Detecting discontinuity orientation and location on two-dimensional grids using neural networks

Pierre Vuillecard

June 10, 2020

Abstract

In a recent paper [1], a neural network is used to detect discontinuities in the numerical solutions of two-dimensional conservation laws. Based on this troubled-cells detector, the present work aims to construct a neural network to predict the orientation and the location of the shock in each troubled-cell. Several numerical results of such a neural network are presented to demonstrate the performance and good generalization properties towards more general problems and grids.

1 Introduction

Solution of (non-linear) hyperbolic conservation laws are known to be inherently discontinuous. As a result, the approximate solution obtained with high-order numerical methods typically suffers from Gibbs oscillations, which can lead to numerical instabilities. Several strategies have been proposed during the past few decades to control these spurious oscillations. One of them consists to limit the numerical solution. To ensure accuracy and efficiency, the limiting procedure has to be applied only in the troubled-cells, i.e., the cells in which the solution loses regularity. Different approaches have been proposed to detect them. The method recently developed in [1] makes use of a neural network, and has the advantage of being free of problem-dependent, user-tunable parameters. Although the approach presented in [1] gives satisfactory results, a few improvements can be considered. On one hand, one could use directional limiters to limit the solution only in the direction orthogonal to the discontinuity. On the other hand, one could consider anisotropic mesh-adaptation algorithms, mainly designed to refine the grid in the direction of the shock. It is evident that both strategies rely not only on the identification of the troubled cells, but also on discontinuity orientation and location within a given cell.

In the same spirit of [1], we want to use neural networks. In particular, this writing is focused on the development of a neural network that can predict discontinuities orientations and locations in troubled-cells on a two-dimensional mesh. The underlying physical problem is a conservation law, discretized using a Discontinuous Galerkin method of degree p. For the purposes of this work, the time-integration scheme is not relevant, as the prediction is done offline. As labeled data can be constructed easily, a good neural network model is the multilayer perceptron, trained with supervised learning. Details on neural network architectures can be found, e.g., in [2]. Moreover the neural network has to be computationally efficient, mesh independent and be able to handle different orders p, with particular effort for $p \geq 2$.

The rest of this paper is decomposed in two major parts. The first one is focused on discontinuity orientation and the second one on discontinuity location. Both parts are structured as follow: a first section that introduces the dataset and all the settings of the problem. The subsequent section presents different approaches that have been done to train the neural network with some preliminary results. After a selection of the best neural network, evoked in the next section, the best one is used to show the final results in a last section. It comprises applications of the model with different polynomial orders and different meshes in order to assess mesh dependency, accuracy and robustness of the neural networks. Finally, concluding remarks are made at the end of the report.

2 Discontinuity orientation

In this section we describe a possible method to predict the orientation, i.e., the direction of a discontinuity.

2.1 The dataset

This section presents how the dataset was generated and all the possible datasets. A supervised learning approach is used to learn the function that maps a troubled-cell to an orientation discontinuity. First of all we have to create a discontinuity in a mesh, whose orientation is known a priori. Simple functions are:

$$f(x, y) = a \times (y < tan(\alpha) \times x) + b \times (y \geq tan(\alpha) \times x) \tag{1}$$

$$f(x, y) = a \times (|y| < \sqrt{0.5 - x^2}) + b \times (|y| \geq \sqrt{0.5 - x^2}) \tag{2}$$

Here α is the angle in radians of the discontinuity orientation, while a and b are used to determine the strength of the shock. In order to generate a complete dataset the angle α was sampled from a uniform distribution $U(0, \pi)$. Similarly a and b came from a uniform $U(-2, 2)$. The equation (1) creates a linear discontinuity, whereas equation (2) creates a circular discontinuity, thus it generates good test set because a circular discontinuity represents every angle. To detect the troubled-cells in the mesh, the troubled-cells detector presented in paper [1] is used.

Secondly, the neural network needs to be mesh independent. As in many discretization algorithms, we construct a reference triangle. In paper [1] a method is described to transform a generic triangle in a fixed right-angled triangle as figure 1 shows.

In fact a point x in the physical space can be mapped to a reference space such that :

$$x = A\hat{x} + d \text{ , with } A = \begin{pmatrix} \frac{x_b - x_a}{2} & \frac{x_c - x_a}{2} \\ \frac{y_b - y_a}{2} & \frac{y_c - y_b}{2} \end{pmatrix} \tag{3}$$

This affine transformation creates a bijection between physical and reference space. In fact the matrix A is always invertible because $det(A) = \frac{1}{2}det(\overrightarrow{ab}, \overrightarrow{ac})$ corresponds to the surface of the triangle abc which is never zero. Furthermore it can be shown that there is a relation between the angle in the physical space and the reference space :

$$\begin{pmatrix} cos\alpha \\ sin\alpha \end{pmatrix} = A \begin{pmatrix} cos\hat{\alpha} \\ sin\hat{\alpha} \end{pmatrix} \Leftrightarrow \alpha = atan(\frac{A_{21}cos\hat{\alpha} + A_{22}sin\hat{\alpha}}{A_{11}cos\hat{\alpha} + A_{12}sin\hat{\alpha}}) \tag{4}$$

Therefore, for any mesh, the dataset will only contain data in the reference space with a discontinuity of angle $\hat{\alpha}$. We let the neural network predict the angle in the reference space. Then, it can be mapped in the physical space thanks to the matrix A. Note that we are implicitly exploiting the spatial locality of the Discontinuous Galerkin method. Indeed, all triangles are treated independently.

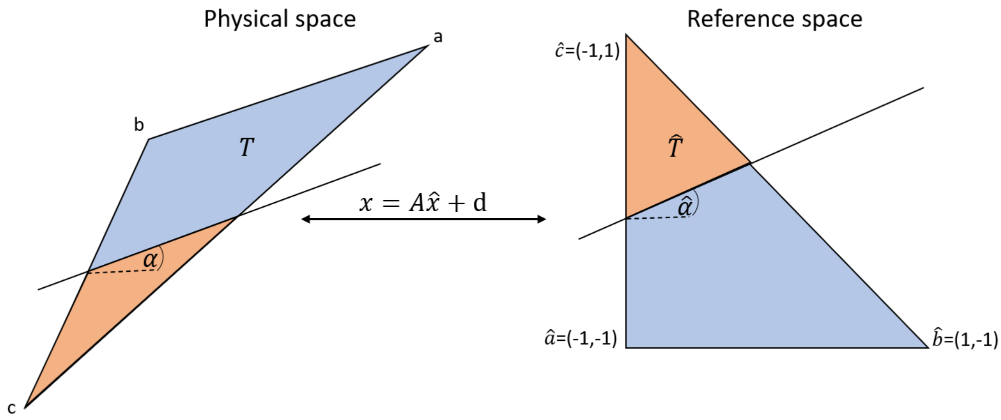


Figure 1: Affine mapping between triangle $T = abc$ in the physical space and a triangle $\hat{T} = \hat{a}\hat{b}\hat{c}$ in the reference space. The blue and red part of the triangle show the step of the discontinuity where for example red equals a and blue equals b in equation (1) and (2). Moreover there is also the discontinuity angle α in the physical space and the angle $\hat{\alpha}$ in the reference space.

Thirdly to get more information about the discontinuity in a troubled-cells, three different settings are possible as inputs. The first one consists in keeping only the information about the troubled-cells T_0, see figure 2, the input dimension is then $D = \frac{(p+1)(p+2)}{2}$. The second setting consists in taking the solution values in T_0 and the values of the neighbours that are *close* to T_0, i.e, adding neighboring information by means of the solution values in the points that share an edge with T_0, thus the input dimension is $D = \frac{(p+1)(p+2)}{2} + 3(p+1)$. Finally one can also take the information of T_0 and the information of *all* its neighbours T_1,T_2 and T_3, the input dimension is then $D = 2(p+1)(p+2)$.

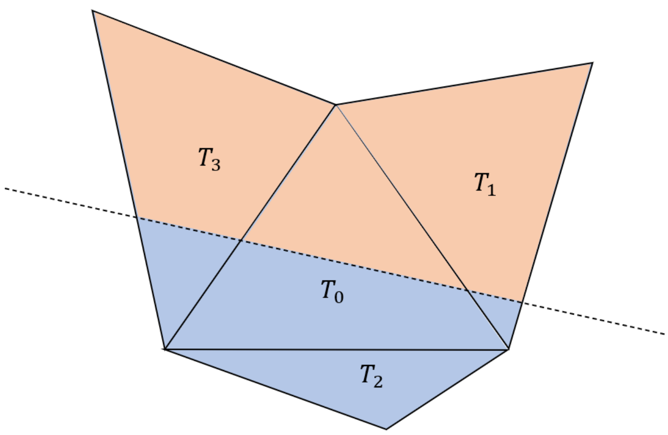
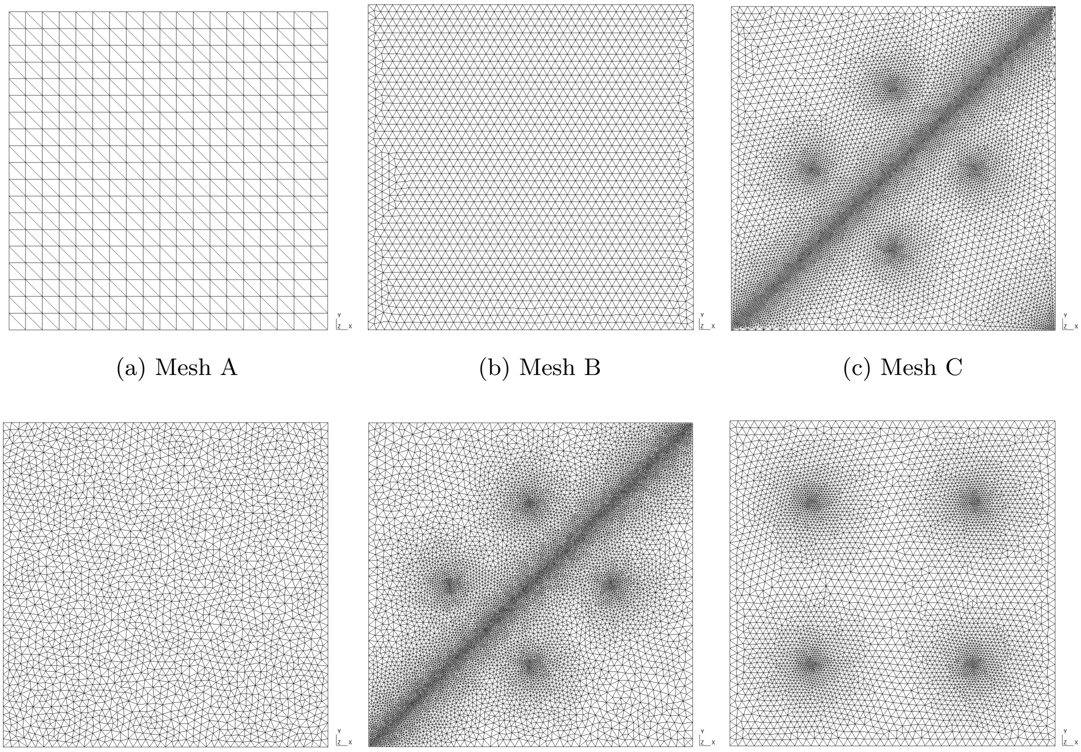


Figure 2: Situation of a troubled-cell T_0 in the physical space. T_1,T_2 and T_3 are the neighbours of a troubled-cell in a mesh. The dash line represents the discontinuity.

Furthermore, the input needs to be normalized before being fed into the network, to ensure faster convergence during training and improve the neural networks ability to generalize [2]. This is achieved by re-scaling the values of the observations between $[-1, 1]$. The following formula is used:



(a) Mesh A (b) Mesh B (c) Mesh C

(d) Mesh B unstr (e) Mesh C unstr (f) Mesh D

Figure 3: Presentation of the meshes used in this reports. Mesh A, B, C and D are *structured* meshes , i.e., meshes that impose periodicity constraints at boundaries. Mesh B unstr, C unstr are *unstructured* meshes. Mesh D is here only to test the neural network. Indeed mesh D could be seen as a combination of mesh B unstr and mesh C unstr.

$$normalization(\mathbf{X}) = \begin{cases} \frac{\mathbf{x}_i}{max(|\mathbf{x}_i|)} & \text{if } \max(|\mathbf{x}_i|) > 1 \\ \mathbf{x}_i & \text{otherwise} \end{cases} \quad \text{i=1,...,N}$$

Here $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{N \times D}$, and $\mathbf{x}_i \in \mathbb{R}^D$ represents the information of a single troubled-cell, N is the total number of troubled-cells in the dataset and D is the input dimension.

Moreover to check if the neural network generalizes well while changing the mesh, two different types of mesh will be used. Either *structured* if a periodicity is imposed in the generation of the mesh in order to get periodic boundary conditions or *unstructured* if there is no condition in the generation of the mesh, see figure 3. Mesh D was created in order to test the neural network. Indeed mesh D could be seen as a combination of mesh B unstr and mesh C unstr. The bases of different orders p are also tested from 2 to 5.

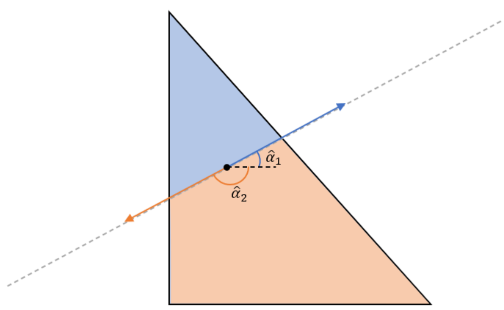
2.2 Different approaches

The training set is an important part in supervised learning because the neural network will learn to predict the discontinuity orientation from the training set. This section presents different tried approaches to train a neural network and get the best prediction of the angle. To highlight the results and the improvements of various approaches we will consider a simple discontinuity with $a = 1$ and $b = 0$ in equation (1). All datasets have

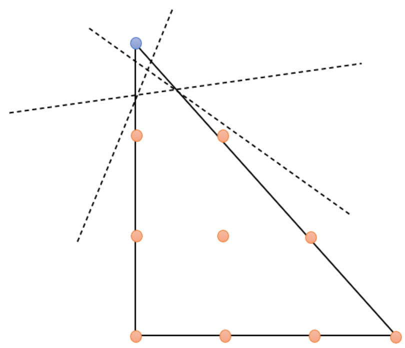
10 000 samples and they're composed of 60% train to compute the gradient and update the weight of the network, 20% validation to control overfitting with an early stopping [2] and 20% for the test set in order to test the network on unseen data. The neural network is composed of 3 hidden layers of dimension equal to the size of the input, a sigmoid activation function and linear output function. The inputs correspond to troubled-cells either with *no* neighbour, with *close* neighbours or with *all* its neighbours as presented in section 2.1. The output is the angle in the reference element. Moreover the order of the base is fixed to p=3 and the model is tested for mesh A, B, and C, see figure 3.

2.2.1 First approach

The simplest idea was to predict the angle from $-\pi$ to π. The loss function used in the neural network was the mean square error between the target and the predicted results . The results of this first approach are shown in table 1. First the neural network does not accurately predict the angle in mesh B and C whereas it shows better performances on mesh A. The mean square error is 2 times smaller on mesh A than on mesh B and C when *no* neighboring data are given. After further investigations, the problem is that for one input there exists two outputs as showed in figure 4a. In fact for one input there are two admissible angles with a difference of π. This can be seen in table 1 where the max of the absolute error is very high and it shouldn't be bigger than $\frac{\pi}{2}$. Thus the neural network is not able to predict the angle well. This effect is less visible in mesh A because by construction the angles are mainly between 0 and π. In table 1 for mesh A, the results improve a lot when *all* neighboring data are given. This is because of the inputs as figure 4b shows. For one input, especially when it passes through a corner of a triangle, there are multiple discontinuities with different orientations due to the discretization. This leads to poor prediction . But if we add information about the neighbours of the troubled-cells, it reduces the impact of this phenomenon. Thus the predictions of the neural network are better, here nearly 7 times better if we compare the loss without neighbour and with *all* neighbours in mesh A in table 1.



(a)



(b)

Figure 4: (a) This is a troubled-cell in the reference space, here for the discontinuity in dash line there are two possibles angles which are $\hat{\alpha}_1$ and $\hat{\alpha}_2$. They both encode the same discontinuity orientation.(b) This is a troubled-cell in the reference space, with order $p = 3$ which corresponds to a discretization with 10 points, blue point corresponds to a value of a and the red of b in the equation (1). For one input, there are degrees of freedoms for the discontinuities orientations

Mesh		A			B			C		
Inputs		no	close	all	no	close	all	no	close	all
loss test		1.36	1.16	0.20	2.74	2.74	2.57	2.75	2.84	2.73
absolute error in degree	mean	42	38	9.7	88	88	85	87	89	86
	std	51	48	24	33	34	32	38	37	38
	median	16	13	4	88	88	86	85	87	83
	min	1e-2	5e-3	2e-2	1e-1	4e-1	4e-1	2e-2	2e-1	1.15
	max	225	231	313	236	197	248	207	200	346
	10 accuracy	39	42	82	1.95	2.05	1.15	2.2	2.1	0.75

Table 1: Results of the first approach. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles are in degree. These are statistics summaries and 10 accuracy corresponds to the percentage of errors that are less than 10 degree . Meshes are those from figure 3. Loss test is the mean square error on the test set. The different inputs *no* neighbour, *close* neighbours and *all* neighbours are explained in section 2.1.

2.2.2 A Second approach

The second approach tries to correct the problem encountered in the first one. In order to encode one orientation of the discontinuity with only one angle, we can train the network with angles between 0 and π. With this technique, the range of possible orientations are covered. Indeed, as an example, if we consider a discontinuity as in figure 4a, the neural network has never seen the angle $\hat{\alpha}_2$ because it was not in the range of the training set. Thus the neural network will naturally predict $\hat{\alpha}_1$. Results of the second approach can be found in table 2 which shows that there is a real improvement compare to the previous approach especially for mesh B and C. Here the results confirm that when *all* neighbours of the troubled-cells are taken into account, the predictions are significantly better. However the maximum of the absolute error is quite high. After further investigations the problem of the first approach partially persists. In fact an horizontal discontinuity is either encoded as 0 or π angle thus when the neural network encounters an horizontal discontinuity it makes a poor prediction.

Mesh		A			B			C		
Inputs		no	close	all	no	close	all	no	close	all
Loss test		0.27	0.24	0.048	0.29	0.29	0.09	0.39	0.37	0.14
absolute error	mean	19	18	5.1	20	20	7.1	22	22	9.4
	std	22	21	11	23	23	15	25	26	20
	median	11	10	3	11	11	3.6	11	11	4.1
	min	3e-4	3e-4	9e-5	1e-3	1e-2	3e-3	3e-4	5e-3	3e-4
	max	105	106	176	106	105	127	111	105	169
	10 accuracy	48.6	50.8	92.5	46.1	46.5	87.6	47.4	47.0	82.6

Table 2: Results of the second approach. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and 10 accuracy corresponds to the percentage of errors that are less than 10 degree . Meshes are those in figure 3. Loss test is the mean square error on the test set. The different inputs *no* neighbour, *close* neighbours and *all* neighbours are explained in section 2.1

2.2.3 A Third approach

Finally one way to fix this problem is to redefine the loss function. In fact for a discontinuity as in figure 4a both angles are correct to describe the discontinuity orientation. However with the standard mean square error, if the true angle is $\hat{\alpha}_1$ and the neural network predicts $\hat{\alpha}_2$, it will penalize it hardly resulting in poor prediction on a test set. This is why, redefining a loss such that both $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are not penalized. The following loss is proposed :

$$Loss(y_{prediction}, y_{true}) = min \begin{cases} (y_{true} - y_{prediction})^2 \\ (y_{true} - (y_{prediction} - \pi))^2 \\ (y_{true} - (y_{prediction} + \pi))^2 \end{cases} \quad (5)$$

After averaging over all training samples, the obtained loss function can be interpreted as a modified mean square error. The results using this loss can be found in table 3. Here again the improvement is visible in every mesh, for every type of input. The inputs with *all* neighbours outperform other inputs by far, it has a 14 times better loss in mesh A and B and a 5 times better for mesh C compared to *close* inputs. The 10 degree accuracy is quite good for mesh A and B with 93% as well as the maximum error which is 28 and 37 degrees. On mesh C the results are slightly worse, as the mean error and standard deviation are bigger. Thus the 10 degree accuracy is 86% and the maximum is high with 152. However the results are still greater in the third approach than in the second approach. The new loss 5 clearly improves the performance of the neural network.

As a result of the three different approaches, one can conclude that the input with *all* neighbours of the troubled-cells really improves the neural network. Moreover the new loss defined in equations (5), helps the neural network to learn the discontinuity orientation.

Mesh		A			B			C		
Inputs		no	close	all	no	close	all	no	close	all
loss test		0.18	0.11	0.008	0.13	0.13	0.009	0.11	0.09	0.02
absolute error	mean	16	14	3.8	15	15	3.8	13	13	5.4
	std	19	14	3.4	14.6	14.7	3.6	13	12	6
	median	10	9	2.8	10	10	2.8	9	8	3.8
	min	9e-3	1e-2	5e-4	1e-4	2e-3	1e-4	4e-3	3e-2	3e-3
	max	95	71	28	80	83	37	85	83	152
	10 accuracy	50.8	53.9	93.1	49.8	49.9	93.5	52.9	54.6	86.0

Table 3: Results of the third approach. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and 10 accuracy corresponds to the percentage of errors that are less than 10 degree . Meshes are those in figure 3. Loss test is the new loss defined in equation (5) on the test set. The different inputs *no* neighbour, *close* neighbours and *all* neighbours are explained in section 2.1

2.3 Models selections

As concluded in section 2.2 the final model will take as inputs all neighbours of the troubled cells. We will also consider the new loss define in equation (5). In this section, different architectures are tested to select the best hyperparameters, including activation functions and network size. In view of the final applications of our model, the neural network needs

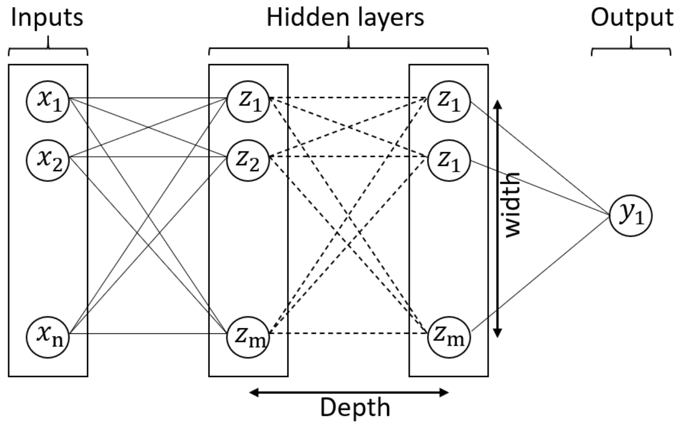


Figure 5: Architecture of a fully connected layers neural network. The depth represents the number of hidden layers and the width which is m corresponds to the number of nodes per hidden layer.

to be computationally efficient, thus a trade-off has to be found between computation time and precision. The different architectures are constructed with different settings such that the activation function ReLu, tanh, sigmoid, as well as different width and depth of the layers, as figure 5 demonstrates. To prevent overfitting and have a better generalisation, dropout and batch normalization layers could be added [2]. Then, in order to compare the neural networks, they had 100 000 samples of data from a mesh with again 60% train, 20% validation and 20% for the test set. The dataset is obtained by computing discontinuities for various α, a and b in equation (1). Moreover each model is trained and tested five times and the results are then averaged to get better statistics. This is because parameter initialization and training are stochastic. In table 4 the performance of these architectures are shown for mesh B with order p=3, similar results are obtained for other meshes. The *base* architecture is a 3 hidden layers neural network with width $m = n$ in figure 5. *Wider* means that the size of the layers is the double of the input size $m = 2 \times n$. *Deeper* means that instead of 3 hidden layers, 4 are taken. *Batch* means that a batch normalisation layers is added between the layers. *Tanh* , *Relu* , *Sigmoid* are activation functions of the hidden layers. The output layer was always a linear activation function. First in table 4, the *base* architecture with *sigmoid* activation function seems to be the best one among the activation functions. If we add *wider* or *deeper* or *batch*, the models are still quite good but since there is no major improvement, also taking into account the computational cost, the best model is the simplest one. In the context of deep networks, the usage of sigmoid might lead to the vanishing gradient problem. Here, we do not experience such a problem. Possible explanations are the relatively small depth or a smart initialization of the parameters. We note that we are not making a thorough study on the network hyperparameters. Thus the best model is a *base* architecture with *sigmoid* activation functions.

Mesh B with 100 000 data		Absolute error in degree					
Architecture	loss test	mean	std	median	min	max	10 accuracy
base+tanh	0.012	4.45	4.47	3.42	5e-4	88.07	91.6%
base+ReLu	0.013	4.56	4.82	3.45	7e-4	88.25	90.8%
base+sigmoid	0.007	3.43	3.12	2.57	9e-5	38.47	95.2%
base+wider+sigmoid	0.007	3.40	3.23	2.47	1e-4	47.64	95.0%
base+deeper+sigmoid	0.007	3.49	3.18	2.63	3e-4	50.07	95.0%
base+batch+sigmoid	0.007	3.50	3.21	2.59	2e-4	39.70	94.7%

Table 4: Results of the models selections. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles are in degree. There are statistics summaries and 10 accuracy corresponds to the percentage of errors that are less than 10 degree. Mesh B is the one in figure 3. loss test is the new loss defined in equation 5 on the test set.

2.4 Results

This section presents the performance of the neural networks selected in section 2.3 with different polynomial degrees on various meshes. In order to compare, the results are obtained using 100000 troubled-cells data. Here again the dataset is decomposed with 60% train, 20% validation and 20% for the test set. Each result is the average of 3 training rounds to obtain better statistics. For each order p from 2 to 5, the neural networks are trained on each mesh and test on the others, Mesh D is used only to test the neural networks. Results can be found in tables 10, 11, 12 and 13, and are reported in the annex 5.

2.4.1 Orders p

As mentioned in section 2.2.1 with figure 4b, due to the discretization of a troubled-cell, there might be cases with same input and different output. That is due to the variation of the discontinuity orientation for a given input. This leads to poor prediction. As expected, increasing the order of the basis reduced the variation of the target value for a given input leading to better prediction. As figure 6 shows, increasing the order clearly improves the results of the models. At first sight, with order p=2 the results, in table 10, are not very accurate because if we test the model on the same mesh as for the train, we obtain a 10 degree accuracy between 70% and 80%. Whereas if we take order p=5 the accuracy of the model is between 98.6% and 91.5%. Thus increasing the order helps the neural networks to make a good prediction.

In figure 13, the prediction of the orientation of a circular discontinuity on mesh C for various orders are shown. As mentioned before, increasing the order p increases the accuracy of the prediction. Here the predictions of the circular discontinuity confirm this. Indeed in the zoomed plot we can see that the orientation predictions for p=5 are very accurate, i.e arrows follow the orientation of the true red discontinuity. Note that for $p = 2$ the orientation accuracy of the 10 degree on mesh C is only 78.5% and the arrows orientation seem acceptable.

2.4.2 Mesh dependency

The method to predict the discontinuity orientation needs to be mesh independent. This part of the report tried to check if the model that was selected in section 2.3 is mesh independent. First, in section 5, the tables show that for every order the neural network

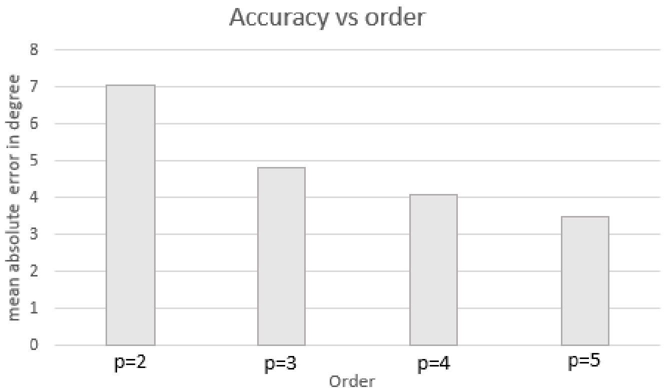


Figure 6: Average of the mean absolute error tested on the same mesh used for training, for each order p.

is slightly less accurate on *unstructured* meshes such as mesh B unstr and mesh C unstr. Moreover if the neural networks is trained on mesh A then the tests on other meshes are bad as shown in figure 7, thus training on mesh A doesn't generalise the neural network well. This can be explained by recalling that mesh A is *structured*, and all possible shock orientations might not be well captured. Furthermore, if the neural network is trained on a *structured* mesh such as mesh B or mesh C then the tests on *unstructured* meshes such as mesh B unstr and mesh C unstr are worse. Mesh B is not very different from mesh A, so this is expected. A less expected result is the one obtained with mesh C. However, one could argue that some symmetry effects are still present, mainly in the construction of the training set. However, if we trained the neural networks with *unstructured* meshes, namely the meshes which are not constructed by enforcing periodicity constraints, then the predictions on *structured* meshes are slightly less accurate. Figure 7 demonstrates that for order p=2 and p=3 training the neural networks on mesh C leads to better generalization. Whereas for order p=4 and p=5 training the neural networks on mesh C unstr leads to slightly better generalization. Moreover, training on mesh B seems to generalize quite well for all orders p but the variance is pretty high compare to others. In fact as stated before the results are very accurate on *structured* meshes and less accurate on *unstructured* meshes.

Therefore, the advantages of mesh B and mesh C unstr can be used by training the neural network on both meshes. Then, we expect that thanks to the training on mesh B the neural network predicts well on *structured* meshes such as C and A and thanks to the training on mesh C unstr the neural network predicts well on *unstructured* mesh such as mesh B unstr. Note that the training set is constructed such that it has again 60 000 samples in order to compare with previous results. The results of such a test are in table 5. The generalization is much better with such training set even from order p=3 where the accuracy for all meshes is above 80% and near 90% for order p=5. Moreover, the results on mesh A are not as good as on the *structured* meshes B and C. Again, this might be due to the peculiar structure of Mesh A.

Mesh		A	B	B unstr	C	C unstr	D
Order p = 2							
Loss		0.123	0.022	0.038	0.028	0.043	0.028
Absolute error	mean	12.3	6.4	8.1	7.0	8.4	7.0
	std	15.8	5.6	7.6	6.5	8.3	6.5
	median	7.0	4.9	6.1	5.3	6.2	5.3
	max	153.7	58.5	121.2	147.4	147.7	120.3
	accuracy	63.3	79.3	70.52	76.57	69.6	77.0
Order p = 3							
Loss		0.072	0.011	0.026	0.031	0.028	0.018
Absolute error	mean	8.2	4.4	6.1	6.0	6.4	5.3
	std	12.9	4.0	6.8	8.0	7.1	5.6
	median	4.4	3.4	4.4	3.9	4.7	3.9
	max	136.6	70.1	140.2	143.2	128.7	138.7
	accuracy	79.9	90.8	81.7	84.4	80.0	86.5
Order p = 4							
Loss		0.050	0.012	0.035	0.023	0.027	0.033
Absolute error	mean	6.3	3.8	6.3	5.0	5.7	5.4
	std	11.1	4.8	8.4	6.8	7.3	8.7
	median	3.3	2.6	4.1	3.3	3.9	3.3
	max	164.8	113.7	137.0	141.1	165.3	162.3
	accuracy	86.5	94.5	82.7	88.7	84.7	88.3
Order p = 5							
Loss		0.037	0.012	0.028	0.023	0.016	0.013
Absolute error	mean	5.2	3.5	5.3	4.5	4.7	3.9
	std	9.6	5.1	7.8	7.2	5.5	5.2
	median	2.8	2.4	3.3	2.8	3.3	2.7
	max	145.3	88.6	145.0	176.5	130.5	161.2
	accuracy	90.6	96.1	87.9	91.1	89.1	93.2

Table 5: Results obtained by training the neural networks with 30000 samples from mesh B and 30000 samples from mesh C unstr. The neural network was the one selected in section 2.3. It was tested for every mesh and every order p. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles are in degree. These are statistics summaries and accuracy corresponds to the percentage of errors that are less than 10 degree. Loss is the new loss defined in equation (5) on the test set.

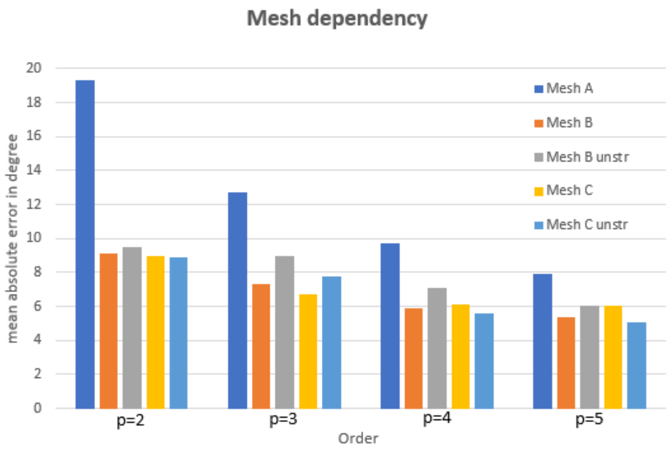


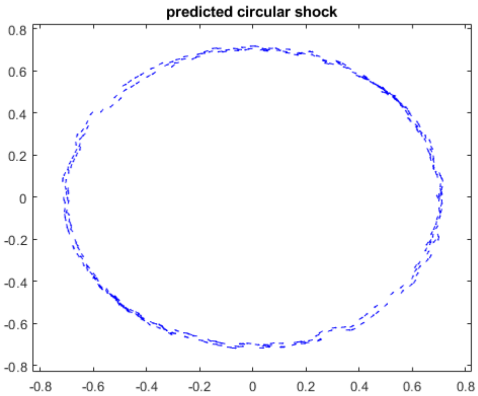
Figure 7: Average of the mean absolute error tested on every mesh except on the training mesh, for each order p. As an example, the blue bar for every order p represents the mean absolute error of the neural network trained on mesh A and tested on the other meshes. The legend indicates the training meshes.

2.4.3 Robustness

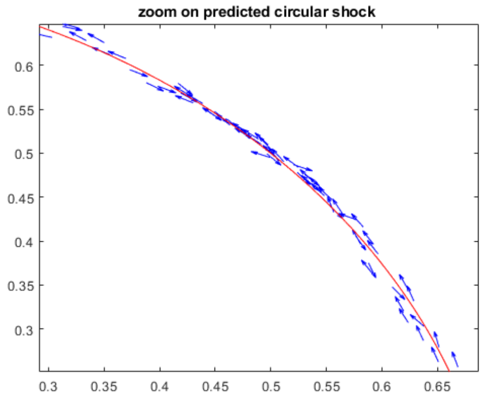
The neural network was designed in a well defined environment, in which training and test sets are assumed to be generated from the same statistical distribution. However, in real applications the input data may not be as *clean*. Numerical (discontinuous) solutions of time-dependent problems might present minor oscillations, that have not been considered in the construction of the training set. This is why we would like to assess the robustness of the neural network against noisy data. We add gaussian noise with various variances σ^2 to the input data of the neural network. The results can be found in table 6. Let us recall that the input data is normalized and that input values are equaled to a and b as in equation (1), i.e, the discontinuity is a simple step function. Results show that with small added noise, with σ=0.01, the neural network performance are almost not affected compared to the original test set. For σ=0.05, the results are somewhat worse and with σ=0.1, the results are worse but still acceptable, especially for order p from 3 to 5. Finally, we could expect that if the neural network is trained with more realistic data, i.e noisy data, the neural network could have better results on unseen noisy data as the one in table 6.

σ		none	0.01	0.05	0.1
Order p=2					
Loss		0.019	0.023	0.048	0.091
Absolute error	mean	6.0	6.2	8.1	11.0
	std	5.2	5.9	9.4	13.2
	median	4.6	4.6	5.7	7.191
	max	57.7	88.8	163.6	188.5
	accuracy	81.2	80.6	73.0	62.8
Order p=3					
Loss		0.007	0.007	0.014	0.028
Absolute error	mean	3.5	3.5	4.4	6.1
	std	3.2	3.2	5.0	7.3
	median	2.6	2.6	3.1	4.1
	max	41.2	56.2	166.6	160.6
	accuracy	94.8	94.6	90.7	82.1
Order p=4					
Loss		0.010	0.012	0.026	0.049
Absolute error	mean	3.2	3.3	4.5	6.3
	std	4.6	5.1	8.0	11.0
	median	2.2	2.2	2.6	3.3
	max	103.4	107.8	145.6	171.1
	accuracy	96.3	96.1	91.8	85.3
Order p=5					
Loss		0.004	0.004	0.009	0.020
Absolute error	mean	2.3	2.4	3.2	4.7
	std	2.4	2.7	4.4	6.4
	median	1.7	1.7	2.2	2.9
	max	68.6	83.3	89.4	88.9
	accuracy	99.0	98.8	95.7	89.3

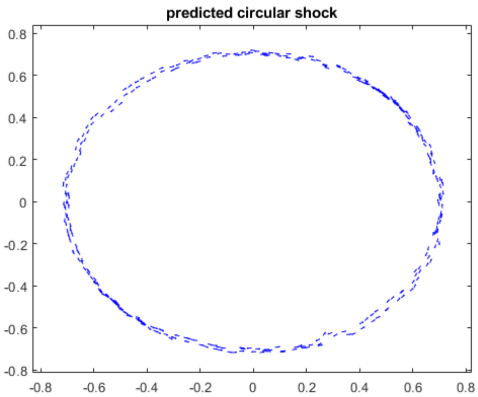
Table 6: Robustness results obtained by training the neural networks with 60000 samples from mesh B and tested on a test set from mesh B with different added noise. σ is the standard deviation of the gaussian noise added to the test set (none means that there is no noise added).The neural network was the one selected in section 2.3. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles are in degree. These are statistics summaries and accuracy corresponds to the percentage of errors that are less than 10 degree. Loss is the new loss defined in equation (5) on the test set.



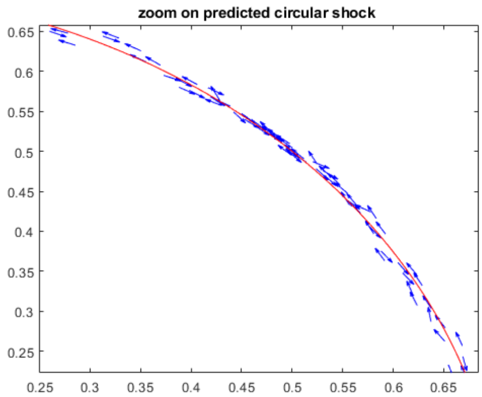
(a) Mesh C p=2



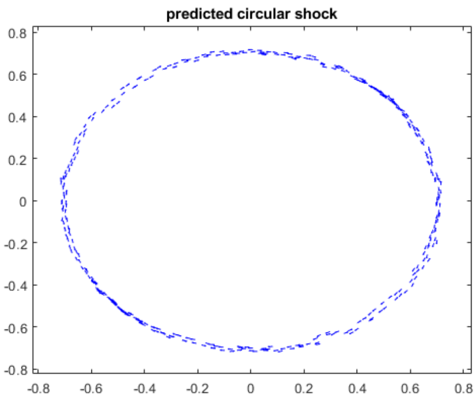
(b) Mesh C p=2



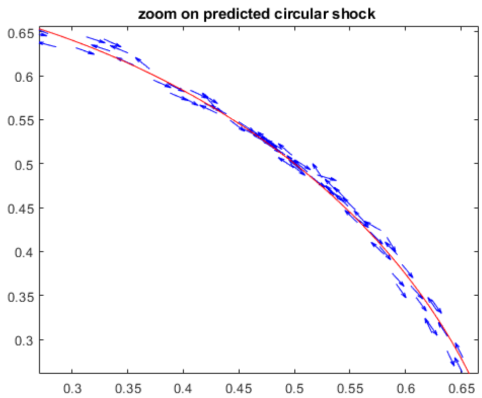
(c) Mesh C p=3



(d) Mesh C p=3



(e) Mesh C p=5



(f) Mesh C p=5

Figure 8: Prediction of the location and orientation of a circular discontinuity as described in equation (2). The prediction is done on mesh C with various order p. In the zoom plots, the red line represents the true circular discontinuity.

3 Discontinuity location

In this section, we keep the spirit of the first part to describe a technique to detect also the location of the discontinuity.

3.1 The dataset

One of the best way to represent the location of a discontinuity in a triangle is to compute the orthogonal vector from the discontinuity to a fix point in a triangle. This allows us to compute both the location and the direction of the shock, simply by taking the magnitude and the angle of the orthogonal vector. Here again, as, the orientation of the discontinuity, the method to find the location has to be mesh independent. So, the method can be constructed by considering the reference space as explained in section 2.1 and must allow mapping back to the physical space. The procedure is less trivial than for the angle prediction. The main challenge is the construction of the training set, namely to identify the location of the shock for the functions used in the training phase. A possible approach is described as follows.

First, in the physical space we compute the point k, as in figure 9, that intersects an edge of a troubled-cell with the discontinuity. Different configurations are possible. In any case the shock intersects at least one of the edges having origin in a, either $\overrightarrow{ab}$ or $\overrightarrow{ac}$. If the shock intersects both edges then we use the horizontal axis $\overrightarrow{ab}$.

Then, we know that the affine transformation 3 keeps the colinearity and preserves ratios of distance between points lying on a straight line. That's why point k and the angle of the discontinuity can be mapped in the reference space such that the position of the discontinuity is known in the reference space.

Next step is to find the distance between point $\hat{a}$ and the discontinuity in the reference space, this corresponds to the orange dash line in figure 9. This can be done by a simple optimization. In fact we just need to find the optimal point $\hat{y}_*$ which minimize the distance between $\hat{a}$ and the discontinuity line, i.e, the orthogonal projection of $\overrightarrow{\hat{k}\hat{a}}$ onto the discontinuity line. We can find an explicit $\hat{y}_*$ such that :

$$\hat{y}_* = \hat{k} + ||\overrightarrow{\hat{k}\hat{a}}||_2 sin(\hat{\alpha})\overrightarrow{dir} \tag{6}$$

where $\overrightarrow{dir}$ is the direction of the discontinuity and $\hat{\alpha}$ the angle of the discontinuity.

Thus, the neural network predicts an approximation of the vector $\overrightarrow{\hat{a}\hat{y}_*}$. Point $\hat{k}$ can be computed quite easily, using again the orthogonal projection, and then mapped back to the physical space. Moreover the angle of the discontinuity in the reference space is known because it is perpendicular to the vector $\overrightarrow{\hat{a}\hat{y}_*}$ thus it can also be mapped to the physical space, so that we have the discontinuity with its location in the physical space.

In conclusion, the dataset is very similar to the one in section 2.1 except that the target is now the orthogonal vector $\overrightarrow{\hat{a}\hat{y}_*}$. In the next section, two different approaches are presented in order to predict the orthogonal vector.

3.2 Approaches

The training set is an important part because the neural networks will learn to predict the orthogonal vector from the training set. This section presents two different approaches tried to train a neural network and to get the best prediction. To highlight the results and the improvement of different approaches, a simple discontinuity with a=1 and b=0 in equation (1) will be considered. All datasets have size 10 000 and they are composed of

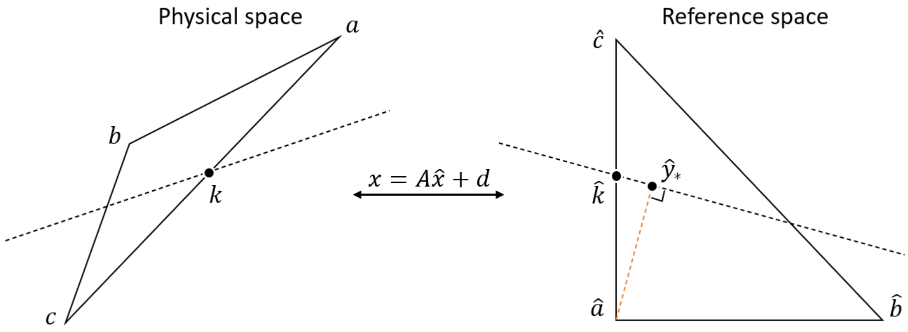


Figure 9: Situation of a troubled-cell abc in the physical space and the same troubled-cell in the reference space $\hat{a}\hat{b}\hat{c}$. The black dash line represents the discontinuity. The red dash line represents the orthogonal vector between $\hat{a}$ and the discontinuity in the reference space. k and $\hat{k}$ represent the intersection between an edge of a triangle and the discontinuity. $\hat{y}_*$ is the optimal point on the discontinuity that minimizes the distance between point $\hat{a}$ and the discontinuity.

60% train, 20% validation and 20% for the test set. The neural network architecture is the one selected in section 2.3, with the only difference being the output dimension greater than one as shown in the next sections. The input corresponds to the troubled-cells with all its neighbours as presented in section 2.1. Moreover the order of the solution is fixed to p=3 and the models are tested for mesh A, B and C as seen in figure 3.

3.2.1 First approach

For the first approach, the two-dimensional target consists of the orthogonal vector components (in cartesian coordinates) $\overrightarrow{\hat{a}\hat{y}_*}$ showed in figure 9. As the discontinuity orientation can be also computed from the orthogonal vector, we can assess the quality of the prediction by comparing with the true discontinuity orientation. The loss of the neural network was the mean square error. The resulting predictions are fairly good as showed in table 7. Instead of analysing the network results by looking at the cartesian components of the predicted orthogonal vector, we believe that considering its distance and angle gives us a better insight on the performances. The prediction of the vector magnitude is quite satisfactory, especially if we recall that there might be cases of different targets for the same input data. The maximum error in the distance is around 0.5 which is 1/4 the distance between $\hat{a}$ and $\hat{b}$. The direction of the vector is not that good, in fact if we consider the accuracy of the predicted angles in table 3 we can see that the prediction of the orthogonal direction is significantly lower. With further investigation on the error, there is bad prediction if the discontinuity is close to point $\hat{a}$ in the reference space . This problem came from the loss function which is the mean square error in this case. In fact if the discontinuity passes near the point $\hat{a}$, then the components of the vector $\overrightarrow{\hat{a}\hat{y}_*}$ are small, thus the error between the prediction and the true is also small leading to a small penalty in the mean square error and bad prediction, especially for the orthogonal orientation.

Mesh			A	B	C
Angle diagnostic	MSE		0.019	0.028	0.035
	Absolute error	mean	5.21	5.56	6.43
		std	5.97	7.61	8.53
		median	3.52	3.56	4.1
		min	3e-3	3e-3	5e-3
		max	65.8	121	147
		10 accuracy	86.8	86.4	81.1
Distance diagnostic	MSE		0.012	0.012	0.015
	Absolute error	mean	8.5e-2	8.3e-2	9.1e-2
		std	7.5e-2	7.3e-2	8.2e-2
		median	6.2e-2	6.2e-2	6.9e-2
		min	4.7e-5	5e-5	6e-5
		max	5e-1	4.8e-1	6.5e-1

Table 7: Results of the first approach to predict the orthogonal vector. The qualities of the orthogonal vector prediction are assessed via the polar coordinate. Angle diagnostic is the quality of the direction of the orthogonal vector. Distance diagnostic is the quality of the size of the vector. The results are computed on a test set and in order to describe the goodness of the prediction, few statistic are showed : MSE is the mean square error. Absolute error corresponds to $|prediction - true|$. 10 accuracy corresponds to the percentage of errors that are less than 10 degree. Note that the results are an average of five rounds of training in order to get better statistics results.

3.2.2 Second approach

The second approach tries to solve the problem mentioned in the previous section by considering the polar representation of a vector. Indeed, we predict the normalized orthogonal vector along with his norm. That is to say if $\vec{v} = \overrightarrow{\hat{a}\hat{y}_*}$ then the targets of the neural network are $v_1 = cos(\beta)$, $v_2 = sin(\beta)$ and $||\vec{v}||$ where β is the angle of $\vec{v}$ w.r.t horizontal axis. Here again the loss was the mean square error. The results of this approach can be found in table 8. Concerning the vector size, the results are similar to the one with the first approach. For the direction of the orthogonal vector, the accuracy is much better. The results are even slightly better for mesh C than the results in section 2.2.3.

Finally using the second approach the prediction of the orthogonal prediction we can predict the location of the discontinuity in a troubled-cell. Combining this approach with the one described in section 2, we have two different ways to predict discontinuity orientation with similar results.

3.3 Results

This section presents the performance of the neural networks architecture selected in section 2.3 using the second approach presented in the previous section with different orders and various meshes. To allow comparison, the results are obtained using 100000 troubled-cells data. Here again the data is decomposed with 60% train, 20% validation and 20% for the test set. Each result is the average of 3 training rounds to obtain better statistics. For each order p, 2 to 5, the neural networks are trained on each mesh and test on the others, Mesh D is used only for testing the neural networks on unseen meshes. Results can be found in tables 14,15, 16 and 17, see section 5.

<table>
<tr><td colspan="3">Mesh</td><td>A</td><td>B</td><td>C</td></tr>
<tr><td rowspan="7">Angle diagnostic</td><td colspan="2">MSE</td><td>0.008</td><td>0.008</td><td>0.014</td></tr>
<tr><td rowspan="6">Absolute error</td><td>mean</td><td>3.82</td><td>3.78</td><td>4.8</td></tr>
<tr><td>std</td><td>3.57</td><td>3.41</td><td>4.8</td></tr>
<tr><td>median</td><td>2.89</td><td>2.89</td><td>3.5</td></tr>
<tr><td>min</td><td>3e-3</td><td>2.7e-3</td><td>4.4e-3</td></tr>
<tr><td>max</td><td>39</td><td>28</td><td>64</td></tr>
<tr><td>10 accuracy</td><td>93.4</td><td>93.7</td><td>88.7</td></tr>
<tr><td rowspan="6">Distance diagnostic</td><td colspan="2">MSE</td><td>0.012</td><td>0.012</td><td>0.015</td></tr>
<tr><td rowspan="5">Absolute error</td><td>mean</td><td>8.2e-2</td><td>8.2e-2</td><td>9.2e-2</td></tr>
<tr><td>std</td><td>7.4e-2</td><td>7.3e-2</td><td>8.2e-2</td></tr>
<tr><td>median</td><td>6.2e-2</td><td>6.0e-2</td><td>7.1e-2</td></tr>
<tr><td>min</td><td>3.3e-5</td><td>3.9e-5</td><td>6e-5</td></tr>
<tr><td>max</td><td>5e-1</td><td>5.3e-1</td><td>6.3e-1</td></tr>
</table>

Table 8: Results of the second approach to predict the orthogonal vector. The qualities of the orthogonal vector prediction are accessed via the polar coordinate. Angle diagnostic is the quality of the direction of the orthogonal vector. Distance diagnostic is the quality of the size of the vector. The results are computed on a test set and in order to describe the goodness of the prediction, few statistic are showed : MSE is the mean square error. Absolute error corresponds to $|prediction - true|$. 10 accuracy corresponds to the percentage of errors that are less than 10 degree. Note that the results are an average of five rounds of training in order to get better statistics results.

3.3.1 Orders p

As mentioned in section 3.2, in order to assess the quality of the predictions of the orthogonal vectors, two criteria are used. The first one is the angle of the orthogonal vector which must be equal to the orientation of the discontinuity, after shifting by $\pm\frac{\pi}{2}$. Thus we can compare the prediction of the method used in section 2 to find discontinuities orientations and the method used to find the angles of the orthogonal vector. The second criteria is the length of the orthogonal vector. As discussed in section 2.4.1, increasing the order p improves the accuracy of the neural network. As figure 10 shows, the mean absolute error of the orthogonal vector angle is improved at least from 35% if the order is increased by 1. Similar results are obtained for the scale of the orthogonal vector. Figure 10 also shows that from order p=4, the angle of the orthogonal vector is more accurate than the angle used to predict the discontinuity orientation. In table 17 for order p=5, the accuracy for the neural network predicted on mesh B is 99.7% with mean 1.7 whereas in table 13 the accuracy is 98.6% and mean 2.4 and it is even better with mesh C and C unstr.

In figure 11, the prediction of the discontinuity location and orientation in a reference triangle on mesh B with p=3 is shown. We can see that the prediction when the orientation error is under 10 degrees is accurate and satisfying, it represents approximately 97% of the test set. When the prediction is between 10 and 20 degrees, the location is not too bad and the orientation of the discontinuity seems acceptable. Note that it concerns only 4% of the test set. Finally for the last one, when the error of the orientation is bigger than 20 degrees, both location and orientation are not accurate anymore. Note that it concerns only 3% of the test set and it happens especially when the discontinuity passes either in the top corner of the triangle or at the bottom right corner.

In figure 13, the predictions of location and orientation of a circular discontinuity on mesh C for various orders are shown. As mentioned before, increasing the order p increases the accuracy of the prediction. The predictions of the circular discontinuity confirm this. Indeed in the zoom plot, we can see that the location predictions for p=5 are very accurate, i.e arrows are close to the true red discontinuity. Note that for $p = 2$ the distance vector is 0.037 and the location seems acceptable. If we compare with the results in figure 8 we can clearly see the effect of the location prediction. Concerning the orientation, the arrow seems to follow the circular discontinuity expect for one in p=2 and 3 and corrected in p=5.

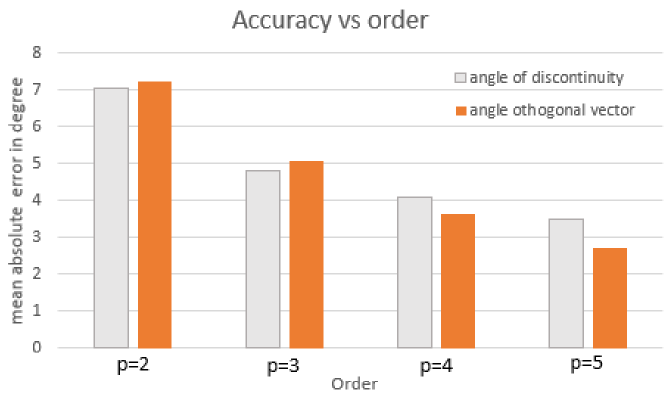


Figure 10: Average of the mean absolute error tested on the same mesh used for training, for each order p. Gray bar correspond to the results obtained in figure 6.

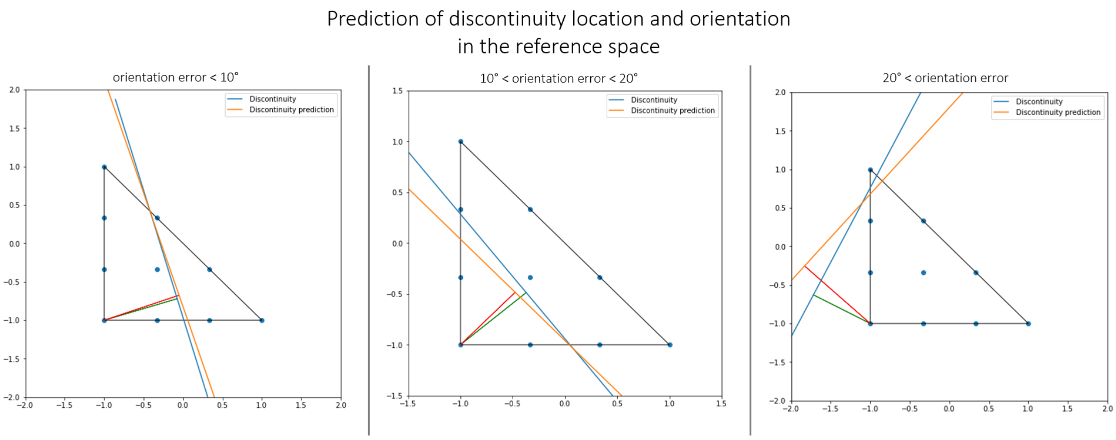
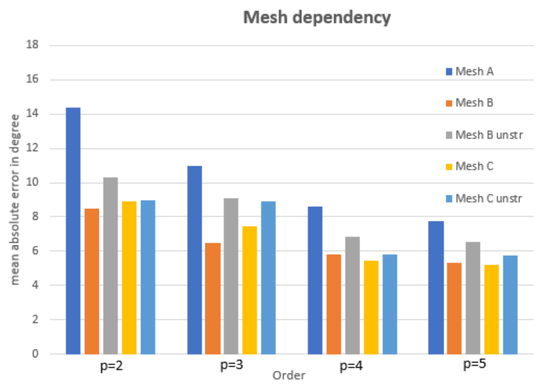


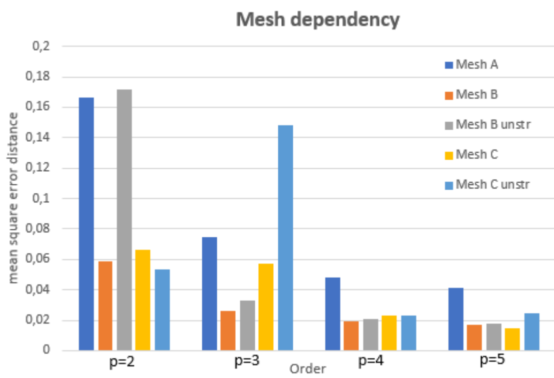
Figure 11: Prediction of the location and orientation of a linear discontinuity as described in equation (1) on mesh B with order p=3. The red vector is the predicted orthogonal vector and the green is the true one. The triangle is a troubled-cells in the reference space. The three plots correspond to different levels of accuracy. The first one shows discontinuity prediction where the absolute error of the orientation is less than 10 degrees, which represents 93% of the test set. The second one shows error that is between 10 and 20 degrees, which represents 4% of the test set and the last one shows error bigger than 20 degrees, which represents 3% of the test set

3.3.2 Mesh dependency

As the neural network presented in section 2, the neural network needs to be mesh independent. In figure 12, we can see that if the neural network is trained using mesh A, then the generalization on other mesh is not good. As mentioned in section 2.3, if the neural network is trained on a *structured* mesh, then the predictions are better on *structured* meshes than on *unstructured* meshes. For order p=4 and p=5, if the neural network is trained on mesh C, then it gives better results on *structured* meshes as presented in table 16 and 17. Moreover if the neural network is trained with *unstructured* mesh, then the generalization on other *unstructured* mesh is not guaranteed. In fact even if the neural networks is slightly more accurate than the model in section 2 , the predictions on unseen *unstructured* mesh is less accurate. Concerning the distance prediction for order p=2 and p=3, train the neural network on *unstructured* mesh lead to poor generalisation. Train the neural network on mesh B seems to give quite good generalisation on other mesh, here also if the model is trained with mesh C the prediction on other meshes is quite good especially for order 5.



(a)



(b)

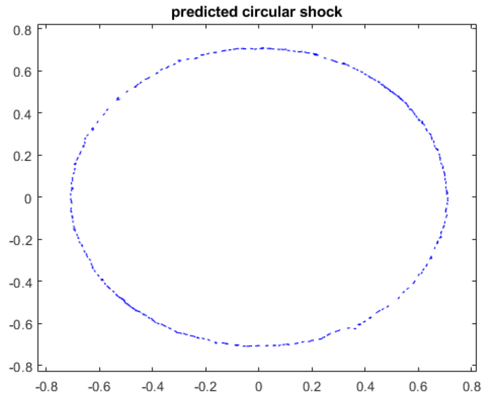
Figure 12: (a) Average of the mean absolute error of the vector angle tested on every meshes except on the training mesh, for each order p. (b) Average of the mean absolute error of the vector norm tested on every mesh except on the training mesh, for each order p. As an example for both, the blue bar represents the mean absolute error of the neural network trained on mesh A and tested on the other meshes. Here the legend represents the training meshes

3.3.3 Robustness

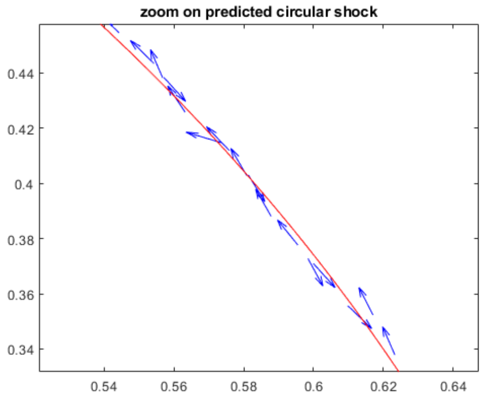
As mentioned in section 2.4.3 the robustness of the neural network is important in view of real applications. The same approach as in section 2.4.3 is used to assess the robustness. The results are slightly worse compared to the one in the previous section. The results are globally satisfactory especially for order p 4 and 5. The length of the orthogonal vector is a bit high for σ=0.1 with order p=2 and p=3. Otherwise the loss is acceptable. Finally, increasing the standard deviation of the gaussian noise shows that the prediction is slightly worse, which is expected. In order to improve the results for real application, one could add noisy data in the training.

σ		None	0.01	0.05	0.1
Order p=2					
Loss angle		0.019	0.023	0.072	0.127
Absolute error	mean	5.9	6.3	8.9	12.0
	std	5.0	5.9	12.4	16.4
	median	4.7	4.8	5.5	6.7
	max	86.7	157.1	165.4	168.4
	accuracy	82.9	81.6	73.6	64.7
Loss distance		0.029	0.031	0.078	0.157
Order p=3					
Loss angle		0.008	0.015	0.044	0.083
Absolute error	mean	3.7	4.1	6.3	8.8
	std	3.4	5.5	10.1	13.9
	median	2.7	2.8	3.5	4.3
	max	52.1	166.0	167.2	172.5
	accuracy	93.9	92.5	84.4	76.2
Loss distance		0.013	0.016	0.046	0.107
Order p=4					
Loss angle		0.004	0.004	0.020	0.052
Absolute error	mean	2.3	2.4	3.9	6.2
	std	2.6	2.9	7.0	11.4
	median	1.6	1.7	2.2	3.1
	max	86.6	93.5	172.7	176.8
	accuracy	98.6	98.1	92.8	85.4
Loss distance		0.005	0.006	0.021	0.050
Order p=5					
Loss angle		0.002	0.003	0.009	0.020
Absolute error	mean	1.7	1.7	2.5	3.7
	std	2.0	2.3	4.6	7.2
	median	1.2	1.2	1.5	2.1
	max	124.8	125.5	175.8	177.5
	accuracy	99.7	99.6	97.5	93.9
Loss distance		0.002	0.002	0.006	0.016

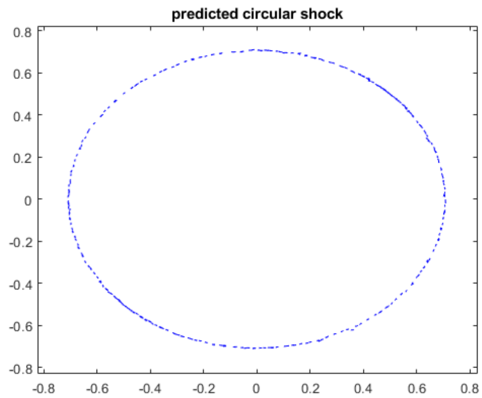
Table 9: Robustness results obtained from training the neural networks with 60000 samples from mesh B and tested on a test set from mesh B with different added noise. σ is the standard deviation of the gaussian noise added to the test set (none means that there is no noise added).The neural network was the one selected in section 2.3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles are in degree. These are statistics summaries and accuracy corresponds to the percentage of errors that are less than 10 degree. Loss angle is the mean square error. It gives an estimation of the prediction quality of the vector angle. Loss distance is also the mean square error and it gives an estimation of the prediction quality of the vector length



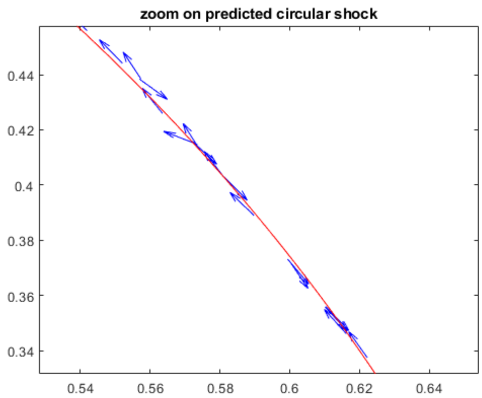
(a) Mesh C p=2



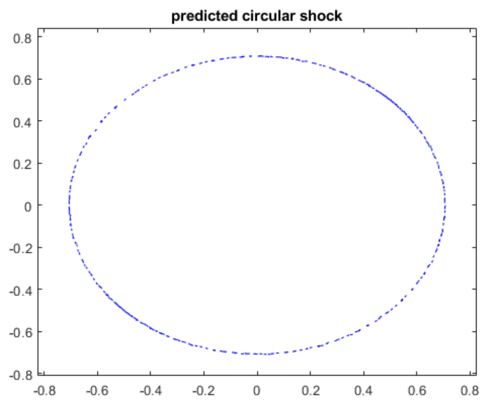
(b) Mesh C p=2



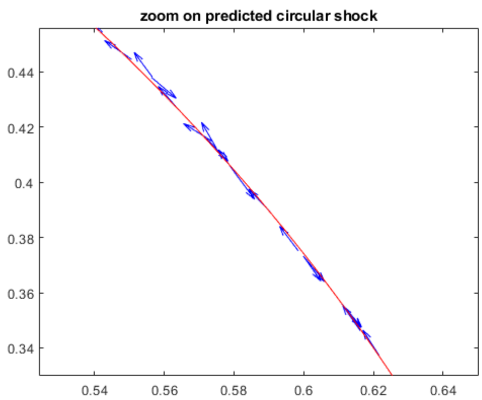
(c) Mesh C p=3



(d) Mesh C p=3



(e) Mesh C p=5



(f) Mesh C p=5

Figure 13: Prediction of the location and orientation of a circular discontinuity as described in equation (2). The prediction is on mesh C with various order p. In the zoom plots, the red line is the true circular discontinuity.

4 Conclusion

In this work, we trained two families of neural networks in order to predict orientation and location of a shock in a troubled-cell. In particular, a different network is trained for each polynomial order p. The training set is obtained by computing various discontinuities on different meshes, whose orientation and location is known a priori. The network is trained offline, after which it can be used on-line as a black box to predict location and orientation of a discontinuity in a troubled-cell for general conservational laws on triangular grids. Furthermore, one advantage of the method we proposed is that it does not require the specification of problem-dependent parameters. Through several tests, we showed that the neural network proposed is globally mesh independent and performs quite well on various types of mesh. Moreover the neural network seems to be robust against noisy data which is good for real applications.

Future work can be done. One should test the trained networks in a concrete, time-dependent problem. Given the good generalization and robustness properties we showed, we expect the networks to be able to capture well both the direction and the location of discontinuities independently of the underlying problem. We could also investigate the computational cost of the online application of the neural network. As it mainly consists of matrix-vector multiplications, we expect it to be quite low. Moreover the neural network can be trained with noisy data and data generated from numerical simulations, we could expect better results in real applications.

5 Annex

Order p=2		Mesh	A	B	B unstr	C	C unstr	D
Train on mesh A	Loss		0.028	0.16	0.43	0.24	0.52	0.20
	Absolute error	mean	7.3	14.8	23.7	17.4	24.9	15.9
		std	6.2	17.8	28.5	22.1	31.6	20.0
		median	5.7	9.1	15.5	10.3	16.0	9.4
		max	93.1	265.6	361.7	308.0	371.1	297.2
		accuracy	74.5	53.5	35.3	48.9	34.3	52.05
Train on mesh B	Loss		0.124	0.019	0.047	0.030	0.062	0.029
	Absolute error	mean	12.2	5.9	8.9	7.2	10.0	7.1
		std	16.0	5.2	8.5	6.7	10.1	6.6
		median	6.8	4.6	6.7	5.4	7.3	5.3
		max	129.4	48.6	108.8	98.6	134.6	121.1
		Accuracy	64.8	81.9	66.9	76.0	62.9	76.0
Train on mesh B unstr	Loss		0.125	0.043	0.031	0.047	0.054	0.048
	Absolute error	mean	13.2	8.3	7.3	8.5	9.0	8.5
		std	15.3	8.4	6.8	9.0	9.8	9.2
		median	8.1	5.9	5.4	5.9	6.5	6.0
		max	153.4	90.5	119.2	181.5	165.2	180.5
		accuracy	57.3	70.9	74.9	70.6	68.0	70.2
Train on mesh C	Loss		0.141	0.025	0.049	0.024	0.069	0.027
	Absolute error	mean	12.6	6.6	9.0	6.5	9.9	6.7
		std	17.3	6.0	8.8	6.0	11.1	6.5
		median	6.9	5.1	6.5	4.9	6.8	5.0
		max	153.7	87.0	142.3	126.6	156.7	151.5
		accuracy	64.3	78.4	67.8	78.5	64.8	78.3
Train on mesh C unstr	Loss		0.121	0.038	0.041	0.037	0.040	0.035
	Absolute error	mean	12.9	8.0	8.3	7.8	8.3	7.6
		std	15.1	7.8	7.9	7.7	7.7	7.5
		median	7.6	5.8	6.2	5.7	6.1	5.6
		max	102.6	89.3	112.6	88.4	105.1	106.1
		accuracy	59.5	72.0	69.9	73.1	70.0	73.9

Table 10: Prediction of the discontinuity orientation with method of order p=2. The results obtained from training the neural network with 60 000 data samples . The neural network is the one selected in section 2.3. It is tested for every mesh with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss is the new loss that is define in equation 5 on the test set.

Order p=3		Mesh	A	B	B unstr	C	C unstr	D
Train on mesh A	Loss		0.012	0.095	0.127	0.116	0.149	0.113
	Absolute error	mean	4.4	10.9	13.9	12.1	14.8	12.0
		std	4.3	13.7	14.8	15.2	16.4	15.0
		median	3.3	5.7	9.3	6.6	9.8	6.6
		max	84.3	118.1	212.4	213.6	218.2	190.0
		accuracy	90.9	66.6	52.5	63.0	50.5	63.2
Train on mesh B	Loss		0.088	0.007	0.045	0.048	0.070	0.020
	Absolute error	mean	8.1	3.4	7.8	6.5	9.0	5.2
		std	14.8	3.1	9.2	10.5	11.9	6.2
		median	3.5	2.5	5.3	3.6	5.7	3.6
		max	130.4	39.0	148.1	147.9	144.6	141.1
		accuracy	83.1	94.9	74.2	83.9	71.0	87.1
Train on mesh B unstr	Loss		0.101	0.044	0.014	0.064	0.053	0.051
	Absolute error	mean	11.2	7.9	4.8	9.0	8.2	8.5
		std	14.3	8.9	4.6	11.3	10.2	9.7
		median	6.3	5.2	3.6	5.4	5.3	5.6
		max	153.7	89.5	87.1	163.0	180.9	158.5
		accuracy	65.9	74.8	88.5	71.4	74.1	71.0
Train on mesh C	Loss		0.075	0.013	0.043	0.019	0.051	0.018
	Absolute error	mean	8.1	4.6	7.4	5.1	8.2	5.2
		std	13.3	4.6	9.1	5.9	9.9	5.6
		median	4.2	3.5	4.9	3.5	5.4	3.8
		max	150.4	85.8	143.1	136.6	150.8	140.2
		accuracy	81.2	89.7	76.0	87.2	72.8	86.5
Train on mesh C unstr	Loss		0.124	0.025	0.040	0.079	0.024	0.028
	Absolute error	mean	11.1	6.0	6.9	8.6	6.3	6.3
		std	16.7	6.7	8.9	13.5	6.2	7.1
		median	5.4	4.2	4.7	4.6	4.6	4.4
		max	182.2	128.7	155.2	166.2	119.3	150.8
		accuracy	70.6	83.1	79.7	77.2	80.5	81.4

Table 11: Prediction of the discontinuity orientation with method of order p=3. The results obtained from training the neural network with 60 000 data samples . The neural network is the one selected in section 2.3. It is tested for every mesh with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss is the new loss that is define in equation 5 on the test set.

Order p=4		Mesh	A	B	B unstr	C	C unstr	D
Train on mesh A	Loss		0.006	0.043	0.113	0.056	0.107	0.063
	Absolute error	mean	3.0	7.5	12.2	8.2	12.2	8.6
		std	3.0	9.1	14.8	10.6	14.1	11.5
		median	2.2	4.4	7.8	4.9	8.1	5.1
		max	88.8	98.3	212.9	188.4	193.7	179.8
		accuracy	97.2	76.2	59.3	73.6	58.3	72.8
Train on mesh B	Loss		0.050	0.005	0.042	0.016	0.035	0.021
	Absolute error	mean	5.8	2.8	7.3	4.6	7.1	4.8
		std	11.4	3.1	9.0	5.6	8.0	6.8
		median	2.7	2.1	4.6	3.0	4.6	3.1
		max	94.2	85.3	89.7	88.5	89.5	89.6
		accuracy	90.2	97.7	77.1	89.9	77.8	89.8
Train on mesh B unstr	Loss		0.062	0.061	0.023	0.062	0.025	0.096
	Absolute error	mean	7.8	6.7	5.3	6.9	5.8	8.4
		std	11.8	12.3	6.8	12.1	6.9	15.4
		median	4.3	3.8	3.6	4.0	4.0	4.3
		max	134.6	179.4	95.8	177.6	155.1	188.8
		accuracy	78.5	83.5	87.9	81.7	83.9	78.9
Train on mesh C	Loss		0.047	0.011	0.052	0.016	0.047	0.024
	Absolute error	mean	6.1	3.8	7.9	4.4	7.7	4.9
		std	10.7	4.4	10.2	5.6	9.6	7.2
		median	3.2	2.7	4.8	3.0	4.8	3.2
		max	145.4	88.9	142.1	145.3	143.3	121.8
		accuracy	87.6	94.2	75.5	91.1	75.8	90.0
Train on mesh C unstr	Loss		0.041	0.019	0.026	0.021	0.016	0.024
	Absolute error	mean	6.6	4.9	5.8	5.3	4.9	5.5
		std	9.4	6.0	7.1	6.4	5.3	6.8
		median	3.9	3.3	3.9	3.6	3.6	3.6
		max	91.1	88.2	89.7	148.4	113.6	89.5
		accuracy	82.7	88.7	84.7	86.4	87.9	86.1

Table 12: Prediction of the discontinuity orientation with method of order p=4. The results obtained from training the neural network with 60 000 data samples . The neural network is the one selected in section 2.3. It is tested for every mesh with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss is the new loss that is define in equation 5 on the test set.

Order p=5		Mesh	A	B	B unstr	C	C unstr	D
Train om mesh A	Loss		0.010	0.026	0.093	0.038	0.077	0.028
	Absolute error	mean	3.0	5.6	10.7	6.8	10.3	6.2
		std	4.6	7.3	13.6	8.8	12.0	7.3
		median	2.0	3.5	6.7	4.1	6.6	3.9
		max	112.1	89.7	219.3	144.0	221.1	143.7
		accuracy	97.0	84.7	64.2	80.0	64.4	80.7
Train on mesh B	Loss		0.045	0.004	0.044	0.016	0.038	0.012
	Absolute error	mean	4.8	2.4	7.1	4.1	6.9	3.7
		std	11.0	2.6	9.5	5.8	8.6	4.9
		median	2.1	1.8	4.4	2.6	4.4	2.4
		max	125.6	81.7	143.5	143.5	147.5	138.3
		accuracy	93.0	98.6	78.0	91.9	78.0	93.4
Train on mesh B unstr	Loss		0.050	0.029	0.015	0.032	0.020	0.026
	Absolute error	mean	7.3	5.8	3.8	6.1	5.3	5.8
		std	10.3	7.8	5.7	8.1	6.2	7.0
		median	4.1	3.5	2.4	3.7	3.4	3.8
		max	89.9	89.7	89.6	89.3	87.8	89.4
		accuracy	79.6	85.0	94.2	83.5	86.0	83.8
Train on mesh C	Loss		0.055	0.025	0.048	0.015	0.046	0.012
	Absolute error	mean	6.2	4.4	7.8	4.1	8.0	3.9
		std	11.8	7.5	9.7	5.4	9.3	4.9
		median	2.8	2.6	4.7	2.7	5.0	2.7
		max	147.2	144.1	152.7	111.5	149.4	130.7
		accuracy	87.5	92.1	74.9	92.4	73.2	93.0
Train on mesh C unstr	Loss		0.037	0.019	0.026	0.027	0.012	0.019
	Absolute error	mean	5.9	4.5	5.1	5.1	4.2	4.7
		std	9.3	6.4	7.5	7.7	4.7	6.2
		median	3.3	2.9	3.1	3.1	2.9	3.1
		max	137.1	102.8	121.1	175.6	122.8	158.1
		accuracy	86.5	90.9	88.5	88.4	91.5	89.3

Table 13: Prediction of the discontinuity orientation with method of order p=5. The results obtained from training the neural network with 60 000 data samples . The neural network is the one selected in section 2.3. It is tested for every mesh with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error corresponds to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angle are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss is the new loss that is define in equation 5 on the test set.

Order p=2		mesh	A	B	B unstr	C	C unstr	D
Train on mesh A	Loss angle		0.018	0.115	0.194	0.106	0.142	0.114
	Absolute error	mean	5.6	12.7	18.1	12.5	15.9	12.8
		std	5.2	14.5	17.5	13.7	14.6	14.3
		median	4.0	8.1	12.8	8.2	11.6	8.1
		max	57.5	156.8	167.6	147.5	133.0	147.9
		accuracy	83.6	58.0	41.1	56.9	44.0	57.5
	Loss distance		0.036	0.157	0.178	0.156	0.170	0.170
Train on mesh B	Loss angle		0.064	0.020	0.083	0.034	0.048	0.037
	Absolute error	mean	7.9	6.0	10.8	7.3	9.0	7.3
		std	12.0	5.3	12.3	7.5	8.6	8.1
		median	4.7	4.7	7.3	5.2	6.6	5.0
		max	164.5	128.9	148.0	118.9	124.6	141.8
		accuracy	78.9	82.2	62.5	75.8	66.5	76.6
	Loss distance		0.073	0.029	0.060	0.051	0.055	0.055
Train on mesh B unstr	Loss angle		0.084	0.058	0.065	0.117	0.059	0.071
	Absolute error	mean	10.5	8.9	9.8	12.5	9.8	9.7
		std	12.7	10.5	10.7	14.7	9.7	11.6
		median	6.6	6.0	6.9	7.9	7.1	6.5
		max	131.7	126.4	139.1	137.3	129.5	137.3
		accuracy	65.9	71.0	64.5	58.8	63.9	67.2
	Loss distance		0.173	0.114	0.041	0.340	0.093	0.138
Train on mesh C	Loss angle		0.067	0.038	0.132	0.026	0.072	0.027
	Absolute error	mean	8.2	7.3	12.5	6.7	9.8	6.7
		std	12.3	8.3	16.4	6.2	11.6	6.5
		median	4.9	5.3	7.6	5.0	7.0	5.0
		max	167.5	147.2	162.2	97.1	155.7	131.3
		accuracy	77.1	77.8	60.6	78.4	65.2	78.1
	Loss distance		0.079	0.073	0.089	0.037	0.054	0.035
Train on mesh C unstr	Loss angle		0.067	0.053	0.086	0.052	0.035	0.043
	Absolute error	mean	9.3	8.4	10.9	8.4	8.0	7.9
		std	11.4	10.1	12.7	9.8	7.1	8.8
		median	6.1	5.7	7.0	5.7	6.1	5.6
		max	139.8	114.4	115.9	105.9	58.2	106.6
		accuracy	69.7	73.3	63.8	72.4	70.6	73.6
	Loss distance		0.073	0.047	0.050	0.053	0.038	0.044

Table 14: Prediction of the location discontinuity with method of order p=2. The results obtained from training the neural networks with 60 000 data samples. The neural network is the one selected in section 2.3. It is tested for every meshes with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error correspond to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles of orthogonal vectors are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss angle is the mean square error computed on the test set. It gives a goodness of the angle of the orthogonal vector. Loss distance which is a mean square error gives an estimation of the goodness of the orthogonal vector length.

Order p=3	mesh		A	B	B unstr	C	C unstr	D
Train on mesh A	Loss angle		0.011	0.083	0.115	0.070	0.101	0.061
	Absolute error	mean	4.3	10.6	12.8	9.6	12.5	9.3
		std	4.1	12.4	14.5	11.5	13.0	10.6
		median	3.2	6.9	8.4	6.3	8.6	6.3
		max	93.2	160.6	159.3	171.7	162.1	169.8
		accuracy	91.6	64.2	56.6	67.4	55.6	67.8
	Loss distance		0.015	0.079	0.073	0.067	0.090	0.064
Train on mesh B	Loss angle		0.045	0.008	0.063	0.020	0.035	0.016
	Absolute error	mean	5.9	3.7	8.5	5.4	7.4	5.1
		std	10.6	3.4	11.6	5.9	7.6	5.0
		median	3.3	2.8	5.2	3.7	5.3	3.7
		max	152.6	54.5	158.1	128.4	169.2	87.8
		accuracy	88.4	93.7	74.0	85.4	75.3	87.0
	Loss distance		0.039	0.013	0.028	0.019	0.027	0.017
Train on mesh B unstr	Loss angle		0.159	0.088	0.044	0.080	0.047	0.075
	Absolute error	mean	11.5	8.8	6.8	8.9	7.8	8.4
		std	19.6	14.5	9.9	13.4	9.6	13.1
		median	5.5	4.9	4.3	5.1	5.2	5.1
		max	172.3	174.2	196.9	166.5	153.5	174.1
		accuracy	69.9	75.6	81.2	74.0	74.9	74.7
	Loss distance		0.046	0.033	0.015	0.033	0.025	0.029
Train on mesh C	Loss angle		0.041	0.034	0.095	0.013	0.080	0.014
	Absolute error	mean	6.0	5.7	10.7	4.7	10.1	4.9
		std	9.8	8.8	13.9	4.5	12.5	4.6
		median	3.6	3.5	6.4	3.4	6.4	3.6
		max	153.7	140.6	186.4	89.4	168.9	118.8
		accuracy	86.4	87.4	65.8	88.5	66.5	87.9
	Loss distance		0.040	0.125	0.073	0.015	0.146	0.016
Train on mesh C unstr	Loss angle		0.100	0.062	0.051	0.090	0.019	0.070
	Absolute error	mean	10.1	8.1	7.5	9.9	5.8	8.9
		std	14.8	11.4	10.5	13.7	5.3	11.9
		median	5.3	4.7	4.6	5.7	4.4	5.5
		max	168.5	160.4	151.2	168.7	130.1	165.0
		accuracy	72.1	76.8	79.2	70.0	82.7	72.5
	Loss distance		0.230	0.127	0.020	0.223	0.018	0.142

Table 15: Prediction of the location discontinuity with method of order p=3. The results obtained from training the neural networks with 60 000 data samples. The neural network is the one selected in section 2.3. It is tested for every meshes with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error correspond to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles of orthogonal vectors are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss angle is the mean square error computed on the test set. It gives a goodness of the angle of the orthogonal vector. Loss distance which is a mean square error gives an estimation of the goodness of the orthogonal vector length.

Order p=4	mesh		A	B	B unstr	C	C unstr	D
Train on mesh A	Loss angle		0.004	0.040	0.097	0.063	0.058	0.052
	Absolute error	mean	2.6	6.6	11.0	8.4	9.4	7.7
		std	2.6	9.2	14.0	11.5	10.0	10.4
		median	1.9	3.8	6.6	4.7	6.4	4.6
		max	64.9	174.8	172.0	173.2	167.3	170.2
		accuracy	98.1	79.7	64.4	73.3	66.4	75.3
	Loss distance		0.006	0.039	0.049	0.052	0.051	0.051
Train on mesh B	Loss angle		0.030	0.003	0.066	0.029	0.033	0.011
	Absolute error	mean	4.5	2.3	8.3	5.2	6.9	4.1
		std	8.7	2.2	12.0	8.0	7.7	4.3
		median	2.4	1.6	4.8	3.0	4.8	2.9
		max	154.9	38.4	173.4	124.1	161.7	82.1
		accuracy	92.4	98.7	76.2	88.5	77.7	91.5
	Loss distance		0.025	0.005	0.023	0.016	0.024	0.010
Train on mesh B unstr	Loss angle		0.075	0.047	0.030	0.057	0.034	0.038
	Absolute error	mean	7.9	6.3	5.2	7.3	6.3	6.3
		std	13.5	10.6	8.4	11.5	8.3	9.1
		median	4.2	3.6	3.2	4.1	4.1	3.9
		max	171.6	163.0	207.2	171.0	176.5	171.6
		accuracy	79.1	84.2	88.3	80.0	81.8	82.3
	Loss distance		0.028	0.017	0.008	0.026	0.015	0.018
Train on mesh C	Loss angle		0.031	0.007	0.066	0.008	0.032	0.011
	Absolute error	mean	4.8	3.2	8.3	3.6	6.9	4.0
		std	8.8	3.6	12.1	3.7	7.4	4.4
		median	2.5	2.3	4.6	2.6	4.7	2.9
		max	156.8	138.0	171.4	87.0	146.0	166.7
		accuracy	91.5	96.3	76.3	94.2	77.4	92.8
	Loss distance		0.039	0.014	0.027	0.008	0.025	0.012
Train on mesh C unstr	Loss angle		0.034	0.017	0.054	0.031	0.012	0.019
	Absolute error	mean	6.2	4.8	7.0	5.9	4.4	5.2
		std	8.4	5.5	11.3	8.2	4.2	5.8
		median	3.7	3.2	3.8	3.6	3.1	3.5
		max	155.5	117.7	174.1	153.5	85.4	132.3
		accuracy	82.9	88.1	82.9	84.8	90.6	86.2
	Loss distance		0.029	0.016	0.013	0.041	0.009	0.015

Table 16: Prediction of the location discontinuity with method of order p=4. The results obtained from training the neural networks with 60 000 data samples. The neural network is the one selected in section 2.3. It is tested for every meshes with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error correspond to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles of orthogonal vectors are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss angle is the mean square error computed on the test set. It gives a goodness of the angle of the orthogonal vector. Loss distance which is a mean square error gives an estimation of the goodness of the orthogonal vector length.

Order p=5		mesh	A	B	B unstr	C	C unstr	D
Train on mesh A	Loss angle		0.003	0.029	0.091	0.049	0.068	0.036
Train on mesh A	Absolute error	mean	1.9	5.5	10.4	7.1	9.5	6.3
Train on mesh A	Absolute error	std	2.0	8.0	13.7	10.5	11.4	8.7
Train on mesh A	Absolute error	median	1.4	3.1	6.1	3.9	6.0	3.8
Train on mesh A	Absolute error	max	77.8	173.3	171.7	173.3	173.9	174.3
Train on mesh A	Absolute error	accuracy	99.6	83.4	67.4	78.7	67.5	81.1
Train on mesh A	Loss distance		0.003	0.029	0.039	0.049	0.054	0.037
Train on mesh B	Loss angle		0.037	0.002	0.069	0.013	0.039	0.011
Train on mesh B	Absolute error	mean	4.1	1.7	8.2	3.8	7.0	3.5
Train on mesh B	Absolute error	std	10.1	1.6	12.5	5.2	8.7	4.7
Train on mesh B	Absolute error	median	1.7	1.2	4.4	2.3	4.5	2.3
Train on mesh B	Absolute error	max	168.4	23.9	166.4	139.3	162.4	174.6
Train on mesh B	Absolute error	accuracy	93.8	99.7	77.1	92.6	77.4	93.8
Train on mesh B	Loss distance		0.025	0.002	0.021	0.010	0.022	0.007
Train on mesh B unstr	Loss angle		0.071	0.036	0.019	0.051	0.037	0.045
Train on mesh B unstr	Absolute error	mean	7.6	5.9	3.5	6.7	6.1	6.5
Train on mesh B unstr	Absolute error	std	13.2	9.0	7.1	10.9	9.1	10.2
Train on mesh B unstr	Absolute error	median	3.8	3.4	1.9	3.7	3.5	3.8
Train on mesh B unstr	Absolute error	max	173.9	173.6	179.6	168.0	163.0	174.9
Train on mesh B unstr	Absolute error	accuracy	79.8	84.4	94.8	81.8	84.2	82.3
Train on mesh B unstr	Loss distance		0.025	0.014	0.004	0.019	0.015	0.016
Train on mesh C	Loss angle		0.027	0.004	0.071	0.005	0.036	0.007
Train on mesh C	Absolute error	mean	4.1	2.5	8.8	2.9	7.3	3.3
Train on mesh C	Absolute error	std	8.4	2.4	12.4	3.0	7.9	3.5
Train on mesh C	Absolute error	median	2.0	1.8	4.7	2.1	4.6	2.3
Train on mesh C	Absolute error	max	170.4	66.7	153.8	83.0	158.4	129.2
Train on mesh C	Absolute error	accuracy	94.0	98.5	73.1	96.9	75.4	95.5
Train on mesh C	Loss distance		0.031	0.004	0.016	0.005	0.017	0.006
Train on mesh C unstr	Loss angle		0.047	0.019	0.051	0.026	0.008	0.023
Train on mesh C unstr	Absolute error	mean	6.6	4.9	6.6	5.5	3.5	5.2
Train on mesh C unstr	Absolute error	std	10.4	6.0	11.1	7.3	3.6	6.9
Train on mesh C unstr	Absolute error	median	3.5	3.1	3.4	3.4	2.4	3.3
Train on mesh C unstr	Absolute error	max	171.5	135.6	147.1	149.7	49.7	157.6
Train on mesh C unstr	Absolute error	accuracy	82.9	87.7	84.3	85.5	94.2	86.8
Train on mesh C unstr	Loss distance		0.050	0.017	0.012	0.025	0.006	0.020

Table 17: Prediction of the location discontinuity with method of order p=5. The results obtained from training the neural networks with 60 000 data samples. The neural network is the one selected in section 2.3. It is tested for every meshes with 20 000 data samples and the neural network is trained with each mesh in order to compare the generalization capability. The meshes are presented in figure 3. Absolute error correspond to $|\hat{\alpha}_{prediction} - \hat{\alpha}_{true}|$ with $\hat{\alpha}_{true}$ in the test set and the angles of orthogonal vectors are in degree. These are statistics summaries and accuracy correspond to the percentage of errors that are less than 10 degree. Loss angle is the mean square error computed on the test set. It gives a goodness of the angle of the orthogonal vector. Loss distance which is a mean square error gives an estimation of the goodness of the orthogonal vector length.

References

[1] Jan S. Hesthaven Deep Ray. Detecting troubled-cells on two-dimensional unstructured grids using a neural network. *Journal of Computational Physics*, 2019.

[2] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, 2006.