Energy-Conservative Data-Driven Modelling for the two-scale Navier-Stokes Equations

Master of Science Thesis

by

Andrea Bettini

Faculty of Aerospace Engineering

Delft University of Technology

Energy-Conservative Data-Driven Modelling for the two-scale Navier-Stokes Equations

Master of Science Thesis

by

Andrea Bettini

to obtain the degree of Master of Science
in Aerospace Engineering at Delft University of Technology,
to be defended publicly on August 29th, 2023 at 10:00

Student number:	4656970	
Project duration:	January 9, 2023 – August 29, 2023	
Thesis committee:	Dr. S. Hulshoff,	TU Delft, Responsible Supervisor
	Dr. R. P. Dwight,	TU Delft, Chair
	Dr. B. Chen,	TU Delft, External Examiner

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

This thesis presents an energy-conservative data-driven approach in modelling the closure terms of the Navier-Stokes equations casted through the Variational Multiscale (VMS) framework. For context, the VMS framework is applied in designing stabilised finite element methods for multiscale phenomena in which stability is not guaranteed. Under standard applications of this framework with appropriate cubature rules and iteration routines, the only error source is model related in the assumptions for the fine scales, thereby influencing the closure terms of the coarse-scale equations. This issue can be rectified by predicting the perfect closure terms with the application of neural networks. To maintain energy conservation (as a generality, not as a guarantee), a constrained objective function was formulated, which was then translated into an unconstrained problem suitable for present day neural network training tools via the Augmented Lagrangian method. The features of the eight neural networks used were extracted from two spatiotemporal stencils. A priori evaluation of the fine-scale model demonstrated a tendency to overpredict closure terms for the $\mathrm{Re}_\tau = 180$ turbulent channel flow test case, yielding desired energy dissipative behaviour. This was true for both regimes of the flow in which the closure terms yield energy gain and energy loss via backscatter and cascade respectively. Unverified a posteriori simulation of the test case with the model displayed excessively high energy dissipation, in addition to high frequency oscillations, which were undampened by the time marching method used.

Preface

With my three year journey for my master degree coming to an end, I must say that it was quite an enjoyable journey, with plenty of interesting courses and professors. Obviously there were quite a few hurdles, mostly presented in finding the motivation necessary to write the actual thesis itself (and not the actual cool experimental stuff you do on the computer). In other words, software documentation is the bane of my existence (and everyone elses).

To get through the hellish nightmare of writing, there have been plenty of people to push me through. Of great importance to me is my family. They have always been there to make sure that I was doing what I wanted to do, and they pushed me to finish whenever I was stressed (phone calls everyday) to whenever I was relaxed (phone calls every week, or two, or three...).

Next are *the boys*, who are at this point are an extension of my family as I have witnessed their experience and stupidity *every day for the past ~12–13 years*. I attribute my hermit behaviour to them draining me of any social battery remaining each day.

On par with the boys is my supervisor Steven Hulshoff. Having exchanged 529 emails as of 2023-08-16 and meeting every week for the past six months to discuss my thesis progress, his help has been immense in getting from the start to the very end.

Last but not least, I'd like to thank everyone else in my endeavours here, predominantly my downstairs neighbours, Lora and Sash, my bouldering group, and my Magic the Gathering group.

Contents

List of Figures **v**

List of Tables **viii**

Nomenclature **ix**

1 Introduction **1**

2 Theory **4**
2.1 Navier-Stokes Equations 4
2.2 Artificial Neural Networks 4
2.3 Variational Multiscale Method 6
2.4 ANN-Coupled VMS Methods 9

3 ANN Construction **11**
3.1 Problem Statement 11
3.2 Feature Set Selection 12
3.2.1 Solver Coupling 12
3.2.2 Norm Selection and Stencil Choice 14
3.2.3 Data Generation and Hardware 17
3.2.4 Features Correlations and Choices 18
3.2.5 Data Augmentation 24
3.3 ANN Architecture 24
3.3.1 ANN Type 24
3.3.2 Objective Function 25
3.3.2.1 Closure Term Error 25
3.3.2.2 Force-Free Weak-Residual Error 26
3.3.2.3 Solver Adherence Penalty 27
3.3.2.4 Energy Transfer Penalty (Conservation of Energy) 28
3.3.2.5 Network Problem Statement 31
3.3.3 Network Hyperparameters 32
3.3.3.1 Unconstrained Problem Initial Guess 32
3.3.3.2 Optimiser Choice and Learning Rate 32
3.3.3.3 Batch Size and Epochs 34
3.3.3.4 Constrained Problem Updates 34
3.3.3.5 Activation Functions 35
3.3.3.6 Neurons and Layers 35

4 A priori Evaluation **38**

4.1 Training History . . . 38
4.2 Truth-Prediction Correlations . . . 39
4.3 Channel Predictions . . . 44
4.4 Data Augmentation Effects . . . 46

5 Preliminary A posteriori Evaluation **48**
5.1 Chapter Note . . . 48
5.2 Energy Time Series . . . 48
5.3 Wall Profiles . . . 49
5.4 Error Profiles . . . 52

6 Conclusion **54**
6.1 Answers to the Research Questions . . . 55
6.2 Recommendations . . . 57

Works Cited **60**

Appendices
A H_0^1 TCF Dataset Histograms . . . 62
A.1 t^{n+1} - t^{n+1} Histograms . . . 63
A.1.1 (i, j, k) Histograms . . . 63
A.1.2 $(i-1, j, k)$ Histograms . . . 72
A.2 t^{n} - t^{n+1} Histograms . . . 81
A.2.1 (i, j, k) Histograms . . . 81
A.2.2 $(i-1, j, k)$ Histograms . . . 91
A.2.3 $(i-2, j, k)$ Histograms . . . 101
A.2.4 $(i-3, j, k)$ Histograms . . . 111
A.2.5 $(i-4, j, k)$ Histograms . . . 121
A.2.6 $(i+1, j, k)$ Histograms . . . 131
A.2.7 $(i, j-1, k)$ Histograms . . . 141
A.2.8 $(i, j+1, k)$ Histograms . . . 151
A.2.9 $(i, j, k \pm 1)$ Histograms . . . 161
A.3 t^{n-1} - t^{n+1} Histograms . . . 171
A.3.1 (i, j, k) Histograms . . . 171
A.3.2 $(i-1, j, k)$ Histograms . . . 181
A.3.3 $(i-2, j, k)$ Histograms . . . 191
A.3.4 $(i-3, j, k)$ Histograms . . . 201
A.3.5 $(i-4, j, k)$ Histograms . . . 211
A.3.6 $(i+1, j, k)$ Histograms . . . 221
A.3.7 $(i, j-1, k)$ Histograms . . . 231
A.3.8 $(i, j+1, k)$ Histograms . . . 241
A.3.9 $(i, j, k \pm 1)$ Histograms . . . 251
B Supplementary Figures . . . 261
B.1 A priori Evaluation . . . 261

List of Figures

2.1 Single hidden layer ANN schematic [9, Fig. 2.] 5
2.2 left: Spacetime domain, right: spacetime slabs [14, Fig. 1.] 7

3.1 Example visualisation of the TCF domain. The utilised face colours are used to distinguish form orientation. 11
3.2 Fine-scale Greens' function of the linear, steady, 2D advection-diffusion problem for linear triangular elements evaluated at the physical-space position $(x_0, y_0)^\mathsf{T} \approx (0.6, 0.8)^\mathsf{T}$. 15
3.3 Poisson fine-scale Greens' function using the H_0^1 projector, (area-normalised) square elements with linear shape functions, and a diffusivity coefficient of 1.0. Courtesy of Shrestha and his framework developed for [15]. 16
3.4 Visualisation of the neural networks' spatial stencils used for the half-channel defined by the $z = 0$ wall. Coordinate system origins denotes the (i, j, k)-element, and its orientation denotes the direction in which the element index positively increments in. 17
3.5 DNS TCF bulk velocity profile as a function of time. Green regions indicates time interval where data set projection occurs. The data sets are labeled using the annotations provided in the top-right corner of the green-shaded regions. . 18
3.6 Example learning rates over 300 epochs, restarted at epoch 100. — Standard Cyclical Learning Rate, — Custom Cyclical Learning Rate 33
3.7 Example ANN validation losses for the $(\mu^k = 0, \lambda^k = 0)$ problem with L_2 data set using different learning rates for shape function $q = 1$. Neural-network parameters: 5 hidden layers, each with 600 ReLU-activated neurons, linearly-activated output layer, batch size of 2500. — Standard Cyclical Learning Rate, — Custom Cyclical Learning Rate, — Learning Rate 10^{-3}, — Learning Rate 10^{-4} 34
3.8 Grid search between neurons and layers for the $\mu^k = 0 \wedge \lambda^k = 0$ problem with L_2 data set for shape function $q = 1$. Measured metric is the minimum achieved Pearson correlation coefficient between prediction and truth, where the correlation coefficient is given as a function of channel height. × Grid-search evaluation point 35
3.9 Grid search between neurons and layers for the $\mu^k \neq 0$ problem with L_2 data set for shape function $q = 1$. Measured metric is the minimum achieved Pearson correlation coefficient between prediction and truth, where the correlation coefficient is given as a function of channel height. × Grid-search evaluation point 36
3.10 Average elapsed user time per ANN evaluation using the in-house solver's MLP-ANN object for shape function $q = 1$. Average based on 10 000 different inputs using a single CPU core per ANN call. Number of hidden layers given on the right margin. 37

4.1 Training history of the neural network associated to shape function $q = 1$. The four minor ticks in each decade are evenly spaced. — Closure-term loss, — Force-free weak-residual loss, — Solver-adherence penalty metric, — Energy penalty metric 38
4.2 Closure term prediction against truth values for shape function $q = 1$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes 40
4.3 Closure term prediction against truth values for shape function $q = 2$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes 41
4.4 Closure term energy transfer prediction against truth values for shape function $q = 1$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes 42
4.5 Closure term energy transfer prediction against truth values for shape function $q = 2$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes 43
4.6 Channel plot of closure term correlations for the $q = 1$ and $q = 2$ networks. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term 44
4.7 Channel plot of closure term fractional deviation for the $q = 1$ and $q = 2$ networks. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term 45
4.8 Channel plot of closure term correlations for the $q = 1$ and $q = 2$ networks with data augmentation and without data augmentation. Networks trained on the L_2 data set. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term 47

5.1 Net kinetic energy within the TCF domain. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22] 49
5.2 A posterori $\overline{u}$ mean profile as a function of channel height. Shaded region denotes ± 1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22] 50
5.3 A posterori $\overline{v}$ mean profile as a function of channel height. Shaded region denotes ± 1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22] 51

5.4 A posterori $\overline{w}$ mean profile as a function of channel height. Shaded region denotes ± 1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22] 52
5.5 RMS error time series between the neural-network model solution and projected DNS solution. $\overline{u}$ error, $\overline{v}$ error, $\overline{w}$ error 52
5.6 Maximum and minimum velocity as a function of time. Projected DNS maximum, Neural-network model maximum (with algebraic continuity closure term and bounded input intervals [22]). and represent minimums. 53

List of Tables

3.1 Network outputs associated to the q-th shape function and (i, j, k)-th element. $\checkmark$ indicates that the quantity of interest is used in the solver. 22
3.2 Feature set associated to the q-th shape function and (i, j, k)-th element. $\checkmark$ indicates inclusion in feature set and the subscript indicates position in feature set array. 23

Nomenclature

Acronyms

Symbol	Description
ANN	Artificial neural network
CPI	Corrector-pass instability
CPU	Central processing unit
DHIT	Decaying homogeneous isotropic turbulence
DNS	Direct numerical simulation
GPU	Graphics processing unit
GRU	Gated recurrent unit
HIT	Homogeneous isotropic turbulence
LES	Large eddy simulation
LTI	Long-term instability
MLP	Multi-layer perceptron
RAM	Random-access memory
RANS	Reynolds averaged Navier-Stokes
RMS	Root mean square
SGS	Subgrid-scale
TCF	Turbulent channel flow
TPU	Tensor processing unit
VMS	Variational multiscale

Dimensionless numbers

Symbol	Description	Definition
Re_λ	Taylor microscale Reynolds number	$\frac{\lambda_g \sqrt{\frac{2}{3}k}}{\nu}$
Re_τ	Friction Reynolds number	$\frac{u_\tau \delta}{\nu} = \frac{1}{\nu}$
CFL	Courant–Friedrichs–Lewy number (Courant number)	$\frac{u\Delta t}{\Delta x} + \frac{v\Delta t}{\Delta y} + \frac{w\Delta t}{\Delta z}$

Latin symbols

Symbol	Description	Dimensions	Units
$\underline{a}$, a	Velocity basis function coefficients array, Velocity basis function coefficient	–	–
$\mathbf{B}_1(\cdot,\cdot)$	Bilinear form	–	–
$\mathbf{B}_2(\cdot,\cdot,\cdot)$	Trilinear form	–	–
$\underline{b}$	Pressure basis function coefficients array	–	–
$\underline{c}$	Velocity-pressure basis coefficients array pair	–	–
d	Wall-normal distance	L^1	–
$\underline{f}$	Body acceleration/forcing vector	LT^{-2}	–
i, j, k	x-, y-, z-index	–	–
$\underline{\underline{I}}$	Identity tensor	–	–
k	(Net) kinetic energy	ML^2T^{-2}	–
$\mathcal{L}(\cdot)$	Differential operator	–	–
$L(\cdot)$	Linear form	–	–
L_x, L_y, L_z	Physical domain x, y, z length	–	–
M_x, M_y, N_z	Modes in the x, y direction, number of cells in the z direction	–	–
n	Time step index	–	–
N_b	Total number of basis functions	–	–
N_e	Batch size	–	–
n_e	Epoch	–	–
p	Static pressure	$ML^{-1}T^{-2}$	–
$\mathcal{R}(\cdot)$	Force-free weak residual	–	–
$\mathsf{Res}_s(\cdot)$	Strong residual	–	–
$\mathsf{Res}_w(\cdot)$	Weak residual	–	–
t	Time	T	–
$\underline{u}$	Velocity vector with components $(u, v, w)^T$	LT^{-1}	–
$\underline{U}, \underline{W}$	Velocity-pressure pair, momentum-continuity weighting function pair	–	–
$\mathcal{V}_s, \mathcal{V}_\omega$	Trial solution space, weighting function space	–	–
$\underline{w}$, q	Momentum, continuity weighting functions	–	–

x, y, z	Streamwise, spanwise, wall-normal coordinates	L	–

Greek symbols

Symbol	Description	Dimensions	Units
β	Growth factor	–	–
ε	Constraint function / minimum function	–	–
κ	Newton iteration number	–	–
λ	Lamé first parameter (dilatational viscosity coefficient)	$ML^{-1}T^{-1}$	–
λ^k	k-th iterate Lagrangian multiplier estimate	–	–
μ	Lamé second parameter (dynamic/shear viscosity coefficient)	$ML^{-1}T^{-1}$	–
μ^k	k-th iterate penalty coefficient	–	–
ν	Kinematic viscosity coefficient	L^2T^{-1}	–
$\underline{\boldsymbol{\phi}}, \psi$	Velocity, pressure shape functions	LT^{-1}, $ML^{-1}T^{-2}$	–
$\underline{\boldsymbol{\Phi}}, \Psi$	Velocity, pressure basis functions	LT^{-1}, $ML^{-1}T^{-2}$	–
$\boldsymbol{\tau}$	Stabilisation parameter	–	–

Superscripts

Symbol	Description
M, C	Momentum, Continuity
n	Time step index
$\overline{(\cdot)}$	Coarse scales
$(\cdot)'$	Fine scales
$\widetilde{(\cdot)}$	Numerical approximation

Subscripts

Symbol	Description
i, j, k	x-, y-, z-index
κ	κ-Newton iteration number
prediction	prediction
q	q-th shape function
truth	truth

1 Introduction

The prevalence of complex fluid-flow numerical studies is increasing in both academia and industry due to complex requirements, enabled by both hardware and software advents. However, the prohibitive computational costs associated with high mesh density from fine-scale phenomena considerations forces the use of coarsely-discretised numerical simulations with models that account for the effects of the unresolved scales. As a consequence, the captured solution fidelity is limited by geometrical considerations and the models used. Despite the decrease in solution fidelity, coarse-scale formations are responsible for loads and performance behaviour, therefore simulations in which a spectra of coarse scales are numerically resolved are still carried out.

The traditional filtered equations for large eddy simulation (LES) is a popular model that resolves a spectra of coarse scales. One caveat is that the filtered equations require that the (convolution) filtering operation commutes with spatial differentiation. This commutation occurs when neighbouring elements are of equal size, therefore grids for wall-bounded flows are automatically incompatible with standard filtering techniques [1]. Meanwhile, flows bounded by complex geometry require non-uniform filters, and the presence of filter width variations introduces additional energy transfer mechanisms between scales [2]. This is problematic as resolved structures from an element with a small filter width suddenly become a subgrid feature in an element with a larger filter width [2]. The reverse situation is also true, where subgrid scales in elements with a larger filter width can contribute to the formation of resolved structures in elements with a smaller filter width [2]. These two issues are circumvented with the use of the variational multiscale (VMS) framework as an a priori separation of scales is enforced, separating the system of equations into a coarse-scale system and a fine-scale system that are coupled together. The coarse-scale system is numerically resolved, and the fine-scale *solution* is analytically modelled with the use of fine-scale Greens' functions. By utilising appropriate cubature routines, the only error within the VMS framework are related to the fine scales, thereby influencing the coarse-scale system via the closure terms (the coupling terms). It is of practical importance that the coupling terms between the two systems are as close as possible to the truth for accurate projected flows, and not necessarily the fine-scale solution itself. Enabled by hardware and software advents, and ease of access to large DNS data sets, data-driven closure term models are relevant as they can be assumption free to a certain extent, i.e. models which are free of model errors.

To the author's knowledge, there is a distinct lack of work that addresses data-driven closure term modelling in the VMS framework for the Navier-Stokes equations or its derivatives. Most of the works are strictly limited to master thesis works at Delft University of Technology starting from 2019. The earliest works are from Janssens [3] and Robijns [4], both coupling a multi-layer perceptron (MLP) to the coarse-scale VMS Burgers' equation, with Janssens formalising his work into [5]. Janssens [3] determined the instabilities that occurred when utilising the artificial neural network (ANN), while Robijns [4] investigated

the impact of the various terms that were predicted by the ANN. Janssens [3] concluded that two key instabilities existed. One instability was related to the existence of spurious corrector-pass roots when solving for the (implicit) weak residual of the Burgers' equations. The other instability was connected to the inability of the ANN to maintain flow statistics over long intervals of time. The two identified instabilities are known as corrector-pass instability (CPI) and long-term instability (LTI) respectively. Robijns [4] showed that the ANN outputs have to include all small-scale quantities at large Courant numbers to retain flow statistics, but that the small-scale velocity time derivative is negligible at small Courant numbers.

CPI and LTI were addressed by Pusuluri [6] via training data noise augmentation, and by Rajampeta [7] by introducing time-explicit feature sets and backscatter limiting. Pusuluri [6] showed that the ANN model possessed improved corrector-pass and long-term stability by introducing white noise and Gaussian noise to the training data. It was concluded that Gaussian noise was preferable over white noise, allowing for better closure term predictions. Rajampeta [7] successfully shifted spurious roots away from the initial guess when using a feature set dependent only on the previous time steps, allowing for improved corrector pass stability. Backscatter limiting addressed LTI by disallowing any increase to the system's total energy, successfully stopping energy divergence due to the ANN model.

The latest work related to data-driven closure term modelling was by Kanala [8], who shifted the attention of the modelling efforts from one to three dimensional. Kanala [8] constructed a MLP to predict the closure terms of a turbulent channel flow (TCF) driven by the two-scale Navier-Stokes equations under constant forcing. While having provided high correlations between prediction and truth in a priori tests, Kanala [8] provided no a posteriori analysis.

This thesis is a direct continuation of the data-driven modelling efforts for three-dimensional problems. An emphasis on energy stability was sought as the issue was resolved in one dimension by Rajampeta [7] but not addressed in three dimensions by Kanala [8]. As this thesis will be foundational for future efforts, the scope was limited to constructing strong a priori correlations and possibly providing a stabilised a posteriori simulation. The data-driven model consisted of eight MLPs working together, and was benchmarked against a $\mathrm{Re}_\tau = 180$ TCF flow. The thesis goal is summarised as the following research objective:

To construct an energy-stable data-driven model for a $\mathrm{Re}_\tau = 180$ TCF driven by the coarse-scale VMS Navier-Stokes equations by assessment of its long-term numerical stability and accuracy.

1) To identify a spatiotemporal stencil from which to draw features from for the feature set, allowing for improved accuracy of the data-driven model.

with the following research questions formed to achieve the objective:

1) What are the differences and similarities between the latest 1D Navier-Stokes VMS-ANN TCF data-driven models and the 3D Navier-Stokes VMS-ANN TCF data-driven models?
2) What coarse-scale variables are appropriate for the feature set?

3) What spatiotemporal stencil is appropriate for the data-driven model to produce strong correlations between predictions and truth?
4) What technique(s) can be used to make the model energy stable?
5) How strongly can the data-driven model establish a functional relationship between the coarse-scale variables and the closure terms?

The thesis is structured to ultimately answer the research questions. Chapter 2 provides a basic understanding of the two-scale Navier-Stokes equations alongside a basic understanding of neural networks. The way that the machine learning aspect ties into the VMS framework is briefly discussed. Chapter 3 details the network choices for this thesis, putting an emphasis on the cost function as it contains the theory used to provide energy-stable simulations. Chapter 4 is an a priori evaluation of the network, providing correlations between predictions and truth before being used in a simulation. Chapter 5 is the a posteriori evaluation of the networks, where the net energy held within the system is examined, and comparisons of velocity profiles from VMS simulations and projected direct numerical simulation (DNS) are made. Chapter 6 concludes the thesis, providing the answers to the research questions in an explicit format, and recommendations are made.

2 Theory

The work involved in the thesis revolves around three topics: the Navier-Stokes equations, Artificial Neural Networks, and Variational Multiscale Methods. This chapter serves as a basic introduction to the topics for the reader.

2.1 Navier-Stokes Equations

The Navier-Stokes equations are partial differential equations that describe the mass and momentum of fluid packets. The equations read

$$\frac{\partial \rho}{\partial t} = -\underline{\nabla} \cdot \left(\rho \underline{\mathbf{u}}\right) \tag{2.1a}$$

$$\rho \frac{\partial \underline{\mathbf{u}}}{\partial t} + \rho\left(\underline{\mathbf{u}} \cdot \underline{\nabla}\right)\underline{\mathbf{u}} = \underline{\nabla} \cdot \left(-p\underline{\underline{\mathbf{I}}} + \lambda\left(\underline{\nabla} \cdot \underline{\mathbf{u}}\right)\underline{\underline{\mathbf{I}}} + 2\mu\left(\frac{1}{2}\underline{\nabla}\underline{\mathbf{u}} + \frac{1}{2}(\underline{\nabla}\underline{\mathbf{u}})^{\mathsf{T}}\right)\right) + \rho\underline{\mathbf{f}} \tag{2.1b}$$

where ρ is the medium density, $\underline{\mathbf{u}}$ is the medium velocity, p is the medium static pressure, $\underline{\mathbf{f}}$ are the body accelerations, $\underline{\underline{\mathbf{I}}}$ is the identity tensor, and λ and μ are the Lamé first and second parameter respectively.

The thesis utilises the incompressible version of the Navier-Stokes equations, therefore a constant density constraint is applied to Sys. 2.1. By the conservation of mass, the flow must be divergence free, i.e. $\underline{\nabla} \cdot \underline{\mathbf{u}} = 0$. The incompressible Navier-Stokes equations read

$$\underline{\nabla} \cdot \underline{\mathbf{u}} = 0$$

$$\rho \frac{\partial \underline{\mathbf{u}}}{\partial t} + \rho\left(\underline{\mathbf{u}} \cdot \underline{\nabla}\right)\underline{\mathbf{u}} = -\underline{\nabla} p + \mu \underline{\nabla} \cdot \underline{\nabla}\underline{\mathbf{u}} + \rho\underline{\mathbf{f}}$$

where μ is relabelled as the medium dynamic/shear viscosity as the coefficient is applied to the diffusive term. The standard form worked with in this thesis is of the form

$$\underline{\nabla} \cdot \underline{\mathbf{u}} = 0 \tag{2.2a}$$

$$\frac{\partial \underline{\mathbf{u}}}{\partial t} + \left(\underline{\mathbf{u}} \cdot \underline{\nabla}\right)\underline{\mathbf{u}} = -\frac{1}{\rho}\underline{\nabla} p + \underline{\nabla} \cdot \left(2\nu \underline{\nabla}^{s}\underline{\mathbf{u}}\right) + \underline{\mathbf{f}} \tag{2.2b}$$

where $\nu = \mu/\rho$ is the medium kinematic viscosity, and $\underline{\nabla}^{s}\underline{\mathbf{u}} = 1/2\left(\underline{\nabla}\underline{\mathbf{u}} + (\underline{\nabla}\underline{\mathbf{u}})^{\mathsf{T}}\right)$ is the symmetric velocity gradient tensor (strain-rate tensor).

2.2 Artificial Neural Networks

An ANN is a network of interconnected nodes that uses the underlying patterns found in data sets to provide an output when given an input. These networks can be used to classify

information (e.g. whether a tumor is benign or not), or used to regress data (e.g. predict distance travelled by a projectile at some unknown time t). The nodes/neurons are placed in layers, and each neuron takes the outputs from a previous layer as input and feeds it through a function. The new output is then used for the next layer. The first layer which takes user inputs is known as the input layer. The last layer which provides the user outputs is known as the output layer. The remaining layers are known as hidden layers. ANNs are differentiated from one another through their hidden layers. Some neural networks convolve the inputs with kernels before passing it to the nodes' output function, thereby becoming convolutional neural networks. Other neural networks can store the current node outputs to influence future outputs, thereby becoming recurrent neural networks.

One of the simplest type of ANNs are MLPs, in which the outputs of the hidden layers are made by passing the weighted sum of the outputs of the previous layer through a function with a bias. An example schematic of a MLP with n_i input values, n_H neurons, n_o outputs, and one hidden layer is given in Fig. 2.1. The connections between neurons determine where an output of a neuron is used as input for another neuron.

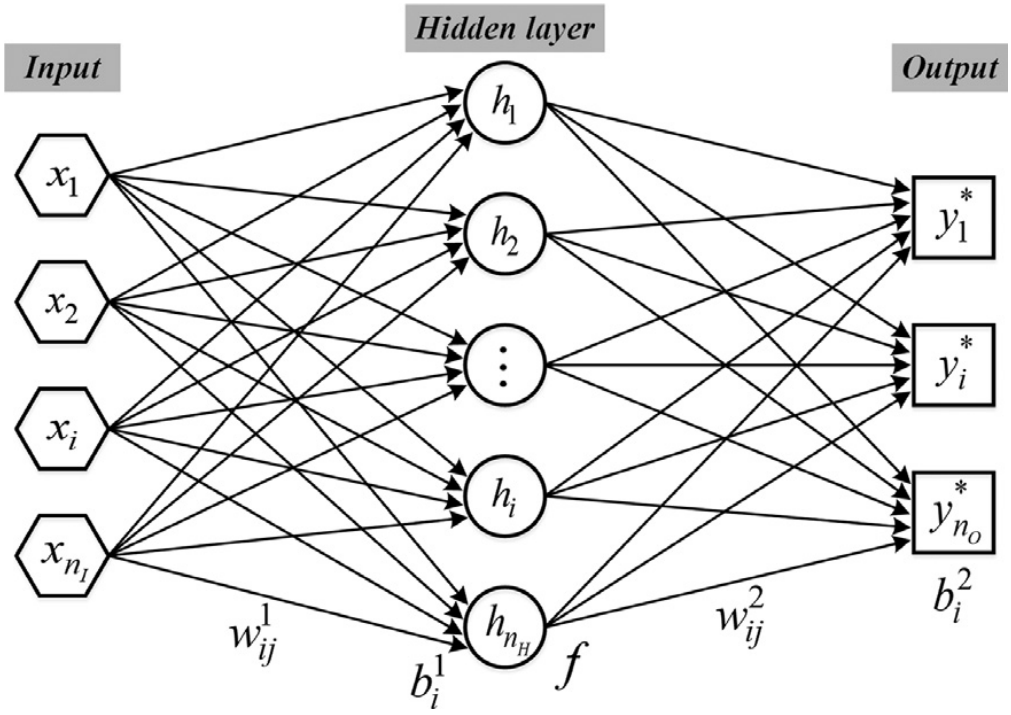


FIGURE 2.1 Single hidden layer ANN schematic [9, Fig. 2.]

As previously mentioned, the output of a neuron in the hidden layer is given by an activation function f with an argument made by a weighted sum of the inputs $w_{ij}x_j$ and their biases b_i:

$$h_i = f\left(\sum_{j=1}^{n_i} w_{ij}^1 x_j + b_i^1\right) \quad \text{or in matrix-vector form:} \quad \underline{\mathbf{h}} = f\left(\mathbf{W}^1\underline{\mathbf{x}} + \underline{\mathbf{b}}^1\right)$$

whereas the output of the ANN is given by

$$y_i = \sum_{j=1}^{n_H} w_{ij}^2 h_j + b_i^2 \quad \text{or in matrix-vector form:} \quad \underline{y} = f\left(W^2 \underline{h} + \underline{b}^2\right)$$

Calibrating the weights require loss functions to be minimised, for which techniques such as backpropagation are used.

The use of machine learning in fluid mechanics has grown in popularity over the past thirty years due to the availability of large data sets from DNS and experiments [10]. With large data sets, data-driven models can learn complex patterns that would not be easily replicable by models with calibrated coefficients. Data-driven models are flexible in the sense that they can be designed to deal with non-stationary statistics and arbitrary inputs [11]. As data-driven modelling is a new discipline within flow numerics, the current challenges exhibited include but are not limited to [10, 11]:

- the high cost of generating training data and/or the lack of data consistency when extracting from existing databases;
- the need to consider the mismatch in software programming languages and hardware usage during the implementation of the data-driven model (software: Python; hardware: GPUs, TPUs) with the flow solver (software: C, C++, Fortran; hardware: CPUs);
- whether the choice of features and physical constraints are correct;
- whether the data-driven model can generalise or not;
- whether the data-driven model can be interpreted or not;
- the impracticality of a posteriori training for LES, forcing model paradigms centred around a priori training.

As no framework exists on how to create the 'correct' data-driven model, it is to this author's opinion that most state-of-the-art research is still exploratory and attempting to signify the capabilities and feasibilities of using neural networks in closure modelling.

2.3 Variational Multiscale Method

The VMS framework is described by Hughes and Sangalli *"... as a framework for incorporating missing fine-scale effects into numerical problems governing coarse-scale behavior"* [12]. In the application to the Navier-Stokes equations, VMS is most analogous to LES, solving for flow phenomena beyond the mean field provided by Reynolds-Averaged Navier-Stokes (RANS), but in lower-fidelity when compared to DNS. The general VMS methodology is to define the solution of a system of partial differential equations as the sum of a coarse-scale solution and a fine-scale solution, with the coarse scales defined as a projection of the solution. The fine-scales are analytically solved with the use of the *fine-scale Greens' operator*, while the coarse scales are solved numerically.

For the purpose of this thesis, the VMS equations for the finite-element Navier-Stokes equations are required, therefore the derivation in this section is heavily inspired from the works of Hughes, Scovazzi, and Franca [13] and Bazilevs et al. [14]. Consider a Galerkin spacetime formulation of the incompressible Navier-Stokes equations (see Eqs (2.2a–2.2b))

over a spacetime domain $Q = \Omega \times [0, T] \subset \mathbb{R}^3 \times \mathbb{R}^+$ with boundaries $P = \Gamma \times [0, T]$. This domain is subdivided into N spacetime slabs $Q_n = \Omega \times [t_n, t_{n+1}]$, $n = 0, 1, 2, 3, \cdots, N$, i.e. $Q = \bigcup_{n=0}^{N} Q_n$, and similarly for $P = \bigcup_{n=0}^{N} P_n$, visualised in Fig. 2.2.

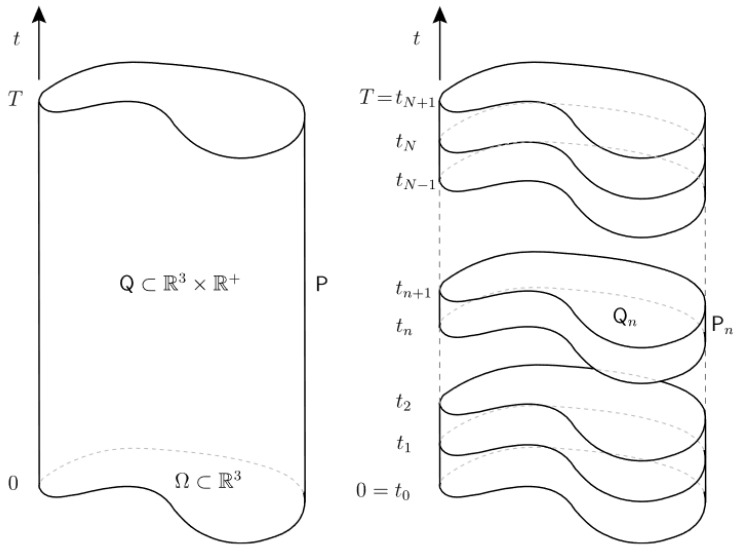


FIGURE 2.2 left: Spacetime domain, right: spacetime slabs [14, Fig. 1.]

The objective $\forall Q_n$ is to find $\underline{U} = \{\underline{u}, p\} \in \mathcal{V}_s(Q_n)$ given an initial condition $\underline{u}_0$ in Q_0 such that $\forall \underline{W} = \{\underline{w}, q\} \in \mathcal{V}_\omega(Q_n)$:

$$\mathbf{B}_1(\underline{W}, \underline{U}) + \mathbf{B}_2(\underline{W}, \underline{U}, \underline{U}) - \mathbf{L}(\underline{W}) = \underline{0}$$

where $\mathcal{V}_s$ and $\mathcal{V}_\omega$ are the trial solution space and weighting function space defined over the spacetime domain Q_n, $\mathbf{B}_1$ is a bilinear form, $\mathbf{B}_2$ is a trilinear form, and

$$\mathbf{B}_1(\underline{W}, \underline{U}) = \left(\underline{w}, \frac{\partial \underline{u}}{\partial t} + \frac{1}{\rho}\underline{\nabla} p - \underline{\nabla} \cdot (2\nu \underline{\nabla}^s \underline{u})\right)_{Q_n} + \left(q, \underline{\nabla} \cdot \underline{u}\right)_{Q_n}$$

$$\mathbf{B}_2(\underline{W}, \underline{U}, \underline{V}) = \left(\underline{w}, (\underline{u} \cdot \underline{\nabla})\underline{v}\right)_{Q_n}$$

$$\mathbf{L}(\underline{W}) = \left(\underline{w}, \underline{f}\right)_{Q_n}$$

with $\underline{V} = \{\underline{v}, \cdot\}$. A separation of scales is made via a direct sum decomposition of $\mathcal{V}_s$:

$$\mathcal{V}_s = \overline{\mathcal{V}}_s \oplus \mathcal{V}'_s$$

where $\overline{\mathcal{V}}_s$ is the coarse-scale trial solution space and represented in a finite-dimensional space, and $\mathcal{V}'_s$ is the fine-scale trial solution space and represented in an infinite-dimensional space. By letting $\underline{u} = \overline{\underline{u}} + \underline{u}'$ and $p = \overline{p} + p'$, where $\overline{\underline{U}} = \{\overline{\underline{u}}, \overline{p}\} \in \overline{\mathcal{V}}_s$ and $\underline{U}' = \{\underline{u}', p'\} \in \mathcal{V}'_s$ (and $\underline{U} = \overline{\underline{U}} + \underline{U}'$), it is known by the linearity of the forms that:

$$\mathbf{B}_1(\underline{W}, \overline{\underline{U}} + \underline{U}') + \mathbf{B}_2(\underline{W}, \overline{\underline{U}} + \underline{U}', \overline{\underline{U}} + \underline{U}') - \mathbf{L}(\underline{W}) = \underline{0}$$

is equivalent to

$$\begin{aligned}\mathtt{B}_1(\underline{W},\overline{\underline{U}})+\mathtt{B}_2(\underline{W},\overline{\underline{U}},\overline{\underline{U}})-\mathtt{L}(\underline{W})=\\ -\mathtt{B}_1(\underline{W},\underline{U}')-\mathtt{B}_2(\underline{W},\overline{\underline{U}},\underline{U}')-\mathtt{B}_2(\underline{W},\underline{U}',\overline{\underline{U}})-\mathtt{B}_2(\underline{W},\underline{U}',\underline{U}')\end{aligned}$$

To numerically resolve the given weighted residual problem, a separation of scales for $\mathcal{V}_\omega$ is made via a direct sum decomposition:

$$\mathcal{V}_\omega=\overline{\mathcal{V}}_\omega\oplus\mathcal{V}'_\omega$$

where like before, $\overline{\mathcal{V}}_\omega$ is the coarse-scale weighting function space and represented in a finite-dimensional space, and $\mathcal{V}'_\omega$ is the fine-scale weighting function space and represented in an infinite-dimensional space. By letting $\underline{w}=\overline{\underline{w}}+\underline{w}'$ and $q=\overline{q}+q'$, where $\overline{\underline{W}}=\{\overline{\underline{w}},\overline{q}\}\in\overline{\mathcal{V}}_\omega$ and $\underline{W}'=\{\underline{w}',q'\}\in\mathcal{V}'_\omega$ (and $\underline{W}=\overline{\underline{W}}+\underline{W}'$), it is trivially known by linearity of the forms that

$$\begin{aligned}\mathtt{B}_1\left(\overline{\underline{W}}+\underline{W}',\overline{\underline{U}}\right)+\mathtt{B}_2\left(\overline{\underline{W}}+\underline{W}',\overline{\underline{U}},\overline{\underline{U}}\right)-\mathtt{L}\left(\overline{\underline{W}}+\underline{W}'\right)=\\ -\mathtt{B}_1\left(\overline{\underline{W}}+\underline{W}',\underline{U}'\right)-\mathtt{B}_2\left(\overline{\underline{W}}+\underline{W}',\overline{\underline{U}},\underline{U}'\right)-\mathtt{B}_2\left(\overline{\underline{W}}+\underline{W}',\underline{U}',\overline{\underline{U}}\right)-\mathtt{B}_2\left(\overline{\underline{W}}+\underline{W}',\underline{U}',\underline{U}'\right)\end{aligned}$$

is equal to

$$\begin{aligned}\mathtt{B}_1(\overline{\underline{W}},\overline{\underline{U}})+\mathtt{B}_2(\overline{\underline{W}},\overline{\underline{U}},\overline{\underline{U}})-\mathtt{L}(\overline{\underline{W}})=\\ -\mathtt{B}_1(\overline{\underline{W}},\underline{U}')-\mathtt{B}_2(\overline{\underline{W}},\overline{\underline{U}},\underline{U}')-\mathtt{B}_2(\overline{\underline{W}},\underline{U}',\overline{\underline{U}})-\mathtt{B}_2(\overline{\underline{W}},\underline{U}',\underline{U}')\\ -\mathtt{B}_1(\underline{W}',\overline{\underline{U}})-\mathtt{B}_2(\underline{W}',\overline{\underline{U}},\overline{\underline{U}})-\mathtt{B}_1(\underline{W}',\underline{U}')-\mathtt{B}_2(\underline{W}',\overline{\underline{U}},\underline{U}')-\mathtt{B}_2(\underline{W}',\underline{U}',\overline{\underline{U}})-\mathtt{B}_2(\underline{W}',\underline{U}',\overline{\underline{U}})+\mathtt{L}(\underline{W}')\end{aligned}$$

and by rearrangement, two coupled systems of partials differential equations are returned:

$$\begin{aligned}&\overbrace{\mathtt{B}_1(\overline{\underline{W}},\overline{\underline{U}})+\mathtt{B}_2(\overline{\underline{W}},\overline{\underline{U}},\overline{\underline{U}})-\mathtt{L}(\overline{\underline{W}})=-\mathtt{B}_1(\overline{\underline{W}},\underline{U}')-\mathtt{B}_2(\overline{\underline{W}},\overline{\underline{U}},\underline{U}')-\mathtt{B}_2(\overline{\underline{W}},\underline{U}',\overline{\underline{U}})-\mathtt{B}_2(\overline{\underline{W}},\underline{U}',\underline{U}')}^{\text{coarse-scale system}}\\ &\underbrace{\mathtt{B}_1(\underline{W}',\underline{U}')+\mathtt{B}_2(\underline{W}',\underline{U}',\underline{U}')-\mathtt{L}(\underline{W}')=-\mathtt{B}_1(\underline{W}',\overline{\underline{U}})-\mathtt{B}_2(\underline{W}',\overline{\underline{U}},\underline{U}')-\mathtt{B}_2(\underline{W}',\underline{U}',\overline{\underline{U}})-\mathtt{B}_2(\underline{W}',\overline{\underline{U}},\overline{\underline{U}})}_{\text{fine-scale system}}\end{aligned}$$

Thus far, the system of equations are exact but non-unique in the sense that $\overline{\underline{U}}$, $\underline{U}'$ and $\overline{\underline{W}}$, $\underline{W}'$ are only defined to exist within their respective spaces. To provide a unique definition for $\overline{\underline{U}}$, $\underline{U}'$ and $\overline{\underline{W}}$, $\underline{W}'$, the coarse-scale quantities $\overline{\underline{U}}$ and $\overline{\underline{W}}$ are defined to be the respective projections of $\underline{U}$ and $\underline{W}$ onto their spaces given some user-defined norm $\hat{\mathtt{N}}$, i.e.

$$\overline{\underline{U}}=\mathsf{proj}_{\hat{\mathtt{N}}}^{\overline{\mathcal{V}}_s}\,\underline{U}$$

$$\overline{\underline{W}}=\mathsf{proj}_{\hat{\mathtt{N}}}^{\overline{\mathcal{V}}_\omega}\,\underline{W}$$

from which the trivial definition for the small scale quantities can be identified:

$$\underline{U}'=\underline{U}-\overline{\underline{U}}=\left(\mathsf{iden}-\mathsf{proj}_{\hat{\mathtt{N}}}^{\overline{\mathcal{V}}_s}\right)\underline{U}$$

$$\underline{W}'=\underline{W}-\overline{\underline{W}}=\left(\mathsf{iden}-\mathsf{proj}_{\hat{\mathtt{N}}}^{\overline{\mathcal{V}}_\omega}\right)\underline{W}$$

where iden is the identity operator, i.e. $\mathsf{iden}\,(\underline{\mathbf{u}}) = \underline{\mathbf{u}}$. Rewriting the fine-scale system into the following form reveals that it is driven by the strong residual of $\overline{\underline{\mathbf{u}}}$ in addition to $\overline{\underline{\mathbf{u}}}$ itself:

$$\mathbf{B}_1(\underline{\mathbf{w}}', \underline{\mathbf{u}}') + \mathbf{B}_2(\underline{\mathbf{w}}', \underline{\mathbf{u}}', \underline{\mathbf{u}}') = -\mathbf{B}_2(\underline{\mathbf{w}}', \overline{\underline{\mathbf{u}}}, \underline{\mathbf{u}}') - \mathbf{B}_2(\underline{\mathbf{w}}', \underline{\mathbf{u}}', \overline{\underline{\mathbf{u}}}) - (\underline{\mathbf{w}}', \mathsf{Res}_s(\overline{\underline{\mathbf{u}}}))$$

where $(\underline{\mathbf{w}}', \mathsf{Res}_s(\overline{\underline{\mathbf{u}}})) = \mathbf{B}_1(\underline{\mathbf{w}}', \overline{\underline{\mathbf{u}}}) + \mathbf{B}_2(\underline{\mathbf{w}}', \overline{\underline{\mathbf{u}}}, \overline{\underline{\mathbf{u}}}) - \mathbf{L}(\underline{\mathbf{w}}')$. As such, the formal solution for $\underline{\mathbf{u}}'$ can be rewritten as a functional $\mathbf{F}$ of the strong residual of $\overline{\underline{\mathbf{u}}}$ and $\overline{\underline{\mathbf{u}}}$ itself, i.e. $\underline{\mathbf{u}}' = \mathbf{F}(\overline{\underline{\mathbf{u}}}, \mathsf{Res}_s(\overline{\underline{\mathbf{u}}}))$. Consequentially the fine-scale system becomes unnecessary if the functional exists, and the problem is reduced to the coarse-scale system:

$$\mathbf{B}_1(\overline{\underline{\mathbf{w}}}, \overline{\underline{\mathbf{u}}}) + \mathbf{B}_2(\overline{\underline{\mathbf{w}}}, \overline{\underline{\mathbf{u}}}, \overline{\underline{\mathbf{u}}}) - \mathbf{L}(\overline{\underline{\mathbf{w}}}) = -\mathbf{B}_1(\overline{\underline{\mathbf{w}}}, \underline{\mathbf{u}}') - \mathbf{B}_2(\overline{\underline{\mathbf{w}}}, \overline{\underline{\mathbf{u}}}, \underline{\mathbf{u}}') - \mathbf{B}_2(\overline{\underline{\mathbf{w}}}, \underline{\mathbf{u}}', \overline{\underline{\mathbf{u}}}) - \mathbf{B}_2(\overline{\underline{\mathbf{w}}}, \underline{\mathbf{u}}', \underline{\mathbf{u}}')$$

where $\underline{\mathbf{u}}' = \mathbf{F}(\overline{\underline{\mathbf{u}}}, \mathsf{Res}_s(\overline{\underline{\mathbf{u}}}))$. Bazilevs et al. [14] utilises a naive perturbation series as the functional, where each order term can be inverted via a Greens' operator. However, it is noted that the Greens' operator is not exactly determinable (and summation of an infinite number of terms is not possible); therefore practical calculations of $\underline{\mathbf{u}}'$ require the approximation of the functional $\tilde{\mathbf{F}}$, in which case only the approximate solution of the following problem is solved:

$$\mathbf{B}_1\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}\right) + \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}, \widetilde{\overline{\underline{\mathbf{u}}}}\right) - \mathbf{L}\left(\overline{\underline{\mathbf{w}}}\right) = -\mathbf{B}_1\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}'\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}, \widetilde{\underline{\mathbf{u}}}'\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}', \widetilde{\overline{\underline{\mathbf{u}}}}\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}', \widetilde{\underline{\mathbf{u}}}'\right) \quad (2.3)$$

where $\widetilde{\underline{\mathbf{u}}}' = \tilde{\mathbf{F}}(\widetilde{\overline{\underline{\mathbf{u}}}}, \mathsf{Res}_s(\widetilde{\overline{\underline{\mathbf{u}}}}))$ and $\widetilde{\overline{\underline{\mathbf{u}}}}$ is the approximation of $\overline{\underline{\mathbf{u}}}$. This is known as the two-scale approach to the incompressible Navier-Stokes equations, and the right hand side of the system are the closure terms. The standard functional approximation yields the following relation for $\widetilde{\underline{\mathbf{u}}}'$:

$$\widetilde{\underline{\mathbf{u}}}' \approx -\boldsymbol{\tau}\, \mathsf{Res}_s(\widetilde{\overline{\underline{\mathbf{u}}}})$$

where $\boldsymbol{\tau}$ is a stabilisation parameter defined to be an element's mean value of the (approximate) fine-scale Green's function.

2.4 ANN-Coupled VMS Methods

It was aforementioned in Sec. 2.3 that solving the two-scale incompressible Navier-Stokes equations in a practical setting requires an approximation of $\mathbf{F}$ in order to calculate $\widetilde{\underline{\mathbf{u}}}'$. By utilising appropriate discretisation methods and cubature rules, the only errors for $\overline{\underline{\mathbf{u}}}$ within this framework are model errors stemming from the functional approximation, and round-off errors. For the standard approximation, the error becomes severe when large elements are used, since the averaging of the approximated Green's function prevents the consideration of the variation of $\mathsf{Res}_s(\overline{\underline{\mathbf{u}}})$ over the element. More importantly, the fine-scale Greens' function is not necessarily localised to a single element, implying that neighbouring elements may also determine local fine-scale quantities. To bypass this modelling error, the approach of this thesis is to predict all closure terms via neural networks. This is best illustrated by labelling Eq. 2.3:

$$\overbrace{\mathbf{B}_1\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}\right) + \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}, \widetilde{\overline{\underline{\mathbf{u}}}}\right) - \mathbf{L}\left(\overline{\underline{\mathbf{w}}}\right)}^{\text{Numerically resolved}} = \underbrace{-\mathbf{B}_1\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}'\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\overline{\underline{\mathbf{u}}}}, \widetilde{\underline{\mathbf{u}}}'\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}', \widetilde{\overline{\underline{\mathbf{u}}}}\right) - \mathbf{B}_2\left(\overline{\underline{\mathbf{w}}}, \widetilde{\underline{\mathbf{u}}}', \widetilde{\underline{\mathbf{u}}}'\right)}_{\text{ANN predicted}}$$

Since no (explicit) assumptions are made by utilising a data-driven approach, the only error that manifests in the solution to Eq. 2.3 are neural-network related, which are influenced by the hyperparameters of the neural network. This data-driven approach is made possible by the:

- availability of large DNS data sets, and the capabilities to manipulate these data sets due to hardware performance;
- and existence of a functional which relates the fine scales to the coarse scales via the coarse-scale quantities.

From a subjective perspective, the success of Rajampeta [7] in ANN modelling hints at the potential success for the full Navier-Stokes equations. Rajampeta effectively[1] modelled the closure terms of the x-momentum equation with a MLP which uses three hidden layers. The last two layers used 128 neuron each, and the first layer used neurons equal to the number of inputs. The networks were trained on a total of 781 L_2 projected DNS time steps $(\Delta t = 0.0064)$, and took inputs from various elements in the direction of the upwind characteristic (the element for which the closure terms are computed for, two upstream elements, and one downstream element). The considered inputs were taken from a total of three time levels, and the outputs were all the closure terms in correspondence to their shape function per element. Rajampeta's network produced a net kinetic energy time series that was conservative in comparison to the true L_2 projected data when utilising backscatter limiting, but the a posteriori velocity solution profile was still comparable to the exact projection after 781 time steps.

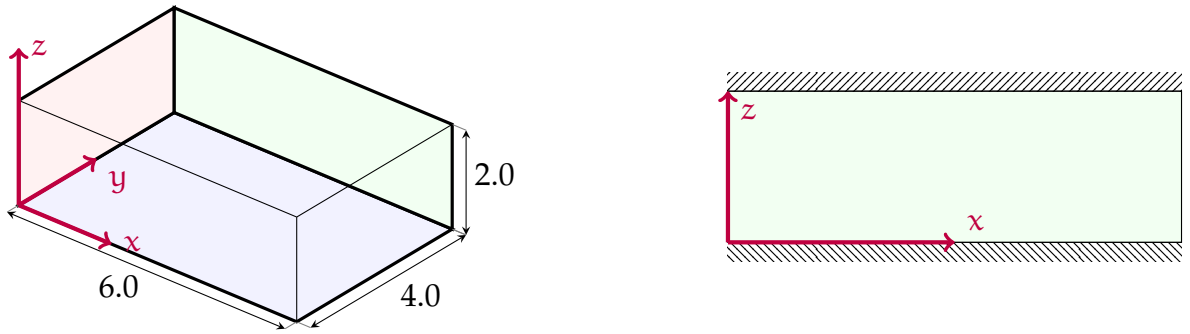
[1] Rajampeta's work explicitly reformulates the x-momentum equation into the 1D Burgers' equation.

