and an increment of 500 from 1000 to 10000. The converged sample-based errors are:  $\delta_{min} = -59.365\%$ ,  $\delta_{max} = 63.634\%$ ,  $|\bar{\delta}| = 2.767\%$ , and  $\sigma_{|\delta|} = 3.420\%$ .



▲ Figure 4.3: Complex geometry: sample-based training error convergence.

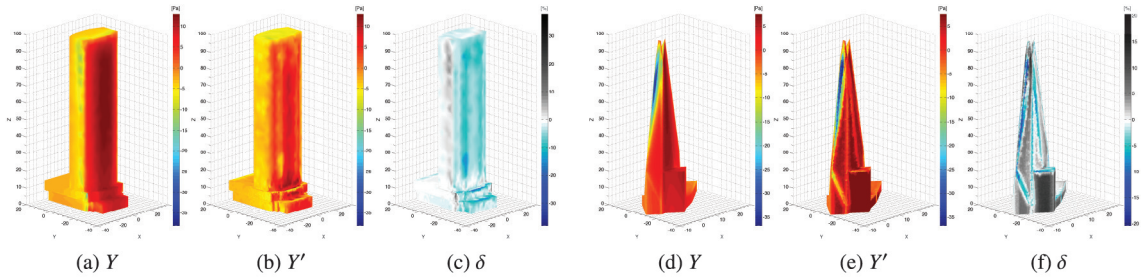
### 4.3. Model-based test

The test set is of mixed geometric complexity which can be seen in the number of vertices on each model,  $m$ , ranging from 528 to 29091. This is a product of the geometric complexity, but, without resampling, also effects the output resolution and speed. Therefore, one of the primary methods for the user to improve the on-line prediction time is to lower the number of vertices on the test model.

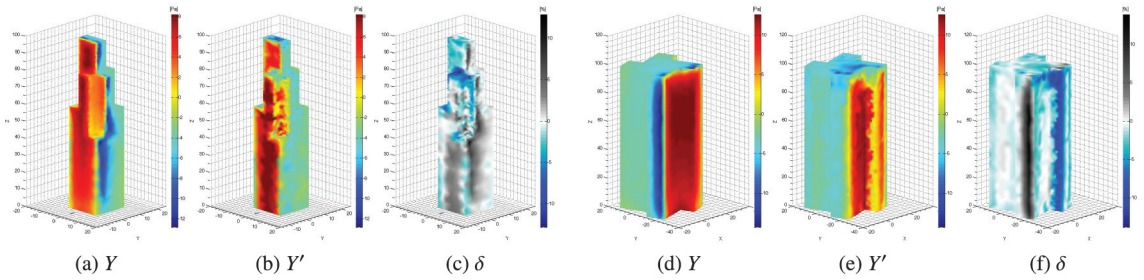
A noticeable trend in these 10 test cases is of under-prediction of negative pressure values, especially in localised regions of very low pressure. To be clear, an under-prediction of a negative value means the prediction is too high, e.g. simulated value is -10 but predicted value is -5.

► Table 4.1:  
Error and time  
results  
summary -  
complex  
geometry:  
model-based.

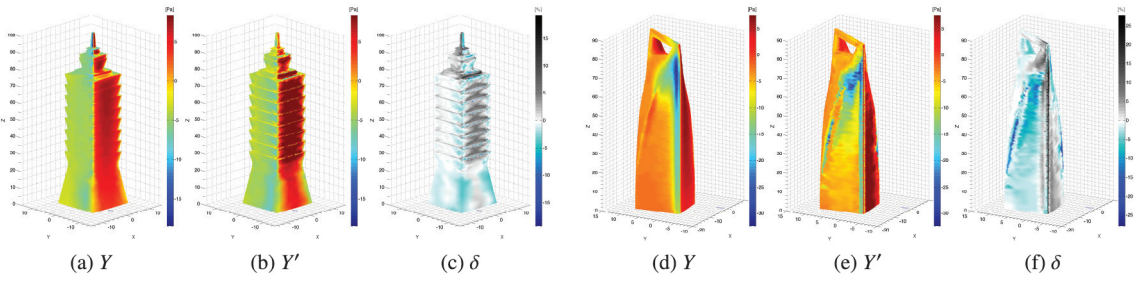
	Case	$m$	Error, $\delta$ [%]				Test time, $t$ [s]		
			$\delta_{min.}$	$\delta_{max.}$	$ \bar{\delta} $	$\sigma_{ \delta }$	$X$	$Y'$	$X + Y'$
1	Met Life	11439	-25.018	33.058	3.629	5.088	318.462	0.0468	318.509
2	Shard	20145	-21.899	13.849	2.392	2.498	560.837	0.0680	560.905
3	Sears	1629	-16.025	9.662	2.467	2.838	45.351	0.0230	45.374
4	Euston	1981	-19.770	12.314	3.669	4.059	55.151	0.0238	55.175
5	Taipei101	29091	-17.352	15.027	2.492	2.871	809.893	0.0898	809.983
6	Shanghai	10469	-23.578	25.268	4.440	4.840	291.457	0.0445	291.501
7	BankOfChina	528	-13.702	4.929	2.237	2.479	14.700	0.0203	14.720
8	Exchange	3931	-23.616	23.699	3.929	4.451	109.439	0.0286	109.468
9	Frankfurter	3688	-16.617	13.299	1.994	2.096	102.674	0.0280	102.702
10	Washington	1837	-19.099	13.380	2.766	2.225	51.142	0.0235	51.166



▲ Figure 4.4: Model-based test: (left) Metlife; (right) Shard.

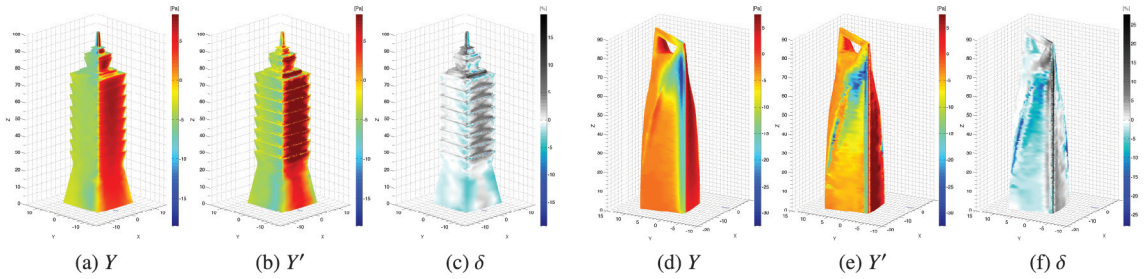


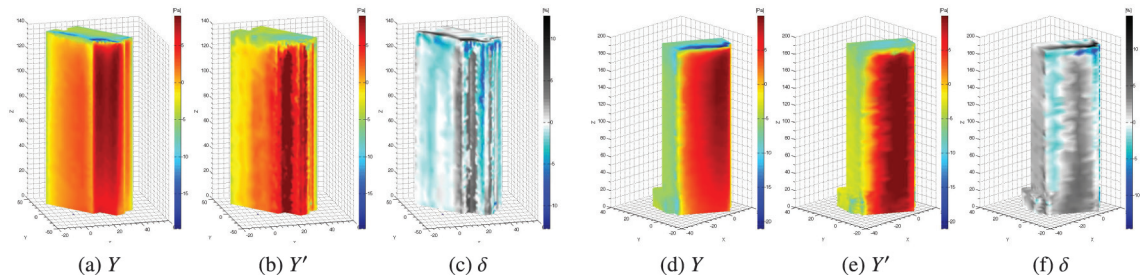
▲ Figure 4.5: Model-based test: (left) Sears; (right) Euston.



▲ Figure 4.6: Model-based test: (left) Taipei; (right) Shanghai.

▼ Figure 4.7: Model-based test: (left) BankOfChina; (right) Exchange.





▲ Figure 4.8: Model-based test: (left) Frankfurter; (right) Washington.

## 5. DISCUSSION

These developments represent an alternative approach that is fundamentally different to previous attempts at generalising tall building aerodynamics found in the literature. The use of local features rather than global parameters allows for arbitrary complexity in the model and for vertex surface pressure visualisation rather than global factors.

Compared to solver approximation techniques, such as the FFD solver, solution approximation has the benefit of being based on a conventional, higher accuracy CFD solver. As such, the validity of the basis data can, to a larger extent, be trusted or verified. The comparative disadvantage is that the FFD can produce field rather than surface data which is useful for identifying flow patterns, assessing pedestrian comfort, and to gauge the secondary downstream effects that a new building will have on others.

A sensitivity analysis was run on the training set size, and found that a sample size of  $n=10000$  was adequate to reach error convergence during the ANN training. The training set is therefore only 0.175% of the full available data set  $D=5726831$  from the 600 evaluated training models. Although this does not mean that only one training simulation is required (the average number of vertices on a training model was 9545), evaluating 600 models may be excessive. Following this, the final model-based test was visualised to check the predicted pressure distribution qualitatively against the simulation. Generally, under-prediction of negative pressures can be seen, but general patterning or distribution of both positive and negative pressure remains intact.

### 5.1. Process time analysis

The feature extraction times are based on a calculation speed of  $0.02784s/sample$  (about 36 samples/s) for off-line ROM generation ( $n=10000$ ) and on-line predictions. The ANN training time, for  $n=10000$ , is averaged over 20 runs; the mean time is 38.269s ( $\sigma:17.143s$ ).

The model-based prediction times show that, in comparing only on-line processes, the ROM is 5.39-times faster than the conventional CFD method. However, this does not take into account the full process. By comparing the off-line plus on-line processes for repetition, where  $x$  is the

number of design iterations, the CFD time=1383.162x and the ROM time=256.482x+1145905.17 (Figure 5.1). In solving for x, the minimum number of iterations before the full ROM process time equals the CFD is x=1017.

5.2. Limitations

A key characteristic of the solution approximation approach is a reduction, where the full field data available through CFD simulation is reduced to the surface data of interest. As such, no direct information about the surrounding wind environment is conveyed as it would be with the full CFD. For certain cases this information is valuable; for instance, pedestrian comfort studies require data on the wind velocity at a horizontal plane above ground level in order to measure the change that a new design can have on an urban environment. The FFD is primarily focused on this field data with surface interactions as a secondary consideration; for the FFD the structured meshes give a poor surface representation and makes direct comparisons with CFD difficult.

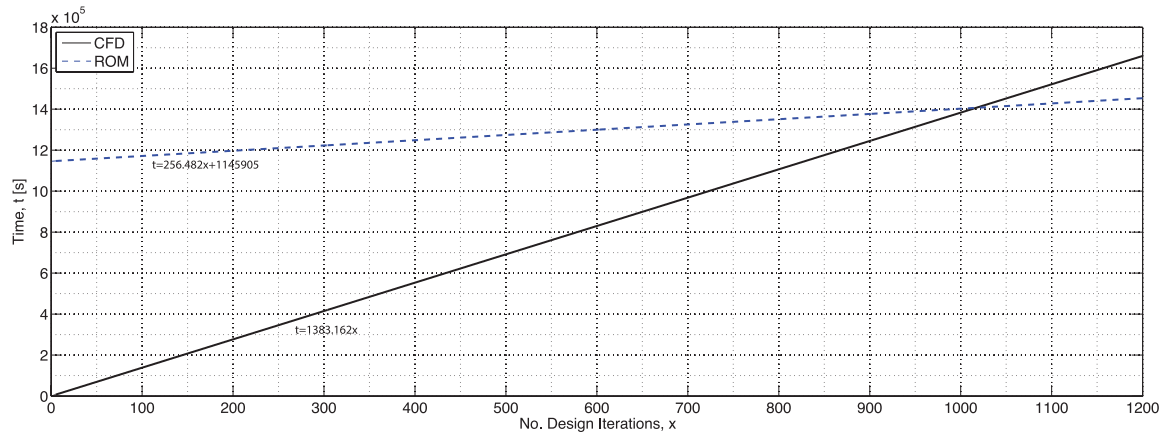
Steady-state RANS CFD was used, therefore the prediction results are not time-dependent peaks but timeaveraged; a common approximation strategy in practice. This is a relatively significant simplification in structural engineering terms, but necessary to allow simulation of a large training set. In later project stages it is of concern to establish quantifiable peak values for the design, for which more accurate in-depth analyses are conducted, i.e. wind-tunnel or LES.

The algorithm calculating the training and test shape features is not currently optimised for efficiency. The most costly part of the calculation lies in the local curvature analysis, where for every vertex the neighbouring vertices must be found and ordered by proximity. Consequently, the reported tests use a feature generation time of 0.02784s/sample (about 36 samples per second) which is open for improvement. Similarly for the input

▼ Table 5.1: CFD and ROM process times.

	Time [s]			
	Mean	$\sigma$	Min.	Max.
<b>Conventional CFD Process</b>				
Test simulation <sup>a</sup>	<b>1383.162</b>	1254.994	334.937	4103.020
<b>Off-line ROM Process</b>				
Training simulations <sup>c</sup> (600)	1145905.17	63046.37	941575.2	1220212.36
Feature extraction <sup>b</sup> (n=10000)	278.4	-	-	-
ANN training	38.269	17.143	19.948	62.165
<b>Total</b>	<b>1146221.84</b>			
<b>On-line ROM Process</b>				
Feature calculation	256.440	272.215	14.700	809.893
Prediction	0.0414	0.0238	0.0203	0.0898
<b>Total</b>	<b>256.482</b>			

<sup>a</sup> Test simulation  $t=0.132m+265.283$  ( $r^2=0.786$ ); <sup>b</sup> Feature calculation speed = 0.02784s/sample; <sup>c</sup> Training simulation individual mean  $t = 1914.470s$  ( $\sigma$ :629.808s, range:1034.209–3759.655s).



▲ Figure 5.1: Process time  $t$  against number of design iterations  $x$  for CFD and ROM.

feature vector; there is no guarantee that the selection used is necessarily optimal. The local curvature analysis was calculated over five neighbourhood rings per vertex since the computational effort escalates quickly with the number of neighbourhoods. For the shape features, the optimal selection likely varies between geometry types and complexity, i.e. a simpler model would require less features to make an acceptable prediction.

## 6. CONCLUSION

In summary, the methodology and results presented here demonstrate an alternative approach to approximating tall building wind pressure for generative, early stage design. The results indicate that significant improvements in response time (5.4-times faster when comparing on-line prediction times with conventional CFD) can be made with a reasonable trade-off in accuracy (mean absolute errors of 1.99–4.44%  $\sigma$ : 2.10–5.09%). Although the off-line time is substantial, requiring around 1000 predictions before the process time becomes an improvement on the traditional CFD approach, there are three conditions that mediate this limitation: firstly, the current feature calculation time is not optimised and an improvement on the current speed can easily be attained through more efficient code; secondly, the converged training set sample size of 10000 suggests that 600 training simulations is in fact too high; and thirdly, the off-line process time is inherently a one-off component as compared to the potential infinite number of on-line predictions it enables.

The most promising aspect of the approach is its applicability to higher-order basis CFD. The basic invariance of the on-line prediction time and accuracy to the basis simulation means that benefits increase with cost of the conventional simulation.



## ACKNOWLEDGEMENTS

This research was sponsored by the EPSRC, Bentley Systems and PLP Architects.

## REFERENCES

1. Stam, J., 1999. Stable Fluids. Tech. rep., Alias-Wavefront, Seattle.
2. Chronis, A., Tsigkari, M., Davis, A. and Aish, F., 2012. Design Systems, Ecology and Time. In *Proceedings of ACADIA 12: Synthetic Digital Ecologies*. San Francisco, CA.
3. Chronis, A., Turner, A. and Tsigkari, M., 2011. Generative Fluid Dynamics: Integration of Fast Fluid Dynamics and Genetic Algorithms for Wind Loading Optimization of a Free Form Surface. In *Proceedings of SimAUD SCS SpringSim'11*, 79–86.
4. Karagkouni, C.S., Fatah, A., Tsigkari, M. and Chronis, A., 2013. Facade Apertures Optimization: Integrating Cross-Ventilation Performance Analysis in Fluid Dynamics Simulation. In *Proceedings of SimAUD SCS SpringSim'13*, Malkawi 2004.
5. Athanailidi, P., Fatah gen Schieck, A., Tenu, V. and Chronis, A., 2014. Tensegrity Systems Acting as Windbreak: Form Finding and Fast Fluid Dynamics Analysis to Address Wind Funnel Effect. In *Proceedings of SimAUD SCS SpringSim'14*, 127–134. Tampa, FL.
6. Zuo, W. and Chen, Q., 2009. Real-Time or Faster-than-Real-Time Simulation of Airflow in Buildings. *Indoor Air*, 19(1), 33–44.
7. Zuo, W. and Chen, Q., 2010. Fast and Informative Flow Simulations in a Building by using Fast Fluid Dynamics Model on Graphics Processing Unit. *Building and Environment*, 45(3), 747–757.
8. Graening, L. and Sendhoj, B., 2014. Shape Mining: A Holistic Data Mining Approach for Engineering Design. *Advanced Engineering Informatics*.
9. Graening, L., Menzel, S., Hasenjaeger, M., Bihrer, T., Olhofer, M. and Sendhoj, B., 2008. Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries. *Journal of Mathematical Modelling and Algorithms*, 7(4), 329–350.
10. Degroote, J., Vierendeels, J. and Willcox, K., 2010. Interpolation Among Reduced-Order Matrices to Obtain Parameterized Models for Design, Optimization and Probabilistic Analysis. *International Journal for Numerical Methods in Fluids*, 2010(63), 207–230.
11. Bui-Thanh, T., Willcox, K. and Ghattas, O., 2008. Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications. *AIAA*, 46(10).
12. Schilders, W., 2008. Introduction to Model Order Reduction. In *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, 13 edn.
13. Rendall, T.C.S. and Allen, C.B., 2008. Multi-Dimensional Aircraft Surface Pressure Interpolation using Radial Basis Functions. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 222(4), 483–495.
14. Srinivasan, R.S. and Malkawi, A.M., 2004. The Use of Learning Algorithms for Real-Time Immersive Data Visualisation in Buildings. In *SIGRAI*, 329–332.
15. Cipriano, G., Phillips, G.N. and Gleicher, M., 2009. Multi-Scale Surface Descriptors. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1201–8.
16. Feng, X., Xia, K., Chen, Z., Tong, Y. and Wei, G.W., 2013. Multiscale geometric modeling of macromolecules II: Lagrangian representation. *Journal of computational chemistry*, 34(24), 2100–20.

17. Darom, T. and Keller, Y., 2012. Scale-Invariant Features for 3-D Mesh Models. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 21(5), 2758–69.
18. Jiao, X. and Heath, M.T., 2002. Feature Detection for Surface Meshes. In *Proceedings of 8th international conference on numerical grid generation in computational field simulations*. University of Illinois.
19. Hubeli, A., Meyer, K. and Gross, M., 2000. Mesh Edge Detection. Tech. rep., Swiss Federal Institute of Technology, Zurich.
20. Hubeli, A. and Gross, M., 2001. Multiresolution Feature Extraction for Unstructured Meshes. In *Proceedings of the conference on Visualization'01*. IEEE Computer Society, 287–294.
21. Szilvasi-Nagy, M., 2006. About Curvatures on Triangle Meshes. In *KoG 10 - Journal of Croatian Society for Geometry and Graphics*, 13–18.
22. Dong, C.S. and Wang, G.Z., 2005. Curvatures Estimation on Triangular Mesh. *Journal of Zhejiang University SCIENCE*, 6(Suppl. 1), 128–136.
23. Walter, N., Laligant, O. and Aubreton, O., 2008. Salient Point SUSAN 3D Operator for Triangles Meshes. In *The 2nd International Topical Meeting on Optical Sensing and Artificial Vision*.
24. Bentley Systems, 2013. GenerativeComponents v08.11.08.296. [www.bentley.com](http://www.bentley.com).
25. Park, S.M., Elnimeiri, M., Sharpe, D.C. and Krawczyk, R.J., 2004. Tall Building Form Generation by Parametric Design Process. In *CTBUH*, 1–7. Seoul Conference.
26. Samareh, J.A., 2001. Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization. *American Institute of Aeronautics and Astronautics*, 39(5), 333–343.
27. ANSYS, 2014. CFX v13.0. [www.ansys.com](http://www.ansys.com).
28. ANSYS, 2009. CFX-Solver Theory Guide. Tech. rep., ANSYS Inc., Canonsburg, PA.
29. Hsu, S.A., Meindl, E.A. and Gilhousen, D.B., 1994. Determining the Power-law Wind-Profile Exponent under Near-Neutral Stability Conditions at Sea. *Journal of Applied Meteorology*, 33(1994), 757–772.
30. Turk, G. and Levoy, M., 1994. The Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep/>.
31. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T. and Alkon, D.L., 1988. Accelerating the Convergence of the Back-propagation Method. *Biological Cybernetics*, 59(4-5), 257–263.
32. Breiman, L., 2001. Random Forests. *Machine Learning*, 2001(45), 5–32.
33. Matlab. TreeBagger algorithm. <http://www.mathworks.co.uk/help/toolbox/stats/treebaggerclass.html>

## Samuel Wilkinson and Sean Hanna

University College London  
 Bartlett School of Graduate Studies  
 132 Hampstead Road  
 S. Wilkinson, [ucftsmw@ucl.ac.uk](mailto:ucftsmw@ucl.ac.uk)