3 ANN Construction

This chapter serves to provide the reader an outlook on the parameters of the neural networks used for the prediction of the closure terms associated to the $Re_\tau = 180$ TCF test case. Information surrounding the feature set and the network architecture is also provided. The choices made in this chapter are the accumulation of many neural-network iterations.

3.1 Problem Statement

This thesis focuses on the construction of neural networks capable of predicting the closure terms belonging to a $Re_\tau = 180$ TCF simulated at a Courant number equal to 1.0. The TCF domain is given by dimensions $L_x \times L_y \times L_z = 6.0 \times 4.0 \times 2.0$, where x, y, z are the streamwise, spanwise, and wall-normal directions respectively. A constant forcing $\underline{f} = (1, 0, 0)^T$ is imposed (constant forcing in the streamwise direction). The discretised indexing is given by (i, j, k)-notation, where i, j, k are the streamwise, spanwise, and wall-normal indices respectively. The domain is visualised in Fig. 3.1, and is both x- and y-periodic. The domain is discretised into a uniform Cartesian $32 \times 32 \times 32$ mesh, yielding first layer centroids in the viscous sublayer / buffer layer ($z^+ \approx 5.45$). The simulation test case assumes solutions to be composed from a linear sum of basis functions, where these functions are trilinear in x, y and z. Temporally, these basis functions are Dirac deltas centred around $n+1$, where n is the current time step. The time derivative is approximated using the backwards Euler method, in contrast to Rajampeta's test case, which utilises a three-point implicit approximation.



(A) Three dimensional visualisation of the TCF domain.

(B) z-x plane visualisation. Shaded regions indicate wall boundary.

FIGURE 3.1 Example visualisation of the TCF domain. The utilised face colours are used to distinguish form orientation.

3.2 Feature Set Selection

The features used are curated to be dependent on already-calculated solver quantities. This prevents the use of post-processing routines in extracting features from the in-house solver. Additionally, the feature set is selected amongst quantities across multiple elements, where the selection of elements forms a stencil. This section describes the feature set and the relevant choices made to establish it. This includes a description on the coupling between the neural network and in-house solver, training data projection norm, feature correlations and choices, and data augmentation.

3.2.1 Solver Coupling

Section 2.4 introduced the required terms to be predicted by a neural network, but does not discuss the method used to solve the system of equations when these predictions are provided. The in-house MEX INS weak-residual solver was utilised for this thesis, which solves the primal monolithic problem with finite-element methods in space and finite-difference methods in time. An indepth consideration of the manipulated weak-form terms that the in-house solver uses is not provided due to discussion irrelevancy. Therefore, the following description gives the solver coupling at a top-level view.

Consider the equivalent weak form found by the decomposition of Eq. 2.3 into its respective equations:

$$B_1^M\left(\overline{\underline{W}},\widetilde{\underline{U}}\right)+B_2\left(\overline{\underline{W}},\widetilde{\underline{U}},\widetilde{\underline{U}}\right)-L\left(\overline{\underline{W}}\right)=-B_1^M\left(\overline{\underline{W}},\widetilde{\underline{U}}'\right)-B_2\left(\overline{\underline{W}},\widetilde{\underline{U}},\widetilde{\underline{U}}'\right)-B_2\left(\overline{\underline{W}},\widetilde{\underline{U}}',\widetilde{\underline{U}}\right)-B_2\left(\overline{\underline{W}},\widetilde{\underline{U}}',\widetilde{\underline{U}}'\right)$$

$$B_1^C\left(\overline{\underline{W}},\widetilde{\underline{U}}\right)=-B_1^C\left(\overline{\underline{W}},\widetilde{\underline{U}}'\right)$$

where

$$\mathbf{B}_1^M(\underline{W},\underline{U})=\left(\underline{w},\frac{\partial\underline{u}}{\partial t}+\frac{1}{\rho}\underline{\nabla}p-\underline{\nabla}\cdot\left(2\nu\underline{\nabla}^s\underline{u}\right)\right)_{Q_n}$$

$$\mathbf{B}_1^C(\underline{W},\underline{U})=\left(q,\underline{\nabla}\cdot\underline{u}\right)_{Q_n}$$

Both the momentum and continuity equations can be rewritten into the following form with the differential operators $\overline{\mathcal{L}}_M$, $\mathcal{L}'_M$ and $\mathcal{L}_C$:

$$\left(\overline{\underline{w}},\overline{\mathcal{L}}_M(\widetilde{\underline{u}},\widetilde{p})\right)-\left(\overline{\underline{w}},\underline{f}\right)=-\left(\overline{\underline{w}},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')\right) \tag{3.1}$$

$$\left(\overline{q},\mathcal{L}_C(\widetilde{\underline{u}})\right)=-\left(\overline{q},\mathcal{L}_C(\widetilde{\underline{u}}')\right) \tag{3.2}$$

where

$$\left(\overline{\underline{w}},\overline{\mathcal{L}}_M(\widetilde{\underline{u}},\widetilde{p})\right)=\mathbf{B}_1^M\left(\overline{\underline{W}},\widetilde{\underline{U}}\right)+\mathbf{B}_2\left(\overline{\underline{W}},\widetilde{\underline{U}},\widetilde{\underline{U}}\right)$$

$$\left(\overline{\underline{w}},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')\right)=\mathbf{B}_1^M\left(\overline{\underline{W}},\widetilde{\underline{U}}'\right)+\mathbf{B}_2\left(\overline{\underline{W}},\widetilde{\underline{U}},\widetilde{\underline{U}}'\right)+\mathbf{B}_2\left(\overline{\underline{W}},\widetilde{\underline{U}}',\widetilde{\underline{U}}\right)+\mathbf{B}_2\left(\overline{\underline{W}},\widetilde{\underline{U}}',\widetilde{\underline{U}}'\right)$$

$$\left(\overline{q},\mathcal{L}_C(\underline{u})\right)=\mathbf{B}_1^C\left(\overline{\underline{W}},\underline{U}\right)$$

The weak residual equations are iteratively solved, with the κ-th iteration defined by providing time indices to Eqs. (3.1–3.2), alongside choosing the weighting functions to be the solution basis functions $\underline{\overline{\Phi}}$ and $\overline{\Psi}$:

$$\left(\underline{\overline{\Phi}}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})\right)_\kappa^{n+1} + \underbrace{\left(\underline{\overline{\Phi}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\right)^{n+1}}_{\text{predicted}} - \left(\underline{\overline{\Phi}}, \underline{f}\right)^{n+1} = \left(\underline{\overline{\Phi}}, \mathsf{Res}_w^M(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')\right)_\kappa^{n+1} \tag{3.3}$$

$$\left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}})\right)_\kappa^{n+1} + \underbrace{\left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}}')\right)^{n+1}}_{\text{predicted}} = \left(\overline{\Psi}, \mathsf{Res}_w^C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)_\kappa^{n+1} \tag{3.4}$$

The left-most term of Eqs. (3.3–3.4) are the coarse-scale sums, and are recalculated for each iteration using new estimates of the velocity and pressure field. The forcing term is omitted from recalculation as it is constant, and the closure term predictions are conducted once at $\kappa = 0$ as no corrector-pass information is needed, justified by choices covered in Sec. 3.2.2. The initial guess for the coarse-scale sum is calculated with the coarse scale solution at t^n, i.e.

$$\left(\underline{\overline{\Phi}}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})\right)_{\kappa=0}^{n+1} = \left(\underline{\overline{\Phi}}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})\right)^n$$

$$\left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}})\right)_{\kappa=0}^{n+1} = \left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}})\right)^n$$

and the Newton-Raphson method is used to solve for the root of the κ-th combined residual

$$\mathsf{Res}_w(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')_\kappa^{n+1} := \left\{ \left(\underline{\overline{\Phi}}, \mathsf{Res}_w^M(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')\right)_\kappa^{n+1}, \left(\overline{\Psi}, \mathsf{Res}_w^C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)_\kappa^{n+1} \right\}$$

on any time slab Q_n. Each Newton iteration is standard, i.e. the following operations are evaluated in order to advance from κ to $\kappa + 1$:

$$-\frac{\partial\left(\mathsf{Res}_w(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')_\kappa^{n+1}\right)}{\partial \widetilde{\underline{c}}} \Delta \widetilde{\underline{c}}_\kappa = \mathsf{Res}_w(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')_\kappa^{n+1}$$

$$\widetilde{\underline{c}}_{\kappa+1}^{n+1} = \widetilde{\underline{c}}_\kappa^{n+1} + \Delta \widetilde{\underline{c}}_\kappa$$

where $\widetilde{\underline{c}}_\kappa := \{\widetilde{\underline{a}}_\kappa, \widetilde{\underline{b}}_\kappa\}$ are the weighting coefficients associated to coarse-scale spatial bases, $\widetilde{\underline{u}} = \widetilde{\underline{u}}(\widetilde{\underline{a}}_\kappa)$, and $\widetilde{p} = \widetilde{p}(\widetilde{\underline{b}}_\kappa)$. To simplify the neural network problem, it is assumed that the contributions to the Jacobian matrix from the fine-scale terms are representable using the terms from the algebraic model (by Bazilevs et al. [14]).

The in-house solver constructs the global residual problem via an assembly routine. Therefore, each of the subproblems associated with the basis functions $\overline{\phi}_x$, $\overline{\phi}_y$, $\overline{\phi}_z$, $\overline{\Psi}$ are assembled with coarse-scale and fine-scale contributions at an element-local level, i.e. with the shape functions $\overline{\varphi}_x$, $\overline{\varphi}_y$, $\overline{\varphi}_z$, $\overline{\psi}$ respectively. The choice to predict the closure terms (the fine-scale contributions) at an element-local level is primarily attributed to implementation simplicity and parallelisability. Assembling the element-local closure terms to the global residual vector utilises the same logic as the element-local coarse-scale contribution. In comparison, predicting at the weighting-function level requires a separate routine that adds the weighting-level closure terms to the correct position in the residual vector.

3.2.2 Norm Selection and Stencil Choice

The development of a model that relies purely on local element features for closure term prediction is attractive from a practical standpoint (e.g. can be applied on nearly any mesh geometry). However, this fails to consider whether the closure terms are dependent on the rest of the domain or past history. Neglecting these dependencies prevents the ANN from considering the larger spatiotemporal structures that are evolving within the flow, reducing its ability to predict closure terms. This motivates the use of multi-element features, with the choice of elements used forming the stencil of the ANN.

A suitable stencil for coarse-scale features can be formed by taking inspiration from the fine-scale Greens' function of the linearised Navier-Stokes equations, based strictly on the definition of the fine scales through the fine-scale Greens' function. The fine-scale *field* is defined by the integral of the fine-scale Greens' function with the coarse-scale residual over integration space. Therefore, the fine scales at a *fixed* physical-space position are defined by the coarse-scale residuals around that physical position. This dependency can be seen by fixing the physical coordinates in the fine-scale Greens' function and varying over the integration space across the problem domain (see Fig. 3.2 for example). To this author's knowledge, there is currently no known literature example in which the fine-scale Greens' function for the linearised two-scale Navier-Stokes equations are given in plotted format. However there are known relevant examples that can be considered simplifications of the Navier-Stokes equations. Therefore, taking appropriate inspiration requires a simplification of the current problem statement. Consider solely the linearised momentum equations from Eq. 2.2b, derived by decomposing the velocity and pressure field into a steady (s) and non-steady component (ns), i.e. $\underline{u} = \underline{u}_s + \underline{u}_{ns}$ and $p = p_s + p_{ns}$:

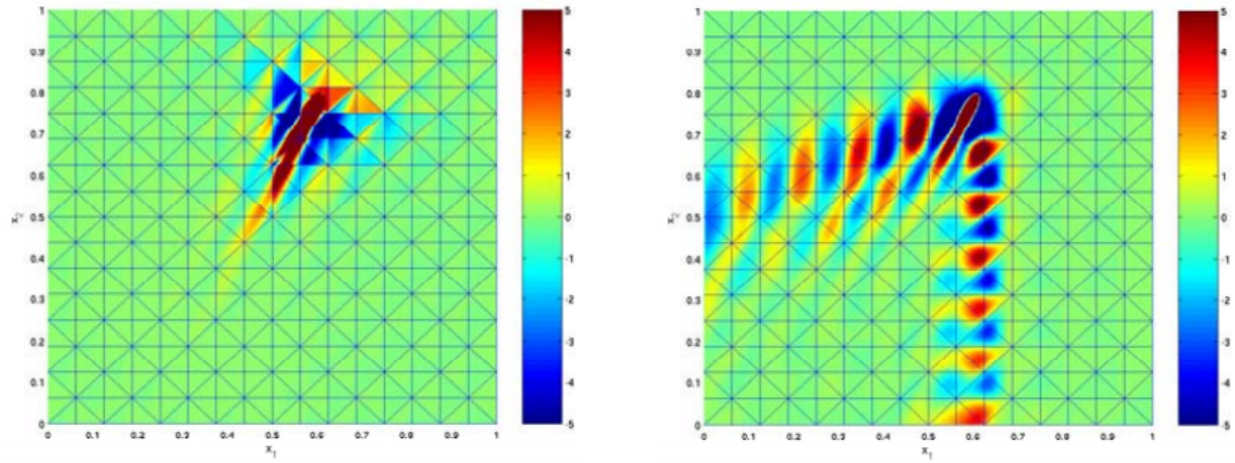
$$\frac{\partial\left(\underline{u}_s + \underline{u}_{ns}\right)}{\partial t} + \left(\left(\underline{u}_s + \underline{u}_{ns}\right)\cdot\underline{\nabla}\right)\left(\underline{u}_s + \underline{u}_{ns}\right) = -\frac{1}{\rho}\underline{\nabla}\left(p_s + p_{ns}\right) + \underline{\nabla}\cdot\left(2\nu\underline{\nabla}^s\left(\underline{u}_s + \underline{u}_{ns}\right)\right) + \underline{f}$$

Collecting all steady terms into a single forcing term $\widehat{\underline{f}}$ while neglecting the second order non-steady advection term yields the linearised momentum equations under the forcing function $(1/\rho)\,\underline{\nabla}p_{ns} + \widehat{\underline{f}}$, which read

$$\frac{\partial\underline{u}_{ns}}{\partial t} + \left(\underline{u}_s\cdot\underline{\nabla}\right)\underline{u}_{ns} + \left(\underline{u}_{ns}\cdot\underline{\nabla}\right)\underline{u}_s - \underline{\nabla}\cdot\left(2\nu\underline{\nabla}^s\underline{u}_{ns}\right) + = -\frac{1}{\rho}\underline{\nabla}p_{ns} + \widehat{\underline{f}} \tag{3.5}$$

It can be shown that Eq. 3.5 is similar to the advection-diffusion equation by Hughes and Sangalli [12]. Hughes and Sangalli derived the VMS formulation of the linear, steady 2D advection-diffusion problem with constant velocity, showcasing the closure terms as a function of the fine-scale Greens' function (see Eq. 3.7 in [12]). In relation to Eq. 3.5, the steady-state 2D problem with constant velocity can be seen as the equations that drive the non-steady component along a single plane defined by the wall-normal direction, where $\underline{u}_s$ is purely a function of channel height. Additionally, the steady z-velocity component w_s and non-steady z-velocity component w_{ns} have zero magnitude, which combined with $\underline{u}_s$ being a function of channel height removes the $\left(\underline{u}_{ns}\cdot\underline{\nabla}\right)\underline{u}_s$ term and all dependencies on the z-equation. The z-equation is also removed assuming the RHS of the $z-$equation is zero. The steady state problem provides inspiration as it produces an 'average' outlook into the

behaviour of the non-steady problem after a period of time where the initial conditions have settled. The fine-scale Greens' function was numerically calculated for different projectors, with the contributions of the functions to a fixed physical-space position shown in Fig. 3.2.



(A) H_0^1 projector [12, FIG. 3.9] **(B)** L_2 projector [12, FIG. 3.10]

FIGURE 3.2 Fine-scale Greens' function of the linear, steady, 2D advection-diffusion problem for linear triangular elements evaluated at the physical-space position $(x_0, y_0)^T \approx (0.6, 0.8)^T$. The unit advection velocity is given as $(0.5,\ 1.0)^T/\sqrt{1.25}$ and diffusivity coefficient is given as 10^{-3}.

Both figures demonstrate that the Greens' functions are not localised to a single element. Between Fig. 3.2a and Fig. 3.2b, the H_0^1 function is localised across a neighbourhood of elements, whereas the L_2 function is completely non-local. The lack of single-element locality implies that any point-wise evaluation of the fine-scale velocity is inferred by the coarse-scale residual from multiple elements, where the spatial influence is given by the fine-scale Greens' function in its chosen projection. In both cases, the fine-scale velocity (and therefore the closure terms) at the physical-space position is determined by upstream information, with the H_0^1 fine-scale being supported by the upwind characteristic passing through the position as pointed out by Hughes and Sangalli [12].

Due to the locality of the H_0^1 function, the ANN training data was chosen to be generated through an H_0^1 projection of DNS TCF data. To capture the upwind fine-scale support, four upstream elements of the (i, j, k)-element were used for the momentum spatial stencil alongside the (i, j, k)-element. The four elements are aligned in the x-direction, which, in relation to Fig. 3.2a would appear as a 'horizontal' support rather than a 'diagonal' support due to the choice in advection direction. The diagonal support that was shown in Fig. 3.2a is characteristic for the TCF test case in the situation where one of the two axes is the time-integration variable axis. It is expected that the TCF support appears narrower in the centre-channel due to the higher Péclet number, explaining the omission of spanwise elements in the stencil. Furthermore, this stencil is assumed to work closer to the channel walls where the Péclet number is lower in order to reduce the size of the feature

set. Since the velocity field may have a negative x-component, an element downstream of the (i, j, k)-element was also added into the momentum spatial stencil.

The continuity equation is treated as a Poisson problem as continuity is (implicitly) enforced through the pressure problem. A visual example of the H_0^1 Poisson fine-scale Greens' function fixed at a physical space position is given in Fig. 3.3 by courtesy of Shrestha [15]. Based on the figure, the Greens' function is localised by the direct neighbours of the (i, j, k)-element. Therefore, the continuity spatial stencil was initially chosen to include one element upstream and downstream of the (i, j, k)-element, alongside one element to both sides of the (i, j, k)-element (spanwise direction). To trivialise the implementation of the continuity stencil, the streamwise elements of the momentum stencil were also used. To include channel height dependency into the stencil, one element *away* from the closest wall was chosen to be part of the stencil (away relative to the (i, j, k)-element, i.e. the $(i, j, k \pm 1)$-element).

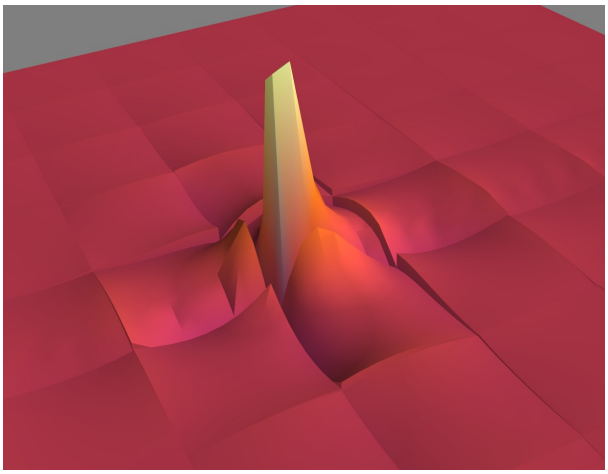


FIGURE 3.3 Poisson fine-scale Greens' function using the H_0^1 projector, (area-normalised) square elements with linear shape functions, and a diffusivity coefficient of 1.0. Courtesy of Shrestha and his framework developed for [15].

The decision to solely include one element away from the closest wall was made as no clear method to handle the wall is known. Kanala [8] utilised 'wall' elements to overcome the issue, where the wall elements are treated as standard elements but with features filled with zeros. This is a sensible choice theoretically as (element-integrated) coarse-scale features tend to a constant in the limit of the element size going to zero[1]. However, this is illogical from a practical standpoint because the test case mesh has no varying element size. The lack of transition between the size of the interior and wall elements poses a learning problem. This choice would be practical in the presence of a mesh with prismatic boundary layers.

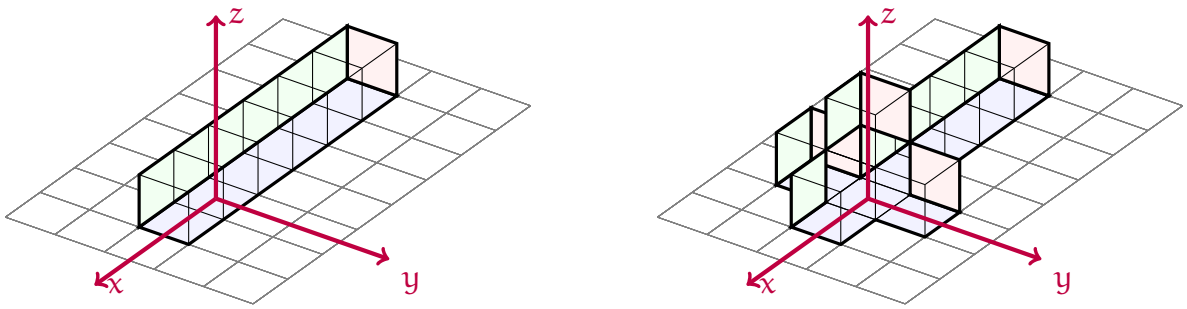
With regards to the temporal stencil, it can be expected that the Greens' functions at t^{n+1} are functions of time up to t^{n+1}. The causality principle prevents any temporal dependency

[1]Wall elements can be seen as elements with zero size: these elements have a defined surface size but no depth in the wall-normal direction. Any integrated quantities would tend to either zero (volume integrals) or a non-zero constant (surface integrals).

beyond t^{n+1}. It is assumed that the fine-scale Greens' functions behaves similarly, and it would motivate a stencil that encompasses several previous time steps and t^{n+1}. However, utilisation of elements at t^{n+1} implies new closure terms predictions are required during each of the solver's Newton iterations to provide the correct closure terms at t^{n+1}. This was studied by Janssens [3], finding that the weak residual topology was populated with spurious roots. Poor ANN predictions during a single iteration likely caused the Newton routine to converge to a spurious root, leading to solution divergence.

The spurious roots, also called the corrector pass instability, was handled by Rajampeta [7] by avoiding t^{n+1} elements in the temporal stencil. This allowed a single ANN prediction to be used for all iterations as no coarse-scale features change during each successive iteration. This temporal stencil is 'lagged' as the closure terms are predicted purely through the historical coarse scales. The feature set that involves this type of stencil is named a 'lagged feature set'. While the stencil provided by Rajampeta involved historical data from t^n up to t^{n-2}, a stencil up to t^{n-1} was used in this thesis to avoid an excessively large stencil.

A summarised visual of the spatial stencil is given in Fig. 3.4 for both t^n and t^{n-1}:



(A) Momentum stencil **(B)** Continuity stencil

FIGURE 3.4 Visualisation of the neural networks' spatial stencils used for the half-channel defined by the $z = 0$ wall. Coordinate system origins denotes the (i, j, k)-element, and its orientation denotes the direction in which the element index positively increments in.

3.2.3 Data Generation and Hardware

The projected TCF data used for model training was retrieved by an H_0^1 projection on a DNS TCF solution. The data was generated via a specialised in-house code for DNS TCFs, which uses finite volume methods in the wall-normal direction and spectral methods in the streamwise and spanwise direction. This code solves the primal problem on a $M_x \times M_y \times N_z = 80 \times 80 \times 128$ grid of size $x \times y \times z = 6.0 \times 4.0 \times 2.0$ in a segregated approach, where $M_{(\cdot)}$ denotes the number of modes in the $(\cdot)$-direction, and N_z are the number of cells in the z-direction. The flow simulation was carried out for a user-defined time period, and the H_0^1 projection of the flow onto a $32 \times 32 \times 32$ mesh retrieved the $\widetilde{\underline{u}}$ flowfield. The $\widetilde{\underline{u}}'$ flowfield was trivially retrieved by subtraction of the resolved flowfield from the DNS flowfield. The $\mathbf{B}_1$ and $\mathbf{B}_2$ terms were then calculated by integration. Any further post-processing was done by min-max scaling for training purposes, or normalisation scaling (per wall-normal height level) for correlational studies.

The time-step projection occurred every $\Delta t = 0.0072$, corresponding to a maximum Courant number between 0.95 and 1.00 ($\max(CFL_x) \approx 0.77$, $\max(CFL_y) \approx 0.18$, $\max(CFL_z) \approx 0.25$) and a mean Courant number of ~0.67 ($\text{mean}(CFL_x) \approx 0.60$, $\text{mean}(CFL_y) \approx 0.03$, $\text{mean}(CFL_z) \approx 0.05$). In terms of the DNS evolution, the time-step projection occurred every 60 DNS time steps, where every DNS time step was associated with an approximate maximum Courant number of 0.095. The DNS Courant number was chosen to minimise time-related discretisation errors. Four data sets of 120 time steps were generated using different projection starting times, starting from $t = 1.0080$, $t = 2.0088$, $t = 3.0096$, and $t = 4.0104$. The choice of using four different data sets was attributable to a memory leak in the post-processing of the data generation, and to the varying bulk velocity, see Fig. 3.5. The four data sets are labelled L_1, L_2, L_3, and L_4 respectively.

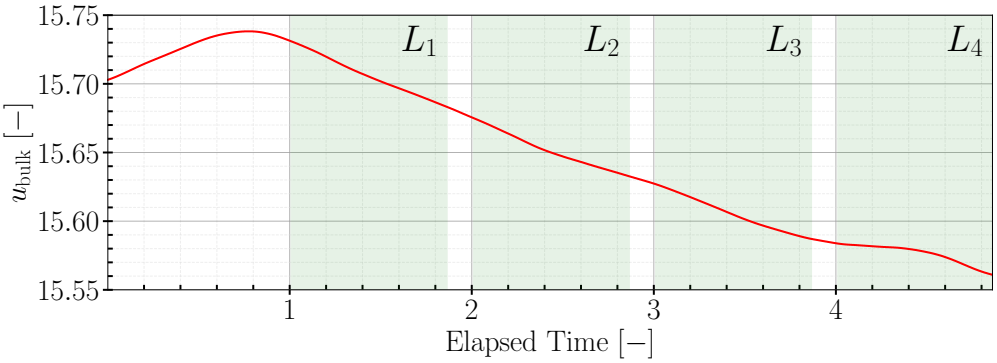


Figure 3.5 DNS TCF bulk velocity profile as a function of time. Green regions indicates time interval where data set projection occurs. The data sets are labeled using the annotations provided in the top-right corner of the green-shaded regions.

The model training was managed by Tensorflow, with all pre-processing completed before training. The generated data was split 25%–75% into test data and validation/training data respectively. The latter was further split 15%–85% into validation and training data respectively. Hardware-wise, the models were trained on a Dell T5810 with 32 GiB RAM and a GeForce GTX 970, and a Dell T5810 with 16 GiB RAM and a GeForce GTX 1070.

3.2.4 Features Correlations and Choices

Kanala [8] utilised primitive variables situated in the spatial centroids of the elements as part of the ANN feature set, which is problematic as post-processing is required to retrieve centroid features during each time step. Post-processing is not necessarily an expensive operation, but the existence of additional overhead is not desirable. An alternative, simpler, feature set is formulated by an observation of Eqs. (3.1–3.2). The two VMS equations describe how the coarse scales evolve as a function of the closure terms, but also describe the closure terms provided the coarse scales quantities. For a solver-compatible lagged feature set, Eqs. (3.3–3.4) at $\kappa = 0$ achieve the same desired effect. However, it notes that the closure terms are further offset by the residual vector.

Appendix A provides the histograms of all weak-form terms used in the in-house solver against the true t^{n+1} closure terms for all elements in the spatiotemporal stencil afore-

mentioned in Sec. 3.2.2. The histograms of the sum of all coarse-scale weak-forms are also provided, alongside histograms of various weak-form terms and coarse-scale sums at t^{n+1}. The histograms demonstrate that the momentum and continuity closure terms are correlated to the coarse-scale terms, but it varies between each term and position within the spatiotemporal stencil. A set of histograms are illustrated in Fig. A.1, which displays the correlational footprint between the t^{n+1} (i,j,k)-element x-momentum coarse-scale sum with the t^{n+1} (i,j,k)-element closure term per shape function, indexed using q. This figure reflects the above observation made regarding Eqs. (3.1–3.2). Figure A.1, and all other histograms in App. A, also illustrate the presence of two correlational supersets. One of these supersets is *weak* and the other superset is *strong*, where weak and strong are defined in regards to the Pearson correlation coefficient ρ. The weak superset is composed by the correlations stemming from shape functions $q = 1, 4, 5, 8$, while the strong superset is composed by the correlations stemming from shape functions $q = 2, 3, 6, 7$. Within these supersets there exist two sets, composed from different shape functions for all investigated terms, which provide similar skewness values, given in the title for the x-axis and y-axis data set respectively. Due to the varying correlations and skewness values, an initial choice to train one neural network per shape function was made. This prevents the networks from attempting to account for the differing statistics across shape functions. The eight trained networks act together as the fine-scale model.

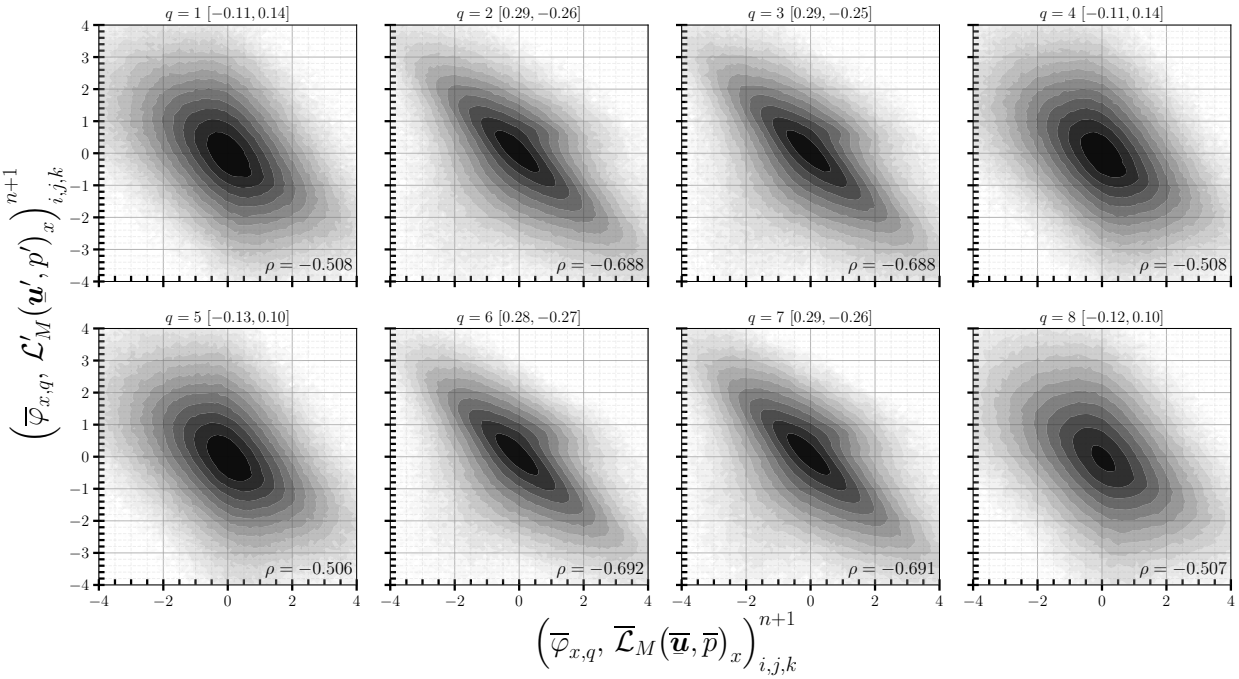


FIGURE A.1 Histograms of the t^{n+1} (i,j,k)-element x-momentum coarse-scale sum against the t^{n+1} (i,j,k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation. (repeated from page 63)

The observation in regards to Eqs. (3.3–3.4) is not reflected as strongly as the observation for Eqs. (3.1–3.2), as exemplified by the histograms of the t^n (i,j,k)-element when compared

to the t^{n+1} (i, j, k)-element, e.g. Fig. A.36 compared to Fig. A.1. The observation is better supported by the correlations of the upstream t^n $(i-1, j, k)$-element, shown in Fig. A.55. The histogram suggests that the advected upstream conditions better reflect the closure terms belonging to the t^{n+1} (i, j, k)-element. The loss of correlation between the time levels is attributed to diffusion and the slight deviations in advection direction from the x-axis.

The histograms of further upstream elements (relative to the (i, j, k)-element at t^n), such as Fig. A.74, Fig. A.93, and Fig. A.112, demonstrate that the correlational coefficients tend to zero when travelling in the upwind direction. These correlations become near-zero at $(i-4, j, k)$. This justifies the selection of four upstream elements to be part of the spatiotemporal stencil.

With regards to the temporal stencil, upstream information at t^{n-1} $(i-1, j, k)$ provides stronger correlations than t^n $(i-1, j, k)$, as displayed by Fig. A.226. It is speculated that this occurs due to the *mean* x-Courant number. With the mean x-Courant number being ~0.60, there is a split of elements requiring two time steps to convect the distance of one element.

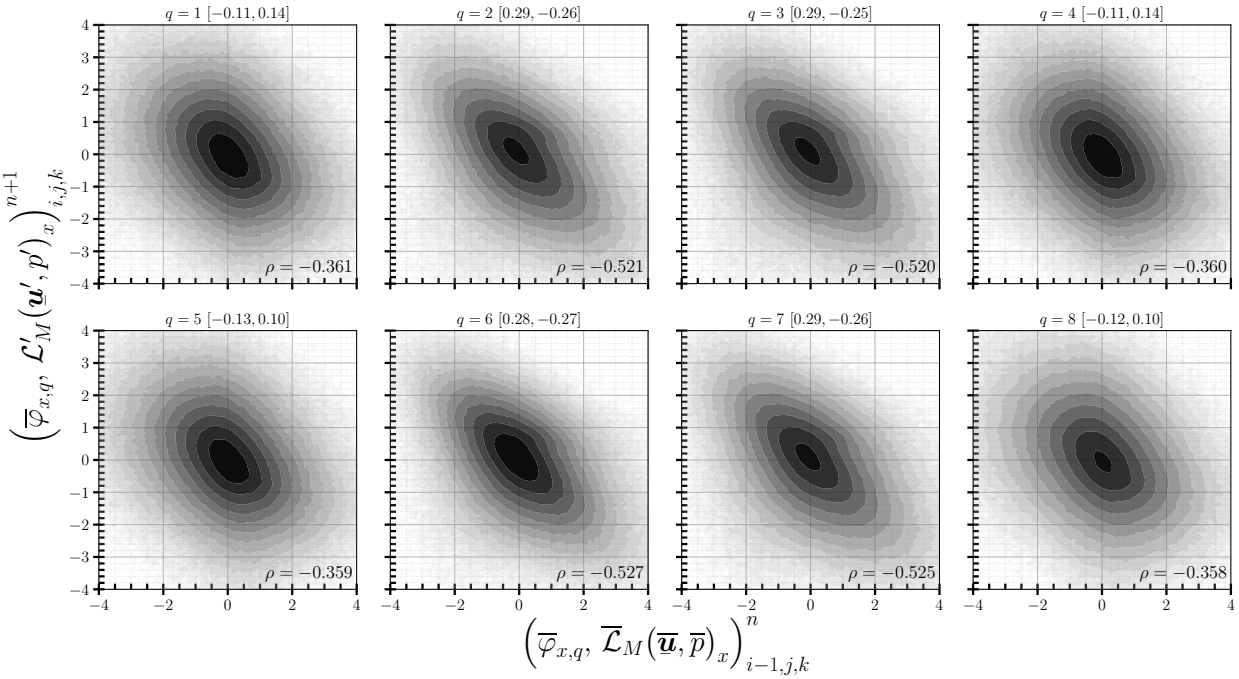


FIGURE A.55 Histograms of the t^n $(i-1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation. (repeated from page 93)

Selecting features from high-correlational histograms is a suitable method for the creation of a feature set, but it is difficult to conclude whether a feature is useless/redundant in the feature set from the low-correlational histograms. A seemingly useless feature may provide significant information to the neural network when paired with other features as stated by Guyon and Elisseef [16], and observed in Appendix A: the individual components of the coarse-scale sum are not as strongly correlated to the corresponding closure term when

compared to the sum itself. Therefore, this thesis included most terms in the feature set even if the correlations are close to zero.

The coarse-scale sum was included in the feature set for each element of the momentum spatiotemporal stencil. Additionally, all spatial weak-form terms of each element in the momentum spatiotemporal stencil were included, i.e. the advection term, pressure gradient term, and the diffusion term. The forcing term was not included as it did not vary in any element. The time derivative was also not included since the differences in the spatial features across the temporal stencil are indicative of temporal changes. The coarse-scale sum is the only term that includes the time march contribution. This is due to the fact that temporal effects are required to replicate the coarse-scale sums correctly. It is accepted that the neural networks depend on the spatiotemporal basis functions used to construct the velocity/pressure field. The velocity divergence contribution was provided for each element of the continuity spatiotemporal stencil. No further terms are included since the continuity equation lacks other terms. The outputs of all eight neural networks are the four closure terms of the two-scale Navier-Stokes equations.

To further augment the feature set, the wall-normal distance d of the (i, j, k)-element's centroid measured from a channel wall was provided. This was motivated by the successes of the neural network designs when using wall-normal distances from the work of Gamahara and Hattori [17]. Additionally, the x-, y-, z-velocity basis coefficients were provided for each spatiotemporal element, as they provide the nodal coarse-scale velocities and a measure of their gradients (for linear basis functions).

A summary of the outputs of the q-th shape function network is given in Tab. 3.1 and its features are given in Tab. 3.2. Table 3.1 provides additional output variables that were not discussed beforehand. These variables are related to the objective function, and serve to inform the network of solver-related quantities (Force-free weak-residual terms, see Sec. 3.3.2.2 and Sec. 3.3.2.3), to inject variables into the objective function within the TensorFlow framework (velocity basis coefficients, see Sec. 3.3.2.4), or to retrieve lost geometrical properties during a priori post-processing.

TABLE 3.1 Network outputs associated to the q-th shape function and (i,j,k)-th element. ✓ indicates that the quantity of interest is used in the solver.

	Output	Used in Solver?
Closure Terms	$\left(\overline{\varphi}_{x,q},\, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1}_{i,j,k}$	✓
	$\left(\overline{\varphi}_{y,q},\, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1}_{i,j,k}$	✓
	$\left(\overline{\varphi}_{z,q},\, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)^{n+1}_{i,j,k}$	✓
	$\left(\psi_q,\, \mathcal{L}_C(\widetilde{\underline{u}}')\right)^{n+1}_{i,j,k}$	✓
Force-Free Weak-Residual Terms	$\left(\overline{\varphi}_{x,q},\, \mathcal{R}_M(\widetilde{\widetilde{\underline{u}}}, \widetilde{\widetilde{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1}_{i,j,k}$	
	$\left(\overline{\varphi}_{y,q},\, \mathcal{R}_M(\widetilde{\widetilde{\underline{u}}}, \widetilde{\widetilde{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1}_{i,j,k}$	
	$\left(\overline{\varphi}_{z,q},\, \mathcal{R}_M(\widetilde{\widetilde{\underline{u}}}, \widetilde{\widetilde{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)^{n+1}_{i,j,k}$	
	$\left(\psi_q,\, \mathcal{R}_C(\widetilde{\widetilde{\underline{u}}}; \widetilde{\underline{u}}')\right)^{n+1}_{i,j,k}$	
Velocity Basis Coefficients	$\left[\widetilde{\widetilde{a}}_{x,q}\right]^{n+1}_{i,j,k}$	
	$\left[\widetilde{\widetilde{a}}_{y,q}\right]^{n+1}_{i,j,k}$	
	$\left[\widetilde{\widetilde{a}}_{z,q}\right]^{n+1}_{i,j,k}$	
Height Level	$\left[z\right]^{n}_{i,j,k}$	

TABLE 3.2 Feature set associated to the q-th shape function and (i, j, k)-th element. ✓ indicates inclusion in feature set and the subscript indicates position in feature set array.

Feature		Stencil																	
		(i,j,k)		$(i-1,j,k)$		$(i-2,j,k)$		$(i-3,j,k)$		$(i-4,j,k)$		$(i+1,j,k)$		$(i,j-1,k)$		$(i,j+1,k)$		$(i,j,k\pm1)$	
		t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}	t^n	t^{n-1}
Coarse-scale Sums	$\left(\overline{\varphi}_{x,q}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})_x\right)$	✓$_0$	✓$_{108}$	✓$_{16}$	✓$_{124}$	✓$_{32}$	✓$_{140}$	✓$_{48}$	✓$_{156}$	✓$_{64}$	✓$_{172}$	✓$_{80}$	✓$_{188}$						
	$\left(\overline{\varphi}_{y,q}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})_y\right)$	✓$_1$	✓$_{109}$	✓$_{17}$	✓$_{125}$	✓$_{33}$	✓$_{141}$	✓$_{49}$	✓$_{157}$	✓$_{65}$	✓$_{173}$	✓$_{81}$	✓$_{189}$						
	$\left(\overline{\varphi}_{z,q}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})_z\right)$	✓$_2$	✓$_{110}$	✓$_{18}$	✓$_{126}$	✓$_{34}$	✓$_{142}$	✓$_{50}$	✓$_{158}$	✓$_{66}$	✓$_{174}$	✓$_{82}$	✓$_{190}$						
Convective Terms	$\left(\overline{\varphi}_{x,q}, (\widetilde{\underline{u}} \cdot \underline{\nabla}\widetilde{\underline{u}})_x\right)$	✓$_3$	✓$_{111}$	✓$_{19}$	✓$_{127}$	✓$_{35}$	✓$_{143}$	✓$_{51}$	✓$_{159}$	✓$_{67}$	✓$_{175}$	✓$_{83}$	✓$_{191}$						
	$\left(\overline{\varphi}_{y,q}, (\widetilde{\underline{u}} \cdot \underline{\nabla}\widetilde{\underline{u}})_y\right)$	✓$_4$	✓$_{112}$	✓$_{20}$	✓$_{128}$	✓$_{36}$	✓$_{144}$	✓$_{52}$	✓$_{160}$	✓$_{68}$	✓$_{176}$	✓$_{84}$	✓$_{192}$						
	$\left(\overline{\varphi}_{z,q}, (\widetilde{\underline{u}} \cdot \underline{\nabla}\widetilde{\underline{u}})_z\right)$	✓$_5$	✓$_{113}$	✓$_{21}$	✓$_{129}$	✓$_{37}$	✓$_{145}$	✓$_{53}$	✓$_{161}$	✓$_{69}$	✓$_{177}$	✓$_{85}$	✓$_{193}$						
Pressure Terms	$\left(\overline{\varphi}_{x,q}, (\underline{\nabla}\widetilde{p})_x\right)$	✓$_6$	✓$_{114}$	✓$_{22}$	✓$_{130}$	✓$_{38}$	✓$_{146}$	✓$_{54}$	✓$_{162}$	✓$_{70}$	✓$_{178}$	✓$_{86}$	✓$_{194}$						
	$\left(\overline{\varphi}_{y,q}, (\underline{\nabla}\widetilde{p})_y\right)$	✓$_7$	✓$_{115}$	✓$_{23}$	✓$_{131}$	✓$_{39}$	✓$_{147}$	✓$_{55}$	✓$_{163}$	✓$_{71}$	✓$_{179}$	✓$_{87}$	✓$_{195}$						
	$\left(\overline{\varphi}_{z,q}, (\underline{\nabla}\widetilde{p})_z\right)$	✓$_8$	✓$_{116}$	✓$_{24}$	✓$_{132}$	✓$_{40}$	✓$_{148}$	✓$_{56}$	✓$_{164}$	✓$_{72}$	✓$_{180}$	✓$_{88}$	✓$_{196}$						
Diffusion Terms	$\left(\underline{\nabla}\overline{\varphi}_{x,q}, (\underline{\nabla}^s\widetilde{\underline{u}})_x\right)$	✓$_9$	✓$_{117}$	✓$_{25}$	✓$_{133}$	✓$_{41}$	✓$_{149}$	✓$_{57}$	✓$_{165}$	✓$_{73}$	✓$_{181}$	✓$_{89}$	✓$_{197}$						
	$\left(\underline{\nabla}\overline{\varphi}_{y,q}, (\underline{\nabla}^s\widetilde{\underline{u}})_y\right)$	✓$_{10}$	✓$_{118}$	✓$_{26}$	✓$_{134}$	✓$_{42}$	✓$_{150}$	✓$_{58}$	✓$_{166}$	✓$_{74}$	✓$_{182}$	✓$_{90}$	✓$_{198}$						
	$\left(\underline{\nabla}\overline{\varphi}_{z,q}, (\underline{\nabla}^s\widetilde{\underline{u}})_z\right)$	✓$_{11}$	✓$_{119}$	✓$_{27}$	✓$_{135}$	✓$_{43}$	✓$_{151}$	✓$_{59}$	✓$_{167}$	✓$_{75}$	✓$_{183}$	✓$_{91}$	✓$_{199}$						
Divergence Term	$-\left(\underline{\nabla}\psi_q, \widetilde{\underline{u}}\right)$	✓$_{12}$	✓$_{120}$	✓$_{28}$	✓$_{136}$	✓$_{44}$	✓$_{152}$	✓$_{60}$	✓$_{168}$	✓$_{76}$	✓$_{184}$	✓$_{92}$	✓$_{200}$	✓$_{96}$	✓$_{204}$	✓$_{100}$	✓$_{208}$	✓$_{104}$	✓$_{212}$
Velocity Basis Coefficients	$\widetilde{\underline{a}}_{x,q}$	✓$_{13}$	✓$_{121}$	✓$_{29}$	✓$_{137}$	✓$_{45}$	✓$_{153}$	✓$_{61}$	✓$_{169}$	✓$_{77}$	✓$_{185}$	✓$_{93}$	✓$_{201}$	✓$_{97}$	✓$_{205}$	✓$_{101}$	✓$_{209}$	✓$_{105}$	✓$_{213}$
	$\widetilde{\underline{a}}_{y,q}$	✓$_{14}$	✓$_{122}$	✓$_{30}$	✓$_{138}$	✓$_{46}$	✓$_{154}$	✓$_{62}$	✓$_{170}$	✓$_{78}$	✓$_{186}$	✓$_{94}$	✓$_{202}$	✓$_{98}$	✓$_{206}$	✓$_{102}$	✓$_{210}$	✓$_{106}$	✓$_{214}$
	$\widetilde{\underline{a}}_{z,q}$	✓$_{15}$	✓$_{123}$	✓$_{31}$	✓$_{139}$	✓$_{47}$	✓$_{155}$	✓$_{63}$	✓$_{171}$	✓$_{79}$	✓$_{187}$	✓$_{95}$	✓$_{203}$	✓$_{99}$	✓$_{207}$	✓$_{103}$	✓$_{211}$	✓$_{107}$	✓$_{215}$
Wall-normal Distance	d	✓$_{216}$																	

3.2.5 Data Augmentation

The data augmentation utilised is related to the wall-normal distance d. As aforementioned in Sec. 3.2.4, the wall-normal distance measures the distance from *a* channel wall, but the distance can be measured from the opposing channel wall. There is no distinction between the two walls other than the semantics associated to 'lower channel wall' and 'upper channel wall' in this test case. Therefore, no distinction is present between two feature sets in which the wall-normal distance is measured from a separate wall. To enforce this generalisation, the data was augmented with a copy of itself, but the copy's wall-normal distance was measured from the opposing wall.

Rotational data augmentation was considered for this thesis, as it provides additional flow cases that the model can be trained on. This is at the expense of a slight increase in cost associated with data generation (data is retrieved by rotational transformations of a single case). This was deliberately ignored due to hardware considerations for model training.

3.3 ANN Architecture

The choice of network ultimately determines the regressional relationship between the coarse scales and fine scales. This relationship is established by minimising an objective function, which can be altered such that it displays desired behaviour. This section covers the network type used in this thesis, its objective function, and network hyperparameters.

3.3.1 ANN Type

Due to the lack of state-of-the-art research on VMS-ANN modelling of the Navier-Stokes equations, it is difficult to justify the use of one type of architecture. It is to this author's opinion that the lack of VMS-ANN modelling suggests that starting with an MLP design is beneficial for two reasons. Firstly, this thesis can act as a foundational work for future comparisons due to the simplistic nature of an MLP architecture. Secondly, an MLP is simple to implement into the in-house solver. From an academic standpoint, the use of MLP designs also shows promise from the work conducted in LES, e.g. [9, 17]. Gamahara and Hattori [17] achieved correlation coefficients between prediction and truth values of 0.50 to 0.70 using a single hidden layer and a hundred neurons for the divergence of the subgrid-scale (SGS) stress tensor of a $\mathrm{Re}_\tau = 180$ top-hat filtered TCF. In comparison, Zhou et al. [9] utilised a single hidden layer with 300 neurons, and achieved correlation coefficients between prediction and truth above 0.90 for the SGS stress tensor of a Gaussian-filtered homogeneous isotropic turbulence (HIT) flow.

Successful architectures utilising convolutional layers and recurrent layers are not negligible from consideration. The related works [18, 19] on perfect LES neural networks are excellent examples of non-MLP architectures for ANN modelling, with both works focusing on modelling decaying homogeneous isotropic turbulence (DHIT) flows at a Taylor microscale Reynolds number $\mathrm{Re}_\lambda = 180$. Beck, Flad, and Munz [18] utilised residual neural networks, where each block was composed of convolutional layers, to predict the exact closure term of each element in the LES grid. Their deepest network of eight residual blocks gave

correlation coefficients between prediction and truth of ~0.70 for the closure term of a DHIT defined by the DNS-to-LES operator in Sec. 2.3, showcasing the success of convolutional networks. The convolutional network worked on interpolated data within each mesh element, hence the usability of convolutional layers. Kurz and Beck [19] instead utilised gated recurrent unit (GRU) layers in their recurrent neural network to predict the exact closure terms of each element in the LES grid of a DHIT flow defined by various filters. In comparison to Beck, Flad, and Munz [18], correlation coefficients between prediction and truth of 0.99 were achieved, showcasing the potential of utilising recurrent neural networks for direct closure term modelling. From an alternative perspective, recurrent neural networks are also excellent architecture candidates given the (auto)correlations provided in examples such as Figs. (A.55–A.57) and Figs. (A.226–A.228).

Both convolutional and recurrent neural networks are suitable architectures for this thesis due to the utilisation of spatial stencils and the temporal dependency of the closure terms on various features discussed in Sec. 3.2.4. However, realistic time constraints and the lack of foundational work in the VMS-ANN field promote the use of simple architectures. Therefore, for the purpose of this thesis, an MLP architecture was used. To simplify construction, each hidden layer in the neural networks shared the same number of neurons as each other hidden layer.

3.3.2 Objective Function

The network objective function establishes the regressional relationship between the feature set and the closure terms. The objective function used in this thesis is composed of several error and penalty functions, with both function types measuring network predictions against projected DNS data. This subsection discusses each of these error and penalty functions individually, before providing a network problem statement.

3.3.2.1 Closure Term Error

The purpose of each neural network is to predict the closure terms for the coarse-scale equations, where each network handles one shape function. Since the weak residual form of these equations are solved around t^{n+1}, and that the global residual vector is assembled element-by-element, the requirement for the q-th shape function $\overline{\varphi}_{x,q}$, $\overline{\varphi}_{y,q}$, $\overline{\varphi}_{z,q}$, $\overline{\psi}_q$ of each element can be stated as[2]:

$$\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

[2]Note that the square brackets only serve to clarify the stacked number of brackets.

This equality requirement can be restated as four separate error functions for the ANN regressor by rearrangement into the form LHS $-$ RHS $=$ 0 from LHS $=$ RHS. The mean-square error was chosen as the metric, assuming a batch size N_e:

$$\left[\varepsilon^{M}_{x,q}\right]^{MSE}_{CT} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{prediction} - \left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{truth}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^{M}_{y,q}\right]^{MSE}_{CT} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{prediction} - \left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{truth}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^{M}_{z,q}\right]^{MSE}_{CT} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{prediction} - \left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{truth}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^{C}_{q}\right]^{MSE}_{CT} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}_{prediction} - \left[\left(\overline{\psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}_{truth}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

The mean-square error was chosen over the mean-absolute error to weigh larger errors more as larger localised errors take longer to diffuse/dissipate.

3.3.2.2 Force-Free Weak-Residual Error

A neural network that only regresses for the closure terms neglects the physics used within the solver. A solution is to train the neural networks to also predict the weak residual, thereby learning the relationship between the coarse scale equations and closure terms used in the solver. The weak residual prediction is known to be an offset of the closure term prediction, where this offset can be determined by manipulation of the equations used in the solver itself. For each of the residual equations Eqs. (3.3–3.4), this aforementioned shift is the coarse-scale sum, in addition to the forcing term $\underline{f}$ for the momentum equations. As the coarse-scale sum is part of the feature set, the neural networks should explicitly learn a relationship between the closure term and the coarse-scale sum feature. However the forcing term is not a feature as it is constant amongst all quantities. Therefore, it is possible that the neural networks erroneously attribute the forcing term offset in the weak residual to other features rather than a bias. To minimise this possibility, a 'force-free' weak residual is predicted, defined by rewriting the residual equations Eqs. (3.3–3.4) into the form

$$\left(\overline{\underline{\Phi}}, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})\right)^n + \left(\overline{\underline{\Phi}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\right)^{n+1} = \left(\overline{\underline{\Phi}}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')\right)^{n+1}$$

$$\left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}})\right)^n + \left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}}')\right)^{n+1} = \left(\overline{\Psi}, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)^{n+1}$$

where

$$\left(\overline{\underline{\Phi}}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')\right)^{n+1} = \left(\overline{\underline{\Phi}}, \mathsf{Res}^M_w(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')\right)^{n+1} + \left(\overline{\underline{\Phi}}, \underline{f}\right)^{n+1}$$

$$\left(\overline{\Psi}, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)^{n+1} = \left(\overline{\Psi}, \mathsf{Res}^C_w(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)^{n+1}$$

The $\mathcal{R}$ terms are so-called force-free since the new equations that describe $\mathcal{R}$ do not contain the forcing term $\underline{f}$ on the LHS. Expansion of the force-free residual equations yield

$$\left(\overline{\phi}_x, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{p})_x\right)^n + \left(\overline{\phi}_x, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1} = \left(\overline{\phi}_x, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{p}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1} \tag{3.6a}$$

$$\left(\overline{\phi}_y, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}})_y\right)^n + \left(\overline{\phi}_y, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1} = \left(\overline{\phi}_y, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1} \tag{3.6b}$$

$$\left(\overline{\phi}_z, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}})_z\right)^n + \left(\overline{\phi}_z, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)^{n+1} = \left(\overline{\phi}_z, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)^{n+1} \tag{3.6c}$$

$$\left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}})\right)^n + \left(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}}')\right)^{n+1} = \left(\overline{\Psi}, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)^{n+1} \tag{3.6d}$$

and since the global residual vector is assembled element-by-element, the requirement for each shape function can be stated as:

$$\left[\left(\overline{\varphi}_{x,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{x,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\varphi}_{y,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{y,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\varphi}_{z,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\varphi}_{z,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\left(\overline{\Psi}_q, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{prediction}} = \left[\left(\overline{\Psi}_q, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

This equality requirement can also be restated as four separate error functions for the ANN regressor by rearrangement into the form LHS − RHS = 0 from LHS = RHS. The mean-square error was chosen as the error function, assuming a batch size N_e:

$$\left[\varepsilon^M_{x,q}\right]^{\text{MSE}}_{\text{FFWR}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{x,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{prediction}} - \left[\left(\overline{\varphi}_{x,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}_{\text{truth}}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^M_{y,q}\right]^{\text{MSE}}_{\text{FFWR}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{y,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{prediction}} - \left[\left(\overline{\varphi}_{y,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}_{\text{truth}}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^M_{z,q}\right]^{\text{MSE}}_{\text{FFWR}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\varphi}_{z,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{prediction}} - \left[\left(\overline{\varphi}_{z,q}, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}_{\text{truth}}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\left[\varepsilon^C_q\right]^{\text{MSE}}_{\text{FFWR}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\left[\left(\overline{\Psi}_q, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{prediction}} - \left[\left(\overline{\Psi}_q, \mathcal{R}_C(\widetilde{\underline{u}}; \widetilde{\underline{u}}')\right)\right]^{n+1}_{\text{truth}}\right]^2_i \quad \forall q = 1, 2, \cdots, 8$$

where the choice of mean-square error was chosen for similar reasons as stated in Sec. 3.3.2.1.

3.3.2.3 Solver Adherence Penalty

Two of the three terms in each equation of Sys. 3.6 are predicted by the ANN and therefore there is an implicit prediction of the coarse-scale sum. This is underlined below:

$$\left(\overline{\phi}_x, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}})_x\right)^n + \underbrace{\left(\overline{\phi}_x, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1}}_{\text{predicted}} = \underbrace{\left(\overline{\phi}_x, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_x\right)^{n+1}}_{\text{predicted}}$$

$$\left(\overline{\phi}_y, \overline{\mathcal{L}}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}})_y\right)^n + \underbrace{\left(\overline{\phi}_y, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1}}_{\text{predicted}} = \underbrace{\left(\overline{\phi}_y, \mathcal{R}_M(\widetilde{\underline{u}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_y\right)^{n+1}}_{\text{predicted}}$$

$$\Big(\overline{\phi}_z, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_z\Big)^n + \underbrace{\Big(\overline{\phi}_z, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\Big)^{n+1}}_{\text{predicted}} = \underbrace{\Big(\overline{\phi}_z, \mathcal{R}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}}; \widetilde{\underline{u}}', \widetilde{p}')_z\Big)^{n+1}}_{\text{predicted}}$$

$$\Big(\overline{\Psi}, \mathcal{L}_C(\widetilde{\overline{\underline{u}}})\Big)^n + \underbrace{\Big(\overline{\Psi}, \mathcal{L}_C(\widetilde{\underline{u}}')\Big)^{n+1}}_{\text{predicted}} = \underbrace{\Big(\overline{\Psi}, \mathcal{R}_C(\widetilde{\overline{\underline{u}}}; \widetilde{\underline{u}}')\Big)^{n+1}}_{\text{predicted}}$$

The implicit prediction of the coarse-scale sum is required to be exact since the predicted quantity exists as a feature of the ANN. In relation to the discussion of Sec. 3.3.2.2, this requirement forces the ANN to learn that the predicted force-free weak-residual contribution is an offset of the closure terms. The violation of this equality requirement was measured via the mean-square error between prediction and truth, assuming a batch size N_e:

$$\Big[\varepsilon^M_{x,q}\Big]^{MSE}_{SA} = \frac{1}{N_e}\sum_{i=1}^{N_e}\Bigg[\Big[\Big(\overline{\varphi}_{x,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_x\Big)\Big]^n_{\text{prediction}} - \Big[\Big(\overline{\varphi}_{x,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_x\Big)\Big]^n_{\text{truth}}\Bigg]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\Big[\varepsilon^M_{y,q}\Big]^{MSE}_{SA} = \frac{1}{N_e}\sum_{i=1}^{N_e}\Bigg[\Big[\Big(\overline{\varphi}_{y,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_y\Big)\Big]^n_{\text{prediction}} - \Big[\Big(\overline{\varphi}_{y,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_y\Big)\Big]^n_{\text{truth}}\Bigg]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\Big[\varepsilon^M_{z,q}\Big]^{MSE}_{SA} = \frac{1}{N_e}\sum_{i=1}^{N_e}\Bigg[\Big[\Big(\overline{\varphi}_{z,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_z\Big)\Big]^n_{\text{prediction}} - \Big[\Big(\overline{\varphi}_{z,q}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})_z\Big)\Big]^n_{\text{truth}}\Bigg]^2_i \quad \forall q = 1, 2, \cdots, 8$$

$$\Big[\varepsilon^C_q\Big]^{MSE}_{SA} = \frac{1}{N_e}\sum_{i=1}^{N_e}\Bigg[\Big[\Big(\overline{\Psi}_q, \mathcal{L}_C(\widetilde{\overline{\underline{u}}})\Big)\Big]^{n+1}_{\text{prediction}} - \Big[\Big(\overline{\Psi}_q, \mathcal{L}_C(\widetilde{\overline{\underline{u}}})\Big)\Big]^{n+1}_{\text{truth}}\Bigg]^2_i \quad \forall q = 1, 2, \cdots, 8$$

The mean-square error was chosen as it provides a function that will be of differentiability class C^1 when implemented as part of the final unconstrained problem, see Sec. 3.3.2.5.

3.3.2.4 Energy Transfer Penalty (Conservation of Energy)

When the momentum closure term predictions deviate from the truth, the net kinetic energy of the flow is impacted. As a consequence, long-term instabilities may manifest. One type of long-term instability is energy accumulation, which will eventually lead to energy divergence as witnessed in a posteriori simulations by Rajampeta [7]. Rajampeta addressed this long-term instability by clipping the energy transfer in the presence of backscatter. In the three dimensional problem, the clipping procedure is achieved by doing a logical check on the fine-scale energy transfer in the coarse-scale kinetic energy equation

$$\Big(\widetilde{\overline{\underline{u}}}, \overline{\mathcal{L}}_M(\widetilde{\overline{\underline{u}}}, \widetilde{\overline{p}})\Big) - \Big(\widetilde{\overline{\underline{u}}}, \underline{f}\Big) = -\Big(\widetilde{\overline{\underline{u}}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)$$

where if

$$\Big[-\Big(\widetilde{\overline{\underline{u}}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)\Big]^{n+1}_{\text{prediction}} \geqslant 0$$

then the closure terms are set to zero. This backscatter limiting technique is drastic as all forms of fine-scale backscatter are omitted, leading to an excessively dissipative solution. It is possible to relax this backscatter restriction by considering that an energy-bound flow is

ensured if the energy transfer predicted by the neural network's closure term is smaller than the truth value, i.e.

$$\Big[-\Big(\widetilde{\underline{\overline{u}}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\Big(\widetilde{\underline{\overline{u}}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)\Big]^{n+1}_{\text{truth}}$$

This would then be implemented as an inequality associated with the neural network's objective function. As no precedent work on energy bounding has been carried out in this framework, this thesis studies a more restrictive version of this relaxed inequality, where each component of the inequality is restricted rather than the net sum, i.e.

$$\Big[-\Big(\widetilde{\overline{u}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_x\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\Big(\widetilde{\overline{u}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_x\Big]^{n+1}_{\text{truth}}$$

$$\Big[-\Big(\widetilde{\overline{v}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_y\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\Big(\widetilde{\overline{v}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_y\Big]^{n+1}_{\text{truth}}$$

$$\Big[-\Big(\widetilde{\overline{w}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_z\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\Big(\widetilde{\overline{w}}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')\Big)_z\Big]^{n+1}_{\text{truth}}$$

where $\widetilde{\overline{u}}$, $\widetilde{\overline{v}}$, and $\widetilde{\overline{w}}$ are the x-, y-, and z-component of $\widetilde{\underline{\overline{u}}}$ respectively. By noting that the velocity field is composed by basis functions, these inequalities are broken down into

$$\Big[-\widetilde{\underline{\overline{a}}}_x^T\Big(\underline{\overline{\Phi}}_x, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\underline{\overline{a}}}_x^T\Big(\underline{\overline{\Phi}}_x, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\Big)\Big]^{n+1}_{\text{truth}}$$

$$\Big[-\widetilde{\underline{\overline{a}}}_y^T\Big(\underline{\overline{\Phi}}_y, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\underline{\overline{a}}}_y^T\Big(\underline{\overline{\Phi}}_y, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\Big)\Big]^{n+1}_{\text{truth}}$$

$$\Big[-\widetilde{\underline{\overline{a}}}_z^T\Big(\underline{\overline{\Phi}}_z, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\underline{\overline{a}}}_z^T\Big(\underline{\overline{\Phi}}_z, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\Big)\Big]^{n+1}_{\text{truth}}$$

where $\underline{\overline{\Phi}}_{(\cdot)} = \big(\overline{\Phi}_{(\cdot),1}, \overline{\Phi}_{(\cdot),2}, \cdots, \overline{\Phi}_{(\cdot),N_b}\big)^T$, N_b is the total number of basis functions associated to the $(\cdot)$-velocity component, and $\widetilde{\underline{\overline{a}}}_{(\cdot)}$ are the $(\cdot)$-velocity basis coefficients. These global-level inequalities are difficult to implement as part of the objective function when considering that all training data is randomly shuffled. However, the implementation of these inequalities can be avoided as long as the energy transfer predictions obey the same inequality requirement at the element-level. The assembly of the element predictions (and then the summation of these weighting-level predictions) satisfy the global-level inequality by the additional property of inequalities. Therefore the following inequalities must be satisfied at an element-level:

$$\Big[-\widetilde{\overline{a}}_{x,q}\Big(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\overline{a}}_{x,q}\Big(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\Big)\Big]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\Big[-\widetilde{\overline{a}}_{y,q}\Big(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\overline{a}}_{y,q}\Big(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\Big)\Big]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

$$\Big[-\widetilde{\overline{a}}_{z,q}\Big(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\Big)\Big]^{n+1}_{\text{prediction}} \leqslant \Big[-\widetilde{\overline{a}}_{z,q}\Big(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\Big)\Big]^{n+1}_{\text{truth}} \quad \forall q = 1, 2, \cdots, 8$$

where $\widetilde{\overline{a}}_{(\cdot),q}$ is the $(\cdot)$-velocity basis function coefficient associated to the element's q-th shape function. Satisfying this inequality either yields the exact energy transfer or excess dissipation, whereas the violation of this inequality can lead to instability formations due to excess energy generation at the global level. This violation manifests in one of the following two manners at the element level: excess backscatter or insufficient dissipation. It should be noted that insufficient dissipation is preferable to excess backscatter as energy is still being taken away from the system.

Since the basis coefficients are unknown at the next time level and are also not predicted by the ANN, the element-level inequalities can be relaxed into the following form:

$$\left[-\widetilde{\overline{a}}_{x,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{x,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_x\right)\right]^{n+1}_{\text{prediction}} \leqslant \left[-\widetilde{\overline{a}}_{x,q}\left(\overline{\varphi}_{x,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_x\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

$$\left[-\widetilde{\overline{a}}_{y,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{y,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_y\right)\right]^{n+1}_{\text{prediction}} \leqslant \left[-\widetilde{\overline{a}}_{y,q}\left(\overline{\varphi}_{y,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_y\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

$$\left[-\widetilde{\overline{a}}_{z,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{z,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_z\right)\right]^{n+1}_{\text{prediction}} \leqslant \left[-\widetilde{\overline{a}}_{z,q}\left(\overline{\varphi}_{z,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_z\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

where a simplification by omission of the basis coefficients yields a simpler inequality. This simplification is not carried out as the basis coefficients are necessary for providing sample weights to the error function associated to the inequalities. While the integrated terms in the inequality already weigh the error functions, the principle idea is to further weigh the energy contributions to minimise overpredictions. If the predicted energy contribution is small from one element due to a small basis coefficient, then it is acceptable that excess backscatter / insufficient dissipation exists. The net contribution is far smaller than excess backscatter / insufficient dissipation stemming from an element with a large basis coefficient.

The measure of error was chosen to be the mean-square error between prediction and truth only when the inequality is violated:

$$\left[\varepsilon^M_{x,q}\right]^{\text{MSE}}_{\text{ETP}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\max\left(0,E^{x,q}_{\text{diff}}\right)\right]^2_i \quad \forall q = 1,2,\cdots,8$$

$$\left[\varepsilon^M_{y,q}\right]^{\text{MSE}}_{\text{ETP}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\max\left(0,E^{y,q}_{\text{diff}}\right)\right]^2_i \quad \forall q = 1,2,\cdots,8$$

$$\left[\varepsilon^M_{z,q}\right]^{\text{MSE}}_{\text{ETP}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\left[\max\left(0,E^{z,q}_{\text{diff}}\right)\right]^2_i \quad \forall q = 1,2,\cdots,8$$

where

$$E^{x,q}_{\text{diff}} = \left[-\widetilde{\overline{a}}_{x,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{x,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_x\right)\right]^{n+1}_{\text{prediction}} - \left[-\widetilde{\overline{a}}_{x,q}\left(\overline{\varphi}_{x,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_x\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

$$E^{y,q}_{\text{diff}} = \left[-\widetilde{\overline{a}}_{y,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{y,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_y\right)\right]^{n+1}_{\text{prediction}} - \left[-\widetilde{\overline{a}}_{y,q}\left(\overline{\varphi}_{y,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_y\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

$$E^{z,q}_{\text{diff}} = \left[-\widetilde{\overline{a}}_{z,q}\right]^{n+1}_{\text{truth}}\left[\left(\overline{\varphi}_{z,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_z\right)\right]^{n+1}_{\text{prediction}} - \left[-\widetilde{\overline{a}}_{z,q}\left(\overline{\varphi}_{z,q},\mathcal{L}'_M(\widetilde{\underline{u}}',\widetilde{p}')_z\right)\right]^{n+1}_{\text{truth}} \quad \forall q = 1,2,\cdots,8$$

When the predicted energy transfer is larger than the truth for a single example, the error estimate contribution is non-zero. However, when the truth is larger than the prediction, the error estimate is zero. The zero-error estimate is acceptable as the inequality is held true. The mean-square error was chosen as it provides a function that will be of differentiability class C^1 when implemented as part of the final unconstrained problem, see Sec. 3.3.2.5.

3.3.2.5 Network Problem Statement

The error functions from the previous sections are combined to form the following constrained problem for the q-th shape function:

$$\begin{aligned}\min \quad & \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{CT} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{CT} \\ & + \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{FFWR}\end{aligned}$$

$$\begin{aligned}\text{subject to} \quad & 0 = \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{SA} = \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{SA} = \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{SA} = \left[\varepsilon^{C}_{q}\right]^{MSE}_{SA} \\ & 0 = \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{ETP} = \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{ETP} = \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{ETP}\end{aligned}$$

which is accordingly transformed into an unconstrained problem. This can be achieved via method of multipliers, but represents an unknown implementation challenge into TensorFlow as TensorFlow is not designed for constrained problems. Instead, the problem is transformed by use of Augmented Lagrangian methods, which provide estimates of the Lagrange multiplier while having the favourable property of ease-of-implementation. Each constraint ε is transformed into the k-th iterate penalty ρ_k as detailed by Nocedal and Wright [20]:

$$\rho_k = \frac{\mu^k}{2}\varepsilon^2 - \lambda^k\varepsilon$$

where μ^k and λ^k are the k-th iterate penalty coefficient and k-th iterate Lagrange multiplier estimate respectively. Substitution of the constraints with the penalties yields the k-th iterate unconstrained problem, which reads

$$\begin{aligned}\min \quad & \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{CT} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{CT} \\ & + \left[\varepsilon^{M}_{x,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{FFWR} \\ & + \left[\rho^{M}_{x,q,k}\right]^{MSE}_{SA} + \left[\rho^{M}_{y,q,k}\right]^{MSE}_{SA} + \left[\rho^{M}_{z,q,k}\right]^{MSE}_{SA} + \left[\rho^{C}_{q,k}\right]^{MSE}_{SA} \\ & + \left[\rho^{M}_{x,q,k}\right]^{MSE}_{ETP} + \left[\rho^{M}_{y,q,k}\right]^{MSE}_{ETP} + \left[\rho^{M}_{z,q,k}\right]^{MSE}_{ETP}\end{aligned}$$

and its solution (the arguments of the unconstrained problem) is followed by updates for μ^k and λ^k for each penalty:

$$\mu^{k+1} = \beta\mu^k$$

$$\lambda^{k+1} = \lambda^k - \mu^k\varepsilon$$

where ε is evaluated using the new arguments of the unconstrained problem, and β is a growth constant. The arguments for the unconstrained problem are the weights and biases. This uses the training data as implied parameters. However, the quality of the regression is judged by the provided validation data, therefore it is chosen to update λ^k using the validation data-set-averaged $\hat{\varepsilon}_{\text{validation}}$, i.e.

$$\lambda^{k+1} = \lambda^k - \mu^k \, \hat{\varepsilon}_{\text{validation}}$$

Note that $\hat{\varepsilon}$ is the average of all ε, which were defined as batch averages. Furthermore, the unconstrained problem contains no regularisation terms, and $\mu^k = 0 \wedge \lambda^k = 0$ represents a penalty-free problem. Further discussion on μ^k, λ^k, and the initial guess for the unconstrained problem is given in Sec. 3.3.3.1 and Sec. 3.3.3.4 due to training-specific choices.

3.3.3 Network Hyperparameters

With the feature set established, alongside the objective function and network type, additional information on network hyperparameters is needed to finalise the network. This subsection covers initial guesses, optimiser choice and learning rate, batch size and number of epochs, how the constrained problem is updated, activation functions, and number of neurons and hidden layers.

3.3.3.1 Unconstrained Problem Initial Guess

The default Tensorflow layer and bias initialisation routines led to poor-performing networks when directly trained for the unconstrained problem based on preliminary tests. To construct an adequate initial guess, the networks were trained using the penalty-free problem ($\mu^k = 0 \wedge \lambda^k = 0$) before switching to the unconstrained problem ($\mu^k \neq 0$) after a number of epochs. These penalty-free networks were initialised using the default Tensorflow layer and bias configurations, i.e. using the Glorot uniform initialiser and zeros initialiser respectively. More information regarding the switch between the two problems can be found in Sec. 3.3.3.4.

3.3.3.2 Optimiser Choice and Learning Rate

The Adam algorithm was used as the optimiser with little consideration of other solvers based on documentation remarks on Adam being computationally efficient for large data/parameters[3]. The default configuration was used with exception to the learning rate.

The cyclical learning rate algorithm from Smith [21] was implemented, with the `triangular2` policy and a step size equal to eight times the number of iterations in an epoch (i.e. 8 epochs). The learning rate varies between 10^{-3} and 10^{-4}, and the algorithm was modified to restart after a user-defined epoch number at a learning rate factor of ten smaller. The learning rate curve is illustrated in Fig. 3.6, where the custom `triangular2` policy is compared to the standard `triangular2` policy.

[3]Keras: Adam [last accessed 2023-05-01]

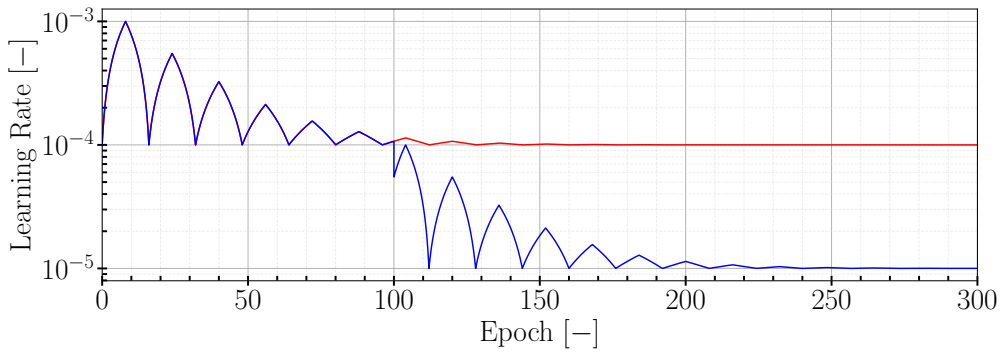
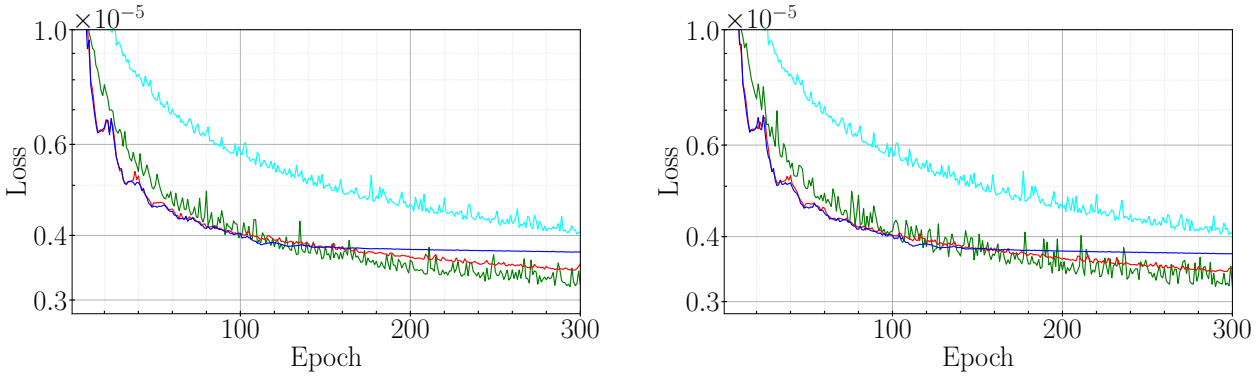


FIGURE 3.6 Example learning rates over 300 epochs, restarted at epoch 100. — Standard Cyclical Learning Rate, — Custom Cyclical Learning Rate

A cyclical learning rate was chosen to unnecessitate learning rate tuning. A user-defined epoch restart was implemented to allow the $\mu^k \neq 0$ solution to evolve at a slower rate than the $\mu^k = 0 \wedge \lambda^k = 0$ solution, mainly to allow for aggressive update strategies while maintaining low training time. Figure 3.7 compares validation losses of neural networks trained for the $\mu^k = 0 \wedge \lambda^k = 0$ problem and the L_2 data set when utilising different learning rates. Both Fig. 3.7a and Fig. 3.7b show similar validation losses for the closure term predictions and the force-free weak residual predictions. The findings of Smith [21] concur with the `triangular2` policies being initially aggressive policies. The policies show faster convergence than the aggressive learning rate of 10^{-3}, even when the standard `triangular2` policy asymptotically decays to a learning rate of 10^{-4}. Additionally, the noise that is present in the standard `triangular2` policy is smaller than the learning rate of 10^{-3}, suggesting fewer update overshoots due to smaller learning rates. The noise level is similar to the learning rate of 10^{-4}. The custom `triangular2` policy demonstrates less noise at the cost of rate of convergence after the first hundred epochs. The difference in validation loss in the first hundred epochs of the standard and custom `triangular2` policies is attributed to using non-deterministic data mapping. The mapping yields better TensorFlow performance when reading data from disk.



(A) $\left[\varepsilon^{M}_{x,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{CT} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{CT} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{CT}$ (B) $\left[\varepsilon^{M}_{x,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{y,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{M}_{z,q}\right]^{MSE}_{FFWR} + \left[\varepsilon^{C}_{q}\right]^{MSE}_{FFWR}$

FIGURE 3.7 Example ANN validation losses for the ($\mu^k = 0, \lambda^k = 0$) problem with L_2 data set using different learning rates for shape function $q = 1$. Neural-network parameters: 5 hidden layers, each with 600 ReLU-activated neurons, linearly-activated output layer, batch size of 2500. — Standard Cyclical Learning Rate, — Custom Cyclical Learning Rate, — Learning Rate 10^{-3}, — Learning Rate 10^{-4}

3.3.3.3 Batch Size and Epochs

Iterative improvements on the model showed that utilising smaller batch sizes yield stronger a priori correlations between prediction and truth at the cost of training time. All intermediate networks trained for this thesis were trained on the L_2 data set using a batch size of 2500 and 300 epochs, which gave subjectively strong correlations for the training time needed. As the final neural networks were also trained on the L_1, L_3, and L_4 data sets, the batch size was increased to 10 000 to maintain the same number of batches evaluated per epoch. The number of epochs was not altered.

3.3.3.4 Constrained Problem Updates

Figure 3.7 illustrates that the network convergence with the $\mu^k = 0 \wedge \lambda^k = 0$ problem and the L_2 data set begins to decay heavily around 100 epochs using the standard cyclical learning rate. It is assumed that the decay for the $\mu^k = 0 \wedge \lambda^k = 0$ problem associated with all data sets combined also occurs at around 100 epochs. Therefore, the solution at the start of epoch 100 served as the initial guess for the $\mu^k \neq 0$ problem. An aggressive strategy for the updates on μ^k and λ^k was chosen to maintain low training time: all μ^k were updated every 25 epochs by a factor of 2 *after* epoch 100, i.e.

$$\beta = \begin{cases} 2 & n_e \bmod 25 \equiv 0 \\ 1 & \text{else} \end{cases}$$

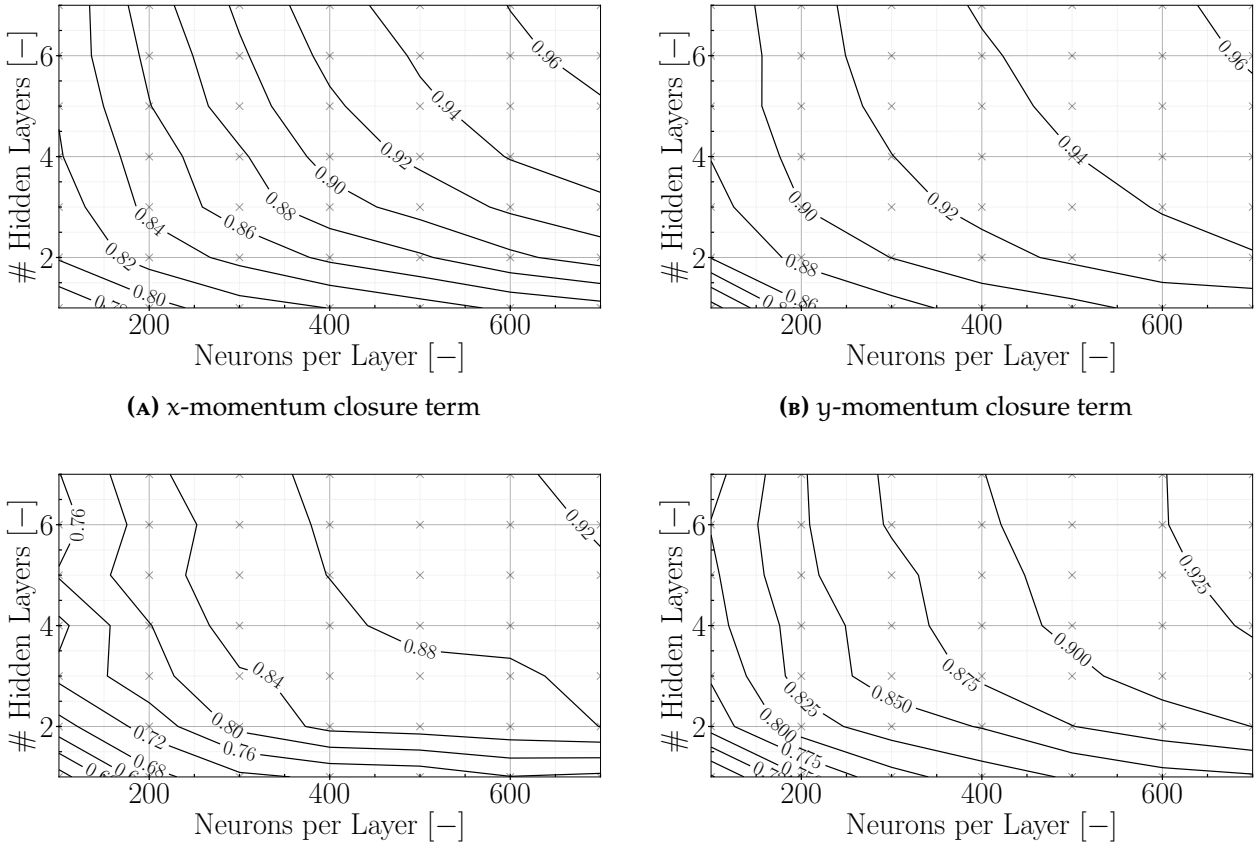
where n_e is the epoch. All λ^k were updated at the end of each epoch *starting from* epoch 100. The starting conditions are $\mu^0 = 2$, $\lambda^0 = 0$ for all penalties. The choice to update μ^k every 25 epochs was subjectively chosen. This number was large enough such that the effects of the penalty enforcement were visibly clear on the losses.

3.3.3.5 Activation Functions

All hidden layers utilised ReLU-activated neurons due to fast solver evaluation times. More complex activation functions that behave better in the negative input-plane of the activation function were considered, such as Leaky ReLU and Softplus. However, these additional functions either introduced new hyperparameters to consider or are expensive to evaluate in the in-house solver.

3.3.3.6 Neurons and Layers

Selecting the number of neurons for each hidden layer of the ANNs required an understanding of the accuracy of the ANNs in relation to the number of neurons and hidden layers. The accuracy of the ANNs is defined as the minimum Pearson correlation coefficient achieved between prediction and truth. The Pearson correlation coefficient is given as a function of channel height level. Figure 3.8 shows the minimum achieved correlations for the $\mu^k = 0 \wedge \lambda^k = 0$ problem with L_2 data set.



(A) x-momentum closure term

(B) y-momentum closure term

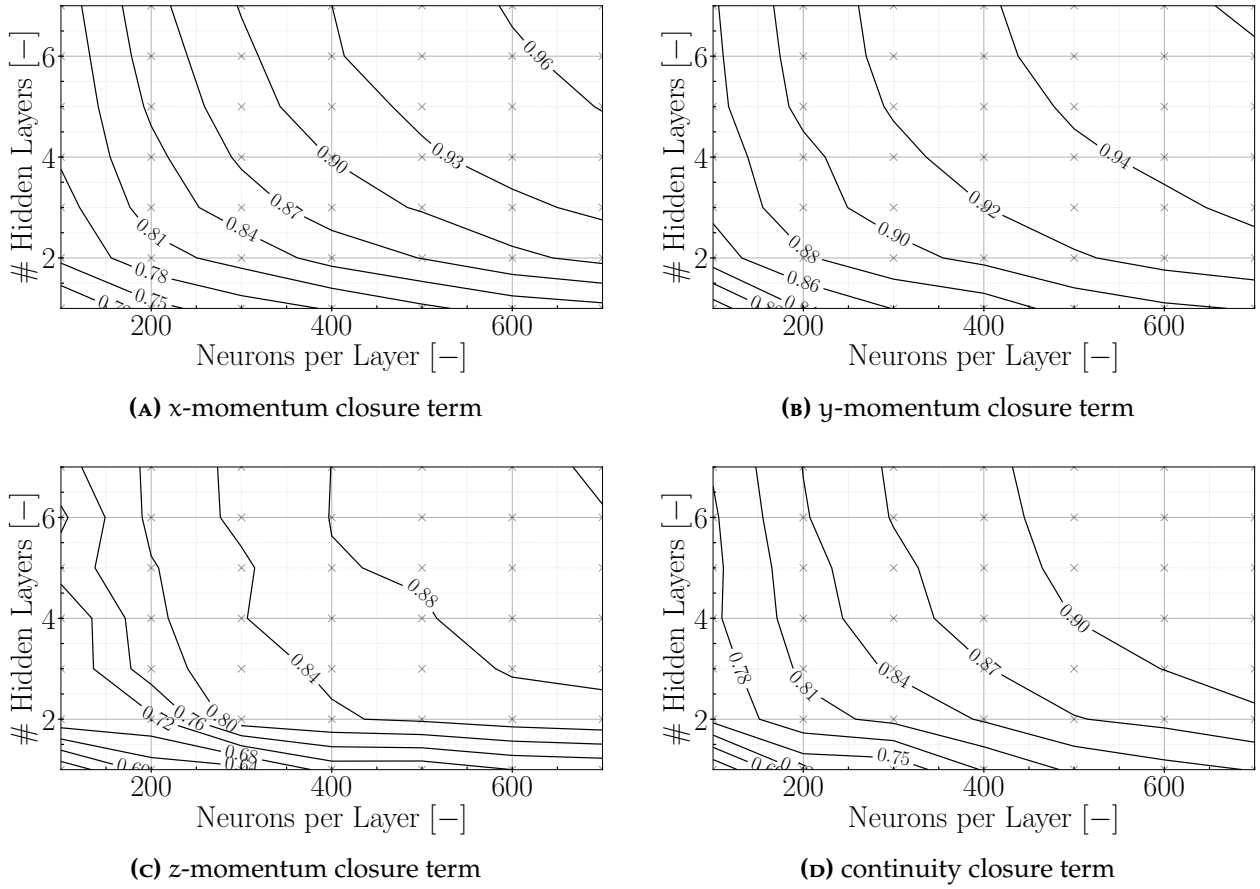
(C) z-momentum closure term

(D) continuity closure term

FIGURE 3.8 Grid search between neurons and layers for the $\mu^k = 0 \wedge \lambda^k = 0$ problem with L_2 data set for shape function $q = 1$. Measured metric is the minimum achieved Pearson correlation coefficient between prediction and truth, where the correlation coefficient is given as a function of channel height. $\times$ Grid-search evaluation point

All four figures demonstrate that better accuracy is achieved by creating both deeper and wider neural networks. This is in agreement with the statement by Beck, Flad, and Munz [18] regarding increased network accuracy via increased network depth. From Fig. 3.8a and Fig. 3.8c, increasing the number of neurons and hidden layers proportionally results in better accuracy for the x- and z-momentum closure terms. In comparison, Fig. 3.8b and Fig. 3.8d show that the y- and continuity closure terms are made more accurate by increasing the number of neurons over the number of hidden layers.

Figure 3.9 shows the minimum achieved Pearson correlation coefficient for the $\mu^k \neq 0$ problem with L_2 data set. The addition of the penalties into the objective function does not affect the correlational contours significantly.



(A) x-momentum closure term

(B) y-momentum closure term

(C) z-momentum closure term

(D) continuity closure term

FIGURE 3.9 Grid search between neurons and layers for the $\mu^k \neq 0$ problem with L_2 data set for shape function $q = 1$. Measured metric is the minimum achieved Pearson correlation coefficient between prediction and truth, where the correlation coefficient is given as a function of channel height. $\times$ Grid-search evaluation point

From the limited results of Fig. 3.8 and Fig. 3.9, the most accurate neural network design considered was the network containing seven hidden layers and 700 neurons. This is at the expense of user elapsed time when calling the ANN function, with example evaluation times given in Fig. 3.10 for a single ANN call. Based on Fig. 3.10, the aforementioned most accurate network costs a user 1.1 ms to evaluate the closure terms. This corresponds to

~287.6 s per time step when considering eight network evaluations for each shape function and 32^3 elements.

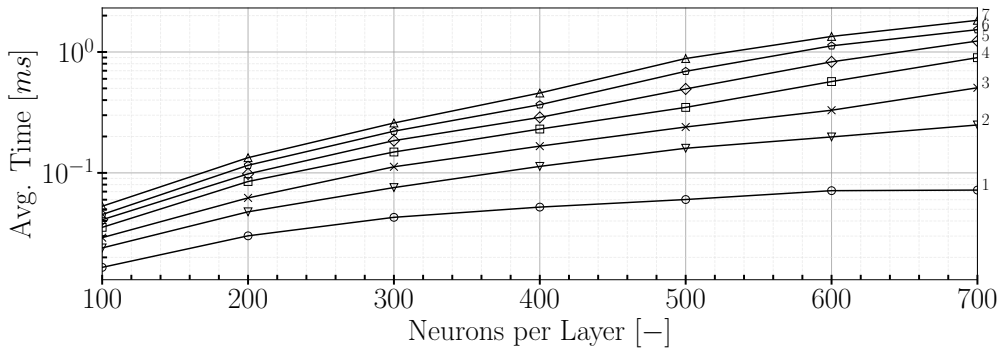


FIGURE 3.10 Average elapsed user time per ANN evaluation using the in-house solver's MLP-ANN object for shape function $q = 1$. Average based on 10 000 different inputs using a single CPU core per ANN call. Number of hidden layers given on the right margin.

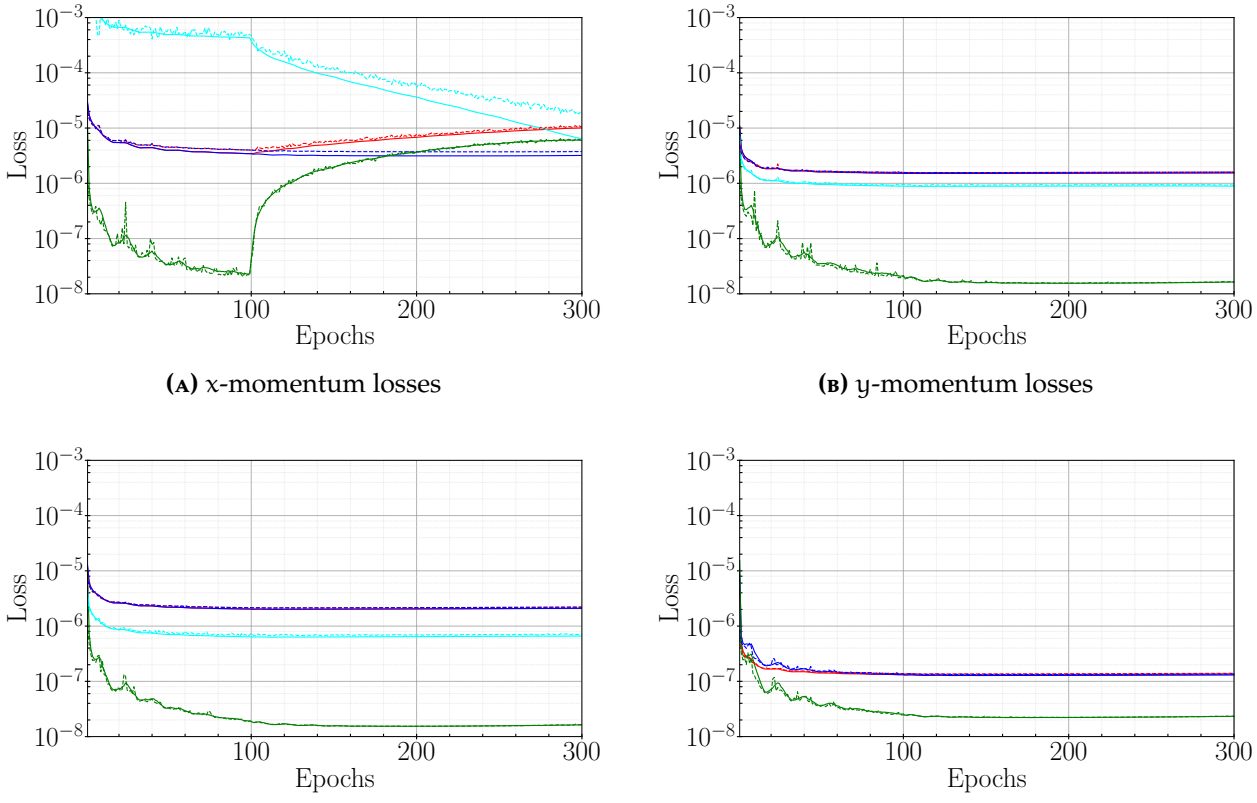
A total of 200 seconds spent on ANN calls per time step is acceptable for single-core performance. From Fig. 3.10, five hidden layers and 600 neurons per hidden layer achieves this requirement. This provides a good ratio between total data points to trainable parameters of ~19.6 : 1 (30 932 992 : 1 579 811) per neural network.

4 A priori Evaluation

This chapter is dedicated to the discussion of the superset networks in an a priori setting. In particular, the training history of the neural network and the correlational coefficients of the closure terms are discussed. A brief discussion on the effects of data augmentation is provided as well.

4.1 Training History

The training history provides behavioural information on the neural network regarding underfitting and overfitting. Additional information on the violation of the constraints through the penalty functions is also given (see Sec. 3.3.2.3 and Sec. 3.3.2.4). Figure 4.1 displays the training history for the $q = 1$ neural network, being an effective representation for all networks in the weak superset. The training history associated to the strong superset are not illustrated as the losses are similar to the weak superset history.



(A) x-momentum losses

(B) y-momentum losses

(C) z-momentum losses

(D) Continuity losses

FIGURE 4.1 Training history of the neural network associated to shape function $q = 1$. The four minor ticks in each decade are evenly spaced. — Closure-term loss, — Force-free weak-residual loss, — Solver-adherence penalty metric, — Energy penalty metric

All closure term losses after epoch 100 suggest the lack of overfitting, as the difference between the validation and training curves does not increase in each new epoch. The slight mismatch between the training and validation curves for the x-momentum closure term loss and x-force-free weak-residual loss before epoch 100 suggests the model has difficulties associating the validation features when trained with the current training data sets. It is uncertain if the cause is related to underfitting or overfitting. Regardless, all four figures demonstrate rapid improvements in the error functions within the first 50 epochs before a decay in the improvement rate up to epoch 100. The metrics measuring the penalty violation also show a general improvement even when the constraints are not present in the objective function at that current epoch. The oscillatory-like behaviour is due to the (custom) cyclical learning rate used.

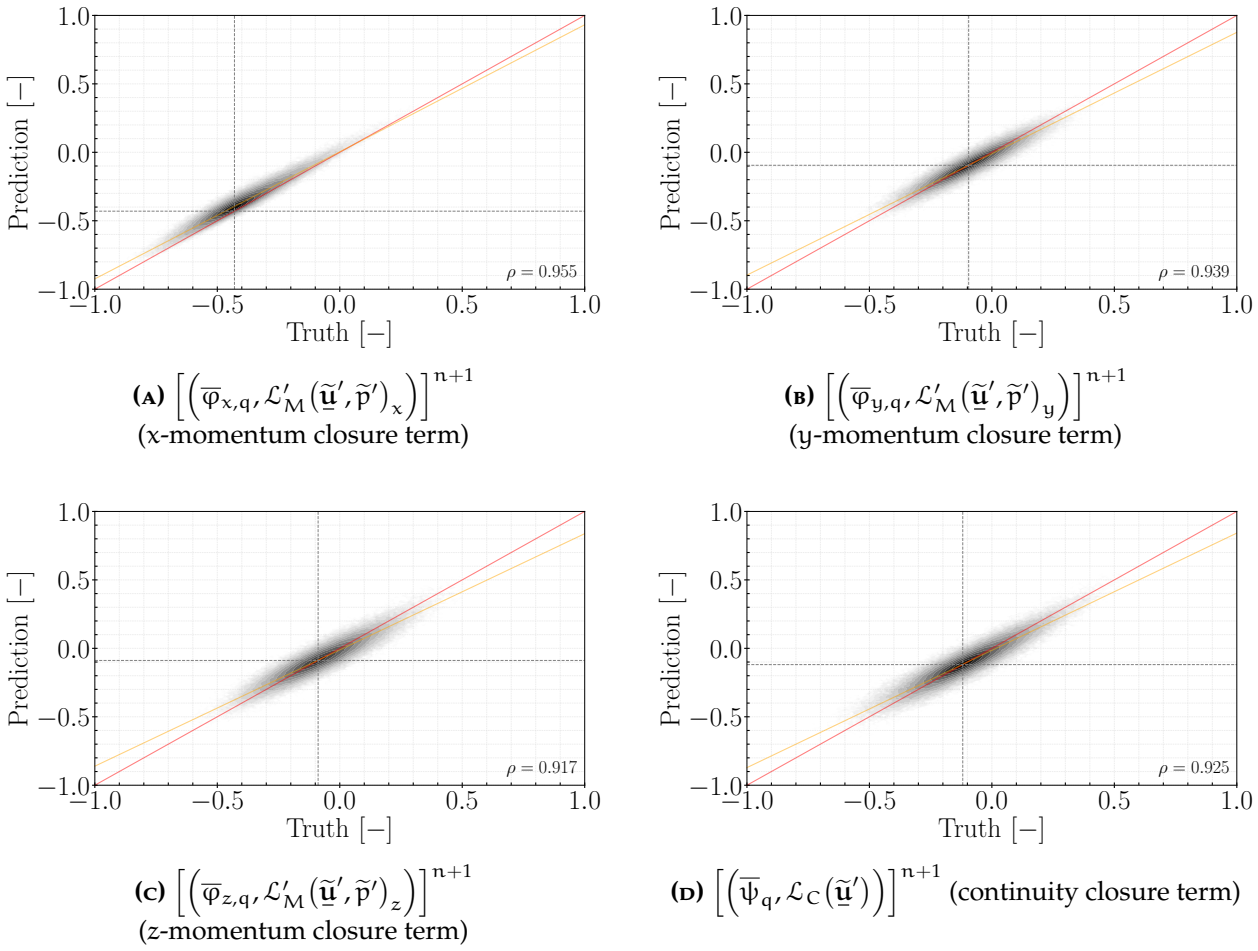
Starting from epoch 100, the constraints appear in the objective function and there are variations in the loss trends between the subfigures of Fig. 4.1. Of the four subfigures, Fig. 4.1b, Fig 4.1c and Fig 4.1d show no visible improvements/degradation in losses. Instead, Fig. 4.1a shows degradation in the solver adherence due to a degradation in the closure term prediction while improving on the energy transfer penalty.

The degradation in the x-momentum closure term prediction is a direct consequence of the energy penalty dominating the objective surface, driving the weight and bias updates in the direction which minimises the penalty over the closure term. The conflict between the two terms exists as the x-energy penalty biases the closure term regressor whereas the x-momentum closure term error attempts to reduce this bias. As the adherence penalty is significantly smaller than all other terms, this term shows the largest degradation as it is unable to bias the x-momentum force-free weak-residual term.

The lack of degradation/improvement in the y-momentum, z-momentum, and continuity surfaces in response to the x-penalty dominating the objective function is hypothetically attributed to the lack of an explicit relationship between the x-momentum terms and the other terms in the objective function. Therefore, the rest of the surfaces are unaffected by the updates in the x-momentum surface. The x-energy penalty is similarly not influenced by predictions associated to other closure terms, hence the x-surface is unaffected by the rest of the surfaces. As the learning rate after epoch 100 is also significantly smaller than the standard cyclical learning rate, as was shown in Fig. 3.6. Any visual improvements in the losses are indistinguishable, as was displayed in Fig. 3.7.

4.2 Truth-Prediction Correlations

The correlations between closure term prediction and truth are provided in Fig. 4.2 for the weak superset and in Fig. 4.3 for the strong superset. The correlations are quantified through the Pearson correlation coefficient as a single scalar rather than a function of channel height.



(A) $\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}$ (x-momentum closure term)

(B) $\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}$ (y-momentum closure term)

(C) $\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}$ (z-momentum closure term)

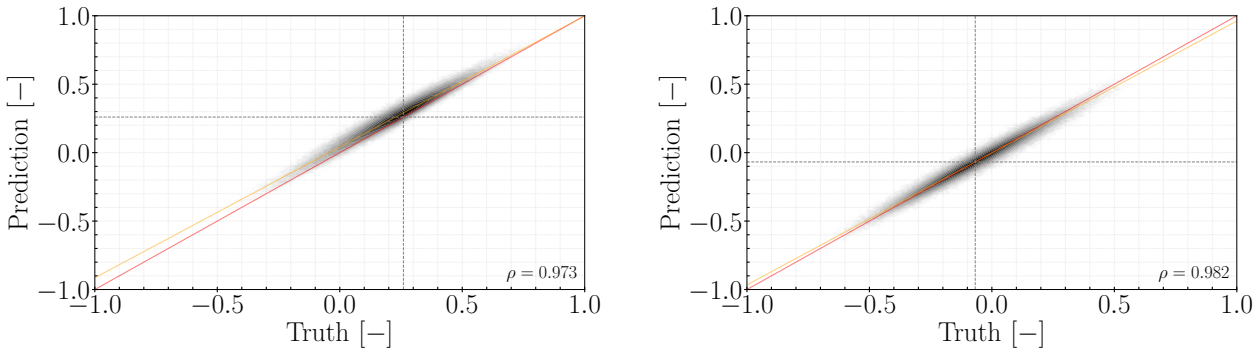
(D) $\left[\left(\overline{\Psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}$ (continuity closure term)

FIGURE 4.2 Closure term prediction against truth values for shape function $q = 1$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes

Both Fig. 4.2 and Fig. 4.3 show $\rho > 0.90$ correlations in all closure terms, with the z-momentum or continuity closure term being the weakest of the correlations. The continuity closure term was expected to be the weakest correlated term. This is due to the lack of utilisable features attached to Eq. 3.4 in comparison to Eq. 3.3, regardless of superset choice. It is believed that the continuity spatiotemporal stencil is responsible for providing better correlations than the z-momentum closure term for $q = 1$, as the neural network finds information from the spanwise and wall-normal neighbours impactful. These are elements that the z-momentum closure term are not influenced by. The two figures demonstrate that the momentum closure terms provided by the strong superset are better correlated than the weak superset. This is attributed to the stronger correlations between the features and closure terms (see App. A).

The prediction trends in the subfigures of Fig. 4.2 and Fig. 4.3 show similarities between both supersets. The x-momentum closure term predictions are overpredicted for the full range of closure terms that the networks are trained on except at large positive estimates,

where both networks switch to underprediction. This is concluded based on where the network-trend line intercepts the perfect-trend line. The main difference between the weak and strong superset is when underprediction occurs. This can happen for smaller values for the weak superset in comparison to the strong superset. The predicted y-momentum, z-momentum, and continuity closure terms are all underpredicted(/overpredicted) for small(/large) closure term estimations.



(A) $\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}$ (x-momentum closure term)

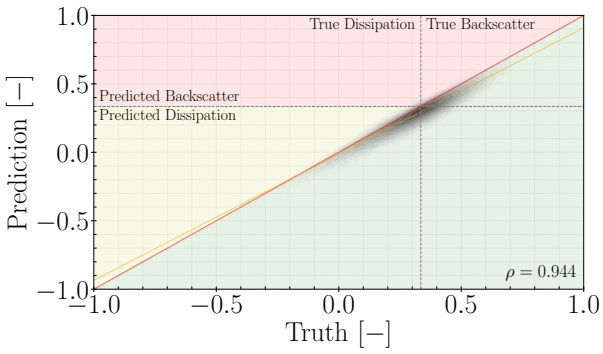
(B) $\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}$ (y-momentum closure term)

(C) $\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}$ (z-momentum closure term)

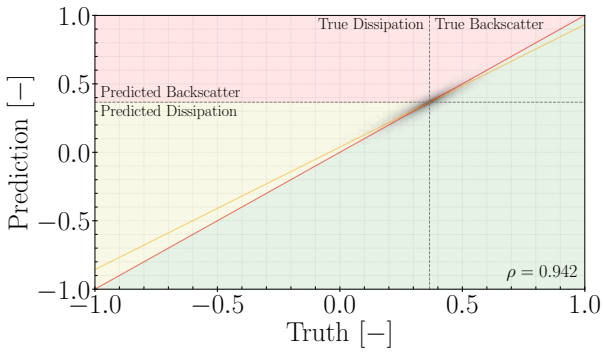
(D) $\left[\left(\overline{\Psi}_q, \mathcal{L}_C(\widetilde{\underline{u}}')\right)\right]^{n+1}$ (continuity closure term)

FIGURE 4.3 Closure term prediction against truth values for shape function $q = 2$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes

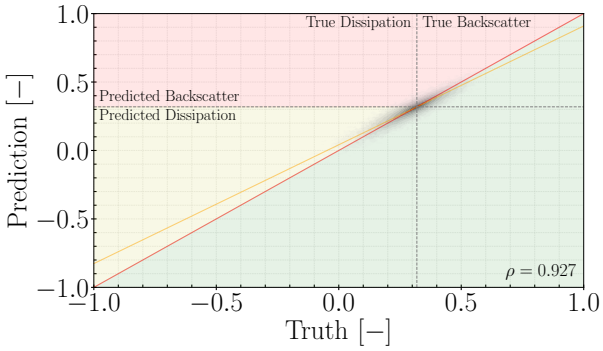
It is easier to interpret the offset significance from Fig. 4.2a and Fig. 4.3a in the context of the energy penalty. To this extent, each closure term prediction composing Fig. 4.2 and Fig. 4.3 is scaled by the velocity basis coefficient attached to the corresponding basis function. This retrieves the closure-term energy transfer contribution as discussed in Sec. 3.3.2.4, and is shown in Fig. 4.4 and Fig. 4.5.



(A) $\left[-\widetilde{\overline{\alpha}}_{x,q}\right]_{\text{truth}}^{n+1}\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\underline{\widetilde{u}}', \widetilde{p}')_x\right)\right]^{n+1}$
(x-momentum closure term energy transfer)



(B) $\left[-\widetilde{\overline{\alpha}}_{y,q}\right]_{\text{truth}}^{n+1}\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\underline{\widetilde{u}}', \widetilde{p}')_y\right)\right]^{n+1}$
(y-momentum closure term energy transfer)



(C) $\left[-\widetilde{\overline{\alpha}}_{z,q}\right]_{\text{truth}}^{n+1}\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\underline{\widetilde{u}}', \widetilde{p}')_z\right)\right]^{n+1}$
(z-momentum closure term energy transfer)

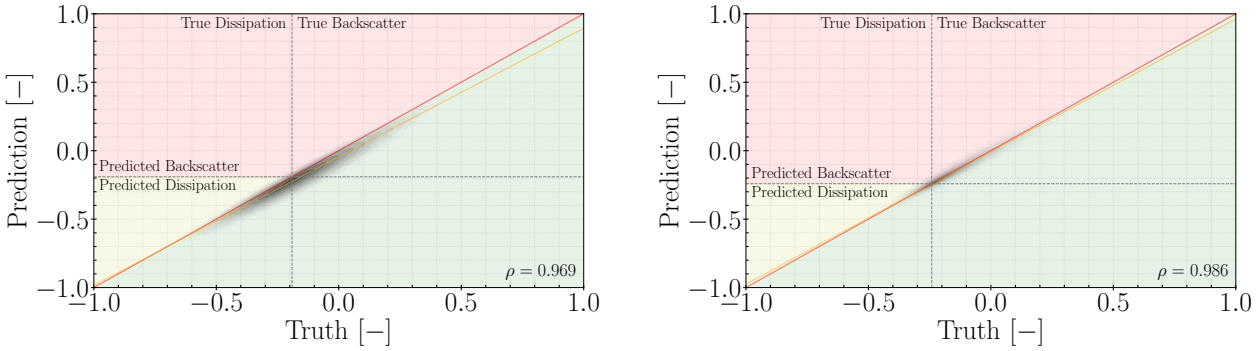
FIGURE 4.4 Closure term energy transfer prediction against truth values for shape function $q = 1$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes

All subfigures contain colour-coded regions representing excess backscatter (red), insufficient dissipation (yellow), and excess dissipation (green). Excess backscatter predictions occurs when the predicted energy transfer is both positive and larger than the true energy transfer, whereas insufficient dissipation occurs when the prediction is both negative and larger than the truth. Excess dissipation occurs when the prediction is smaller than the truth. Of the three colour-coded regions, it is preferred that the data-driven outputs are excessively dissipative as the net energy of the system remains bounded, regardless of whether energy is added (the net energy transfer of the true system will remain larger than the predicted system). Insufficient dissipation is acceptable as energy is pulled from the system, but not enough energy is pulled by the closure term. Excessive backscatter is unfavourable as there is a non-physical increase in the system's energy.

All subfigures show that there are closure terms that add excess energy into the TCF system. This is a primary concern for the dominant energy transfer mode, i.e. the x-closure terms, as the energy added into the system from these closure terms is significantly larger than

the y- and z-closure terms. Despite excess backscatter being occasionally predicted, the general x-energy transfer trend is to predict smaller transfer values than expected, which is the result of the energy penalty biasing the outputs during training. Both supersets have exceptions for large negative energy transfers, but this exception represents the trend belonging to a small number of samples.

The y- and z-energy transfer graphs instead show a balance between excessive dissipation and insufficient dissipation / excess backscatter. In principle this is a concern as excess energy will be retained(/added) in(to) the system. In practice, this concern is secondary due to the aforementioned statement about the x-closure terms dominating the energy transfer.



(A) $\left[-\widetilde{\overline{a}}_{x,q}\right]^{n+1}_{truth}\left[\left(\overline{\varphi}_{x,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_x\right)\right]^{n+1}$
(x-momentum closure term energy transfer)

(B) $\left[-\widetilde{\overline{a}}_{y,q}\right]^{n+1}_{truth}\left[\left(\overline{\varphi}_{y,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_y\right)\right]^{n+1}$
(y-momentum closure term energy transfer)

(C) $\left[-\widetilde{\overline{a}}_{z,q}\right]^{n+1}_{truth}\left[\left(\overline{\varphi}_{z,q}, \mathcal{L}'_M(\widetilde{\underline{u}}', \widetilde{p}')_z\right)\right]^{n+1}$
(z-momentum closure term energy transfer)

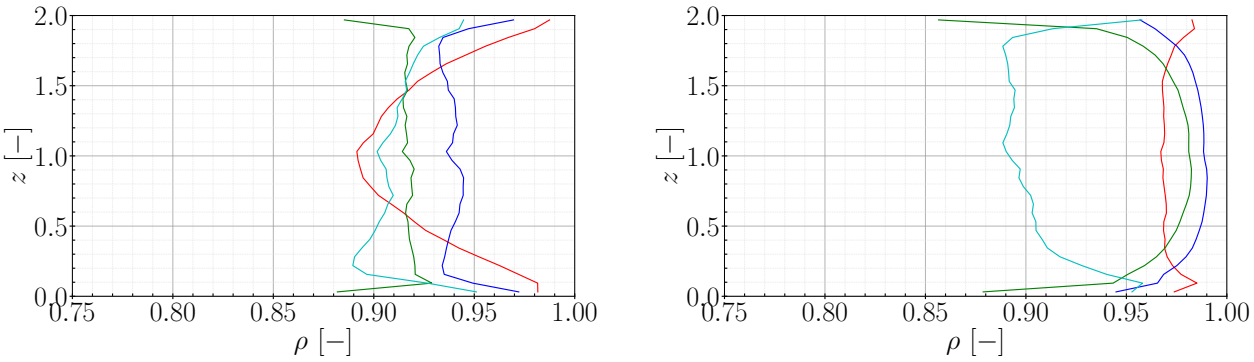
FIGURE 4.5 Closure term energy transfer prediction against truth values for shape function $q = 2$ in histogram format based on 500 000 samples, data normalised between -1.0 and 1.0 relative to the truth. Pearson correlation coefficient is given in the bottom right corner of the graphs. — Perfect-relation line, — Network-trend line, - - - Denormalised data axes

In the context of the prediction trend line, it is hypothesised that *any* deviation from the perfect-relation trend line will yield temporal instabilities. This is due to erroneous closure term predictions driving the TCF system sufficiently far from its training conditions

over time. These instabilities are expected to manifest themselves as high-frequency perturbations (excess backscatter) or incorrect flow statistics (insufficient dissipation and/or excessive dissipation). The former leads to an accumulation of energy within the system, whereas the latter may lead to closure-term underprediction (energy overprediction) stemming from the flow conditions becoming significantly different than the conditions used to train the network. Both instabilities eventually grow to energy divergence.

4.3 Channel Predictions

The network correlations given in Fig. 4.2 and Fig. 4.3 are visualised in Fig. 4.6 as a function of channel height. The two subfigures show asymmetric correlations around the centre-channel arising from transient mean-flow behaviour (see Fig. 3.5). The asymmetric closure-term statistics across each plane are shown in App. B. The mirrored $q = \{5, 8\}$ and $q = \{6, 7\}$ graphs are not shown (when compared to $q = \{1, 4\}$ and $q = \{2, 3\}$ respectively).



(A) $q = 1$ network

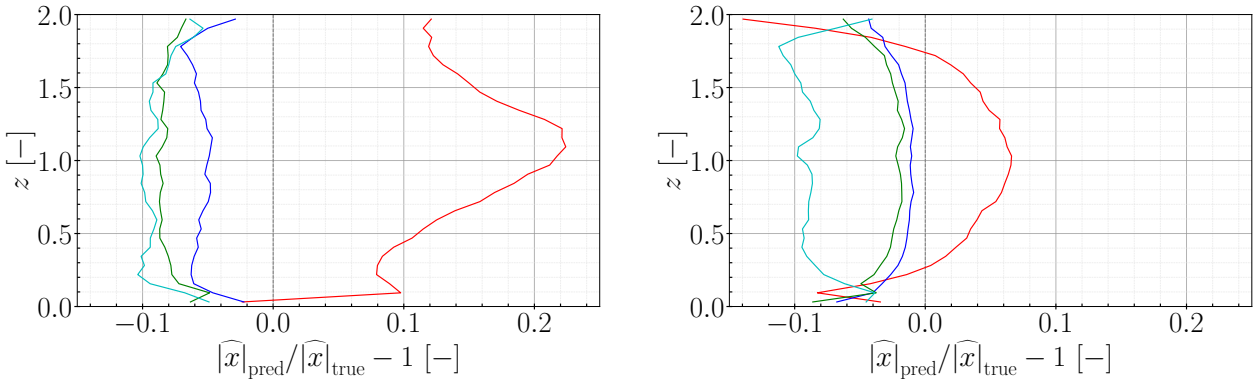
(B) $q = 2$ network

FIGURE 4.6 Channel plot of closure term correlations for the $q = 1$ and $q = 2$ networks. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term

Both superset networks predict strongly-correlated near-wall continuity closure terms, but struggle in predicting the centre-channel terms. Between the two supersets, the weak superset provides slightly better continuity closure terms in the centre-channel. In the context of the momentum closure term predictions, the strong superset network provides better centre-channel momentum closure term, while the weak superset network provides better near-wall momentum closure terms.

The network bias is visualised in Fig. 4.7, displaying the prediction mean magnitude as a fraction of the truth mean magnitude. Both networks underpredict the magnitude of the closure terms with the sole exception of the x-momentum closure term, where magnitude overestimation is present. The magnitude deviation is a direct representation of the prediction trend lines in the closure term figures of Sec. 4.2. In particular, prediction trends that are less steep than the perfect-relation line or have a negative intercept (relative to the prediction denormalised data axis) yield magnitude underestimation. In contrast, magnitude overestimation is given from steeper slopes or positive intercepts. The training

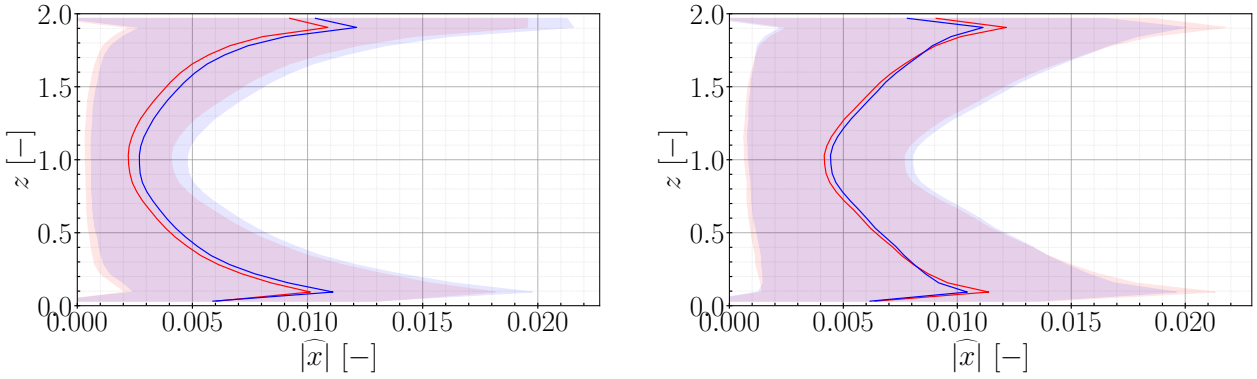
mechanism leading to magnitude underprediction is not known, but the overestimation comes from the energy penalty which pushes for a positive line intercept.



(A) shape function $q = 1$ **(B)** shape function $q = 2$

FIGURE 4.7 Channel plot of closure term fractional deviation for the $q = 1$ and $q = 2$ networks. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term

The nature of the magnitude overestimation difference between the centre-channel and near-wall is partly attributed to the magnitude of the closure term. As the magnitude of the closure terms is smaller in the centre-channel as displayed in Fig. B.1, any fractional deviation measured relative to the truth is more sensitive than the fractional measurements in the near-wall regions. The rest of the attribution is unknown. The x-momentum deviation is the only deviation which demonstrates a large change between the near-wall and centre-channel, despite the other deviations displaying smaller magnitudes in the centre-channel. The large difference in the fractional deviations for the x-momentum suggests that both superset networks struggle in learning the behavioural patterns in the centre-channel.



(A) $q = 1$ network **(B)** $q = 2$ network

FIGURE B.1 Channel profile of the mean x-momentum closure term. Shaded region denotes ±1 standard deviation. — Truth, — Prediction (repeated from page 261)

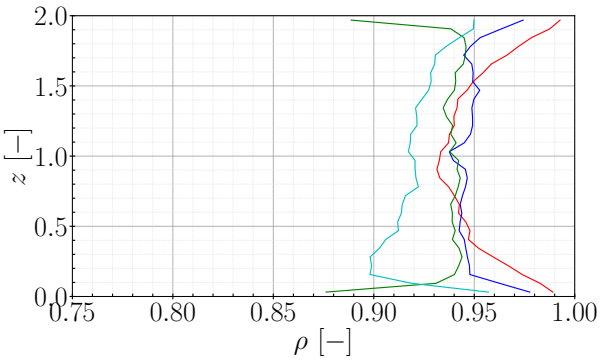
The energy transfer channel plots using the quantities given in Figs. (4.4–4.5) are not provided due to the near-similarity with the closure term channel plots of this section. The remaining channel plots displaying the mean magnitudes of the y-momentum, z-momentum, and continuity closure terms are provided in App. B.

4.4 Data Augmentation Effects

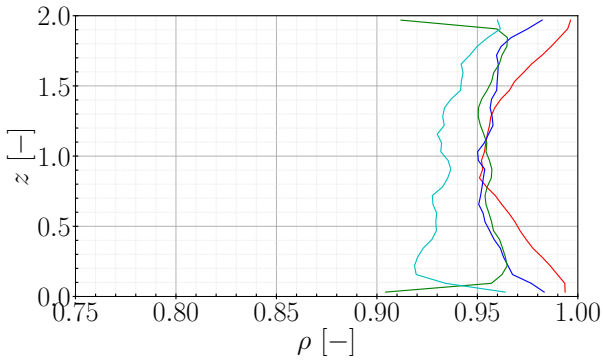
The purpose of the data augmentation through the wall-normal distance was to prevent a dependency on the choice of wall from which to measure the wall-normal distance from. To understand the effects of the data augmentation, additional networks were trained on unaugmented L_2 data sets, with correlations provided in Fig. 4.8.

The data augmentation used does not provide correlational symmetry (around the centre-channel) for the neural network, but improves existing correlations. In comparison to the data-augmented networks, both unaugmented superset networks show weaker correlations overall. This is also present to a lesser extent in the momentum correlations. The weaker correlations stemming from the lack of augmentation suggests that the networks struggle to learn the output pattern associated with the wall-normal distance feature.

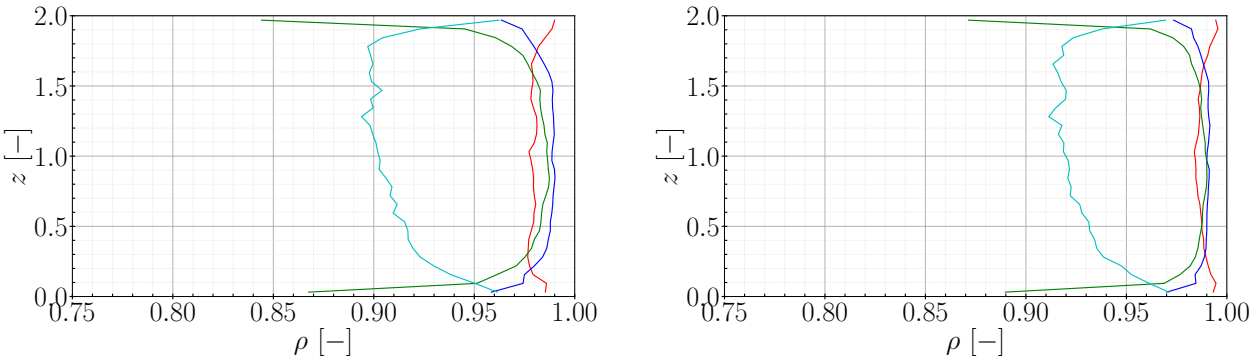
The higher correlations from the superset networks trained on the L_2 data set (Fig. 4.8b and Fig. 4.8d) in comparison to the networks trained on the full data set (Fig. 4.6) are attributed to the bulk velocity history of the data set. As the bulk velocity changes between the four data sets, the networks trained on all data sets are required to account for this variation by regressing to larger/different range of closure terms. The networks have difficulty in accounting for the increased data variation using the current architecture, manifesting in lower correlations.



(A) shape function q = 1, without data-augmentation



(B) shape function q = 1, with data-augmentation



(c) shape function q = 2, without data-augmentation

(d) shape function q = 2, with data-augmentation

Figure 4.8 Channel plot of closure term correlations for the q = 1 and q = 2 networks with data augmentation and without data augmentation. Networks trained on the L_2 data set. — x-momentum closure term, — y-momentum closure term, — z-momentum closure term, — Continuity closure term

5 Preliminary A posteriori Evaluation

This chapter deals with the evaluation of the eight neural networks as the fine-scale model for a running TCF simulation, detailed in Sec. 3.1. The chapter covers the ability of the neural networks in maintaining a bound on the net kinetic energy of the TCF flow, the ability to replicate projected DNS velocity profiles across different time levels, and the accumulated errors between the VMS simulation and the projected DNS flow. For clarity purposes, all ~ notation is dropped off variables to improve readability.

5.1 Chapter Note

It should be noted that at the time of the submitted thesis, the numerical code used to conduct the simulations utilising the neural networks has not been fully verified. While the results shown hereby are reflective of the model in a running simulation, it may be subjected to change post-graduation.

Preliminary observations utilising the neural-network model displayed instantaneous divergence in the Newton iteration passes for a posteriori simulations starting from $t = 1.0368$. This was found to be due to the continuity closure term through trial and error. Switching the continuity closure terms to be driven by an algebraic model (Taylor, Hughes, and Zarins [22]) prevented instantaneous divergence.

As the data-driven model is a regression model, a soft enforcement on interpolation was set by limiting each input to be between the observed maximum and minimum in the training data sets. Harsher enforcements were not imposed, e.g. further limits when considering multiple inputs together.

5.2 Energy Time Series

It was aforementioned in Sec. 3.3.2.4 that the net kinetic energy k within the flow is impacted when the momentum closure term predictions deviate from the truth. Maintaining a bound on the net kinetic energy is required to prevent energy divergence as witnessed in the work of Rajampeta [7]. While Rajampeta addressed energy divergence with backscatter limiting, it negates any form of backscatter within the simulation. The developed fine-scale model in this thesis attempts to bound the net kinetic energy of the flow via a penalty term in the objective function used to train the neural network. Figure 5.1 displays the net kinetic energy as a function of time until divergence for an a posteriori TCF simulation starting from $t = 1.0368$. The figure demonstrates that the fine-scale model provides excessive energy dissipation. The neural network simulation diverges in the Newton procedure after $t = 1.1664$ (time step 19).

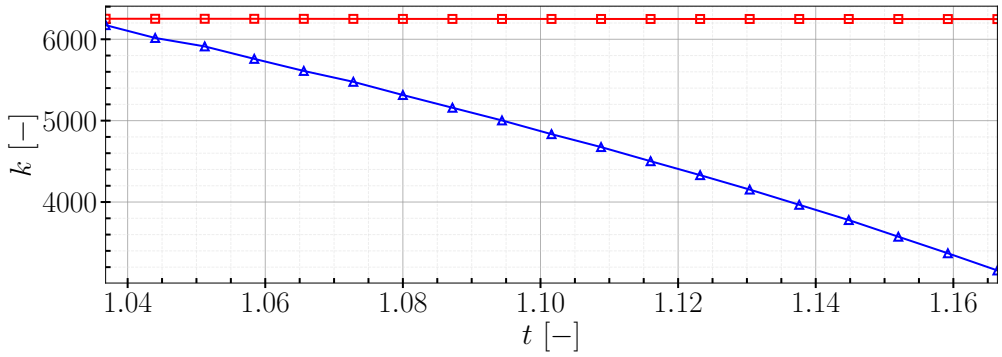
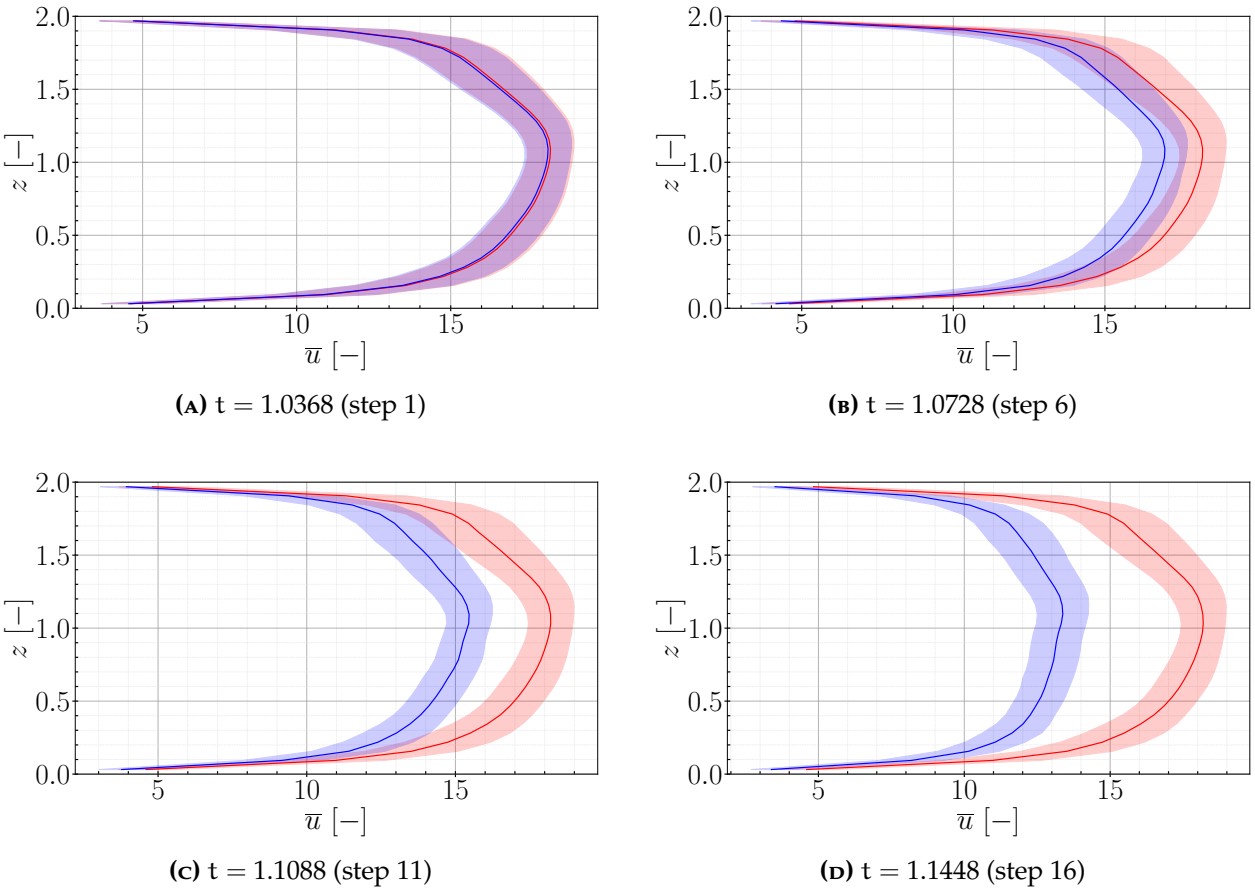


FIGURE 5.1 Net kinetic energy within the TCF domain. Projected DNS, Neural-network model with algebraic continuity closure term and bounded input intervals [22]

The fine-scale model is capable of maintaining the decay in energy sufficiently far away from the conditions of the projected DNS solution. An explanation for this behaviour is that the neural network is (correctly) extrapolating overpredicted closure terms due to the fine-scale model having learnt to be energy dissipative. Alternatively, the neural network may be interpolating overpredicted closure terms when it encounters conditions similar to the training conditions of later time steps, e.g. from the L_3 data set, which displays a smaller bulk velocity in comparison to the L_1 data set (the data set which overlaps the simulation time).

5.3 Wall Profiles

The velocity profiles of the a posteriori simulation using the neural-network model and projected DNS are displayed in Figs. (5.2–5.4), with each figure visualising the mean profile of a separate flow component across four different time steps. Based on Fig 5.2, the neural-network $\overline{u}$ mean profile decays over time, which is also reflected in the net kinetic energy time series as the x-velocity component predominantly determines the net kinetic energy of the domain. The standard deviation / fluctuation profile is comparable to the projected DNS standard deviation despite the difference in conditions used to train the neural network. The decay in the mean profile is attributed to the energy penalty influencing the x-momentum closure term predictions to be excessively energy dissipative.



(A) t = 1.0368 (step 1)

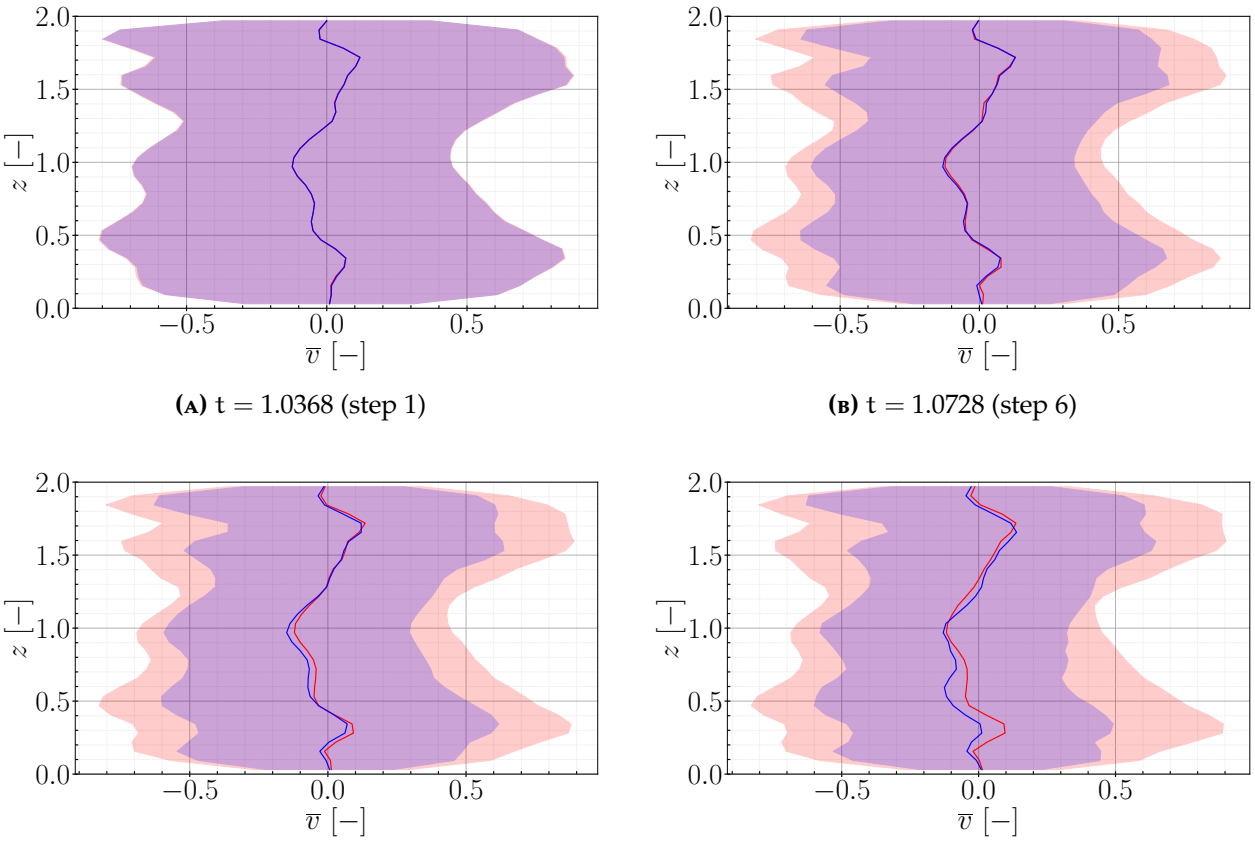
(B) t = 1.0728 (step 6)

(C) t = 1.1088 (step 11)

(D) t = 1.1448 (step 16)

FIGURE 5.2 A posterori $\overline{u}$ mean profile as a function of channel height. Shaded region denotes ± 1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22]

While the neural-network $\overline{u}$ mean profile deviated significantly from the projected DNS $\overline{u}$ mean profile, the neural-network $\overline{v}$ mean profile displayed in Fig. 5.3 is in agreement with the projected DNS $\overline{v}$ mean profile. This is explained by the energy penalty not influencing the y-momentum closure term to be excessively energy dissipative, allowing any (excess) dissipation to be balanced by overpredicted backscatter over the duration of the simulation. It cannot be inferred whether the deviations in the $\overline{v}$ mean profile is a suggestion of backscatter or dissipation dominance. This is due to the existence of cross terms in the momentum equations, allowing energy contributions to be transferred from one flow component to another. However, it is hypothesised that the fluctuation profile is smaller in magnitude in comparison to the projected DNS profile due to the highly dissipative x-momentum terms influencing the y-momentum profile through the cross terms.



(A) t = 1.0368 (step 1)

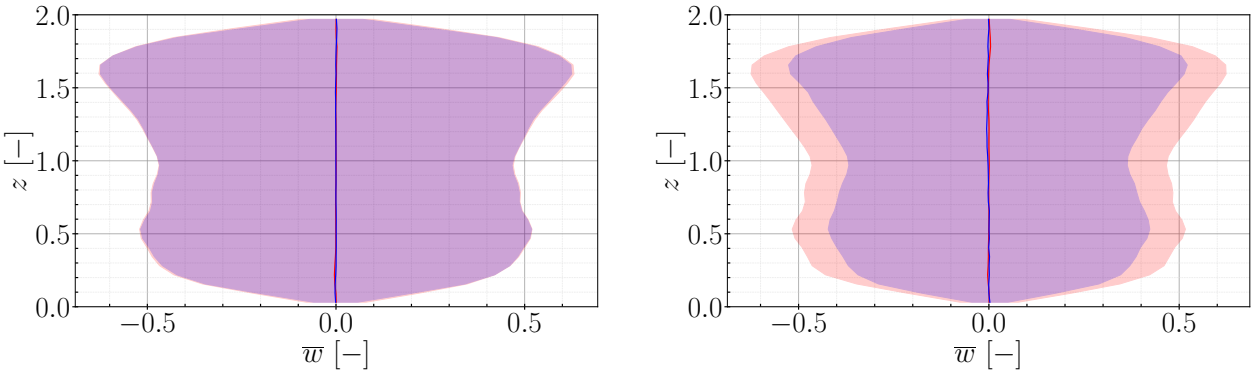
(B) t = 1.0728 (step 6)

(C) t = 1.1088 (step 11)

(D) t = 1.1448 (step 16)

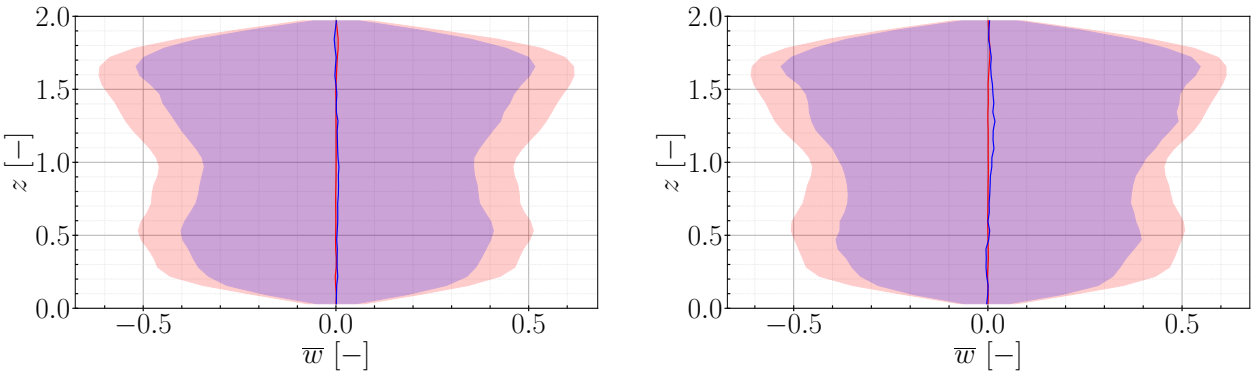
FIGURE 5.3 A posterori $\overline{v}$ mean profile as a function of channel height. Shaded region denotes ±1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22]

Figure 5.4 displays the $\overline{w}$ mean and fluctuation profiles, but most comments on these two profiles are similar to the $\overline{v}$ profiles.



(A) t = 1.0368 (step 1)

(B) t = 1.0728 (step 6)



(c) t = 1.1088 (step 11) **(d)** t = 1.1448 (step 16)

Figure 5.4 A posterori $\overline{w}$ mean profile as a function of channel height. Shaded region denotes ±1 standard deviation. — Projected DNS, — Neural-network model with algebraic continuity closure term and bounded input intervals [22]

5.4 Error Profiles

The accumulation of error stemming from the neural-network model is visualised in Fig. 5.5 as a time series of the root-mean-square (RMS) error between the projected DNS solution and neural-network model solution. $\overline{u}$ has the largest error in comparison to the other velocity components, while $\overline{v}$ and $\overline{w}$ have similar error profiles.

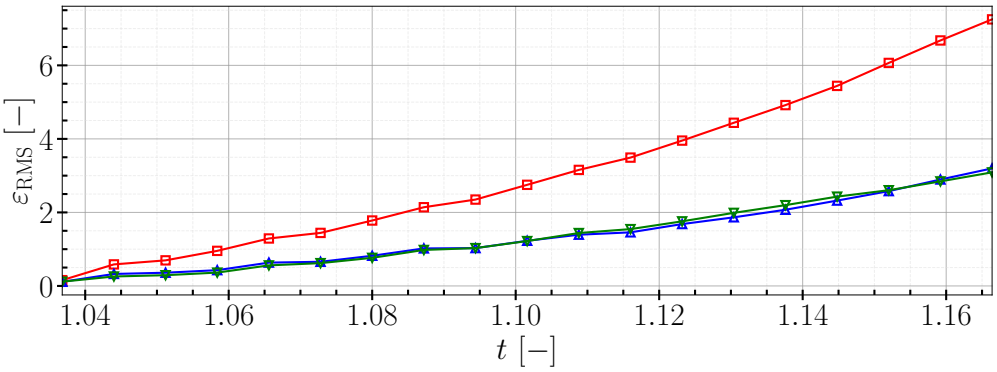
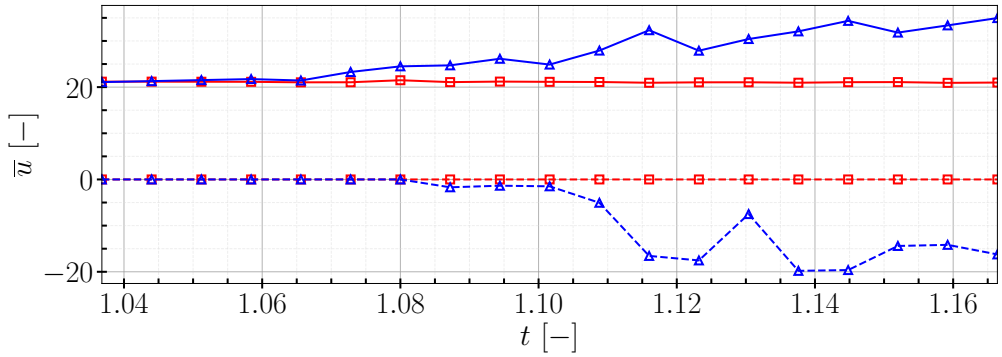


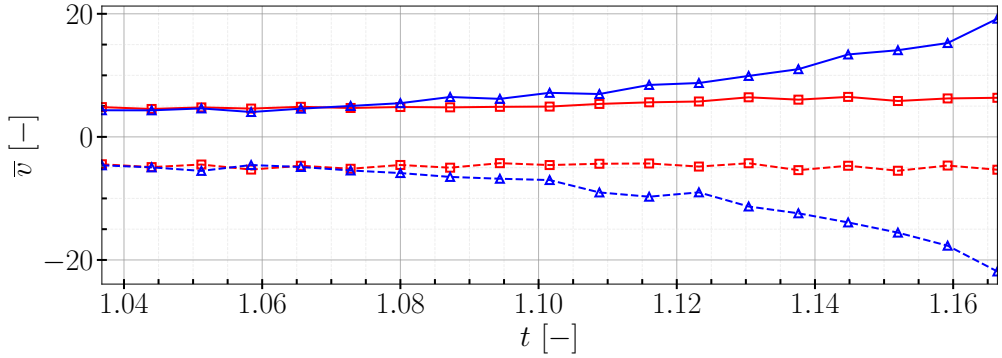
Figure 5.5 RMS error time series between the neural-network model solution and projected DNS solution. — $\overline{u}$ error, — $\overline{v}$ error, — $\overline{w}$ error

While the gradual increase in error for the $\overline{u}$ component could have been predicted based on the net kinetic energy time series, the mean error for both $\overline{v}$ and $\overline{w}$ is demonstrated to gradually increase as well. This is also true for the bounds (the maximum and minimum) of the velocity components, as shown in Fig. 5.6. All three subfigures show deviations from the projected DNS bounds, with $\overline{w}$ gradually growing four times larger than the projected DNS bounds, as displayed in Fig. 5.6c. Similar behaviour is shown for $\overline{v}$, as demonstrated by Fig. 5.6b. Figure 5.6a displays a negative minimum $\overline{u}$ after t = 1.12, implying a flow direction in the upstream direction. All deviations in the flow-component bounds are

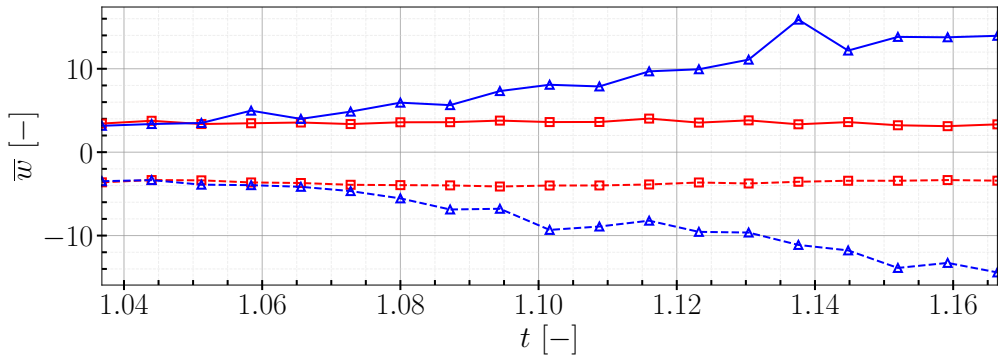
non-physical, stemming from high-frequency velocity oscillations that the backward Euler method was unable to dampen over time.



(A) $\overline{u}$ bounds



(B) $\overline{v}$ bounds



(C) $\overline{w}$ bounds

FIGURE 5.6 Maximum and minimum velocity as a function of time. Projected DNS maximum, Neural-network model maximum (with algebraic continuity closure term and bounded input intervals [22]). and represent minimums.

6 Conclusion

This thesis presented an energy-conservative data-driven approach in modelling the closure terms of the two-scale Navier-Stokes equations for a $Re_\tau = 180$ TCF case discretised into 32^3 elements. Eight neural networks were constructed, each predicting the momentum and continuity closure terms associated with a separate shape function. Each network was constructed using five hidden layers, each with 600 neurons. The neural networks used weak-form coarse-scale terms as part of the feature set, with the wall-normal distanced added in consideration of the success seen in the work of Gamahara and Hattori [17], see Tab. 3.2 and Tab. 3.1. The coarse-scale features were pulled across various local elements, chosen based on how two different fine-scale Greens' functions varied across integration-variable space for a single point in physical space. In an attempt to provide a posteriori simulation stability, the neural-network objective function was biased to provide larger closure term predictions in comparison to the truth values, translating to excessive energy dissipation.

A priori evaluation revealed that the neural networks provided strongly correlated values between prediction and truth ($\rho > 0.90$ for all outputs, see Fig. 4.2 and Fig. 4.3). Additionally, the networks gave larger predictions than truth on average for the x-momentum closure term, which was the dominant term in the fine-scale energy transfer. Between the eight shape functions used, the networks could be categorised into two 'supersets' of four shape functions each, where each set provided similar correlations to one another in the superset. The supersets are described by a strength adjective, denoting how strongly one superset's coarse-scale features correlate to the closure terms relative to the other superset. The adjective serves as a distinction between which superset is being discussed. When considering the neural network outputs as a function of channel height, the weak network provided correlation coefficients above 0.88 (see Fig. 4.6a), and the strong superset provided coefficients above 0.86 (see Fig. 4.6b). The 'weak' superset predicted average x-momentum closure terms that were between 8% and 22% larger than the truth (with exception at $z < 0.1$, which abruptly provides terms up to -2% larger than the truth), while the rest of the terms were between -5% to -10% larger than the truth. The 'strong' superset predicted average x-momentum closure terms that were between -8% to 6% larger than the truth (with exception at $z > 1.9$, which provides terms up to -14% larger than the truth), while the rest of the terms were between -2% to -12% larger than the truth.

An a posteriori simulation of the TCF case with the neural networks acting as the fine-scale model revealed that the model is highly energy dissipative. This was observed through the rapidly decaying kinetic energy held within the domain due to the mean x-velocity profile reducing in magnitude over time. However, the mean and fluctuation profiles of the y- and z-velocities are captured well in time, but the fluctuations tend to be smaller in magnitude than the projected DNS solution. While the mean and fluctuation profiles appear to be captured correctly, high frequency oscillations were observed in the simulation which were not dampened by the backward Euler method. It should be noted that the numerical code

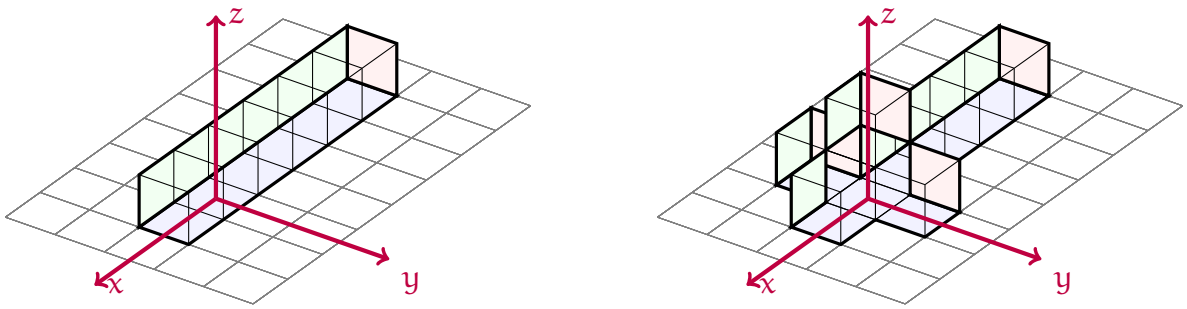
is not fully verified currently and is subjected to change in the future.

6.1 Answers to the Research Questions

The research questions given in Ch. 1 are answered below:

1) What are the differences and similarities between the latest 1D Navier-Stokes VMS-ANN TCF data-driven models and the 3D Navier-Stokes VMS-ANN TCF data-driven models?

   The main similarity between the latest 1D data-driven model and the 3D data-driven model is the MLP architecture used in addition to the usage of a lagged feature set. The main differences are summarised by consulting the work of Rajampeta [7]:
   - **The system of equations:** the 1D model was developed for the Navier-Stokes' x-momentum equation, while the 3D model was developed for the full Navier-Stokes equations.
   - **The depth and width of the networks:** The 1D model used three hidden layers. The first hidden layer used the same number of neurons as inputs, whereas the latter two used 128 neurons each. The 3D model used five hidden layers, each with 600 neurons.
   - **The inputs:** The 1D model utilised a spatiotemporal stencil that considered the $(i-2,j,k)$, $(i-1,j,k)$, (i,j,k) and $(i+1,j,k)$ elements from t^n up to t^{n-2}. The 3D model utilised two spatialtemporal stencils from t^n to t^{n-1} and is best summarised spatially by Fig. 3.4 (see also the answer to research question 3):



(A) Momentum stencil **(B)** Continuity stencil

FIGURE 3.4 Visualisation of the neural networks' spatial stencils used for the half-channel defined by the $z = 0$ wall. Coordinate system origins denotes the (i,j,k)-element, and its orientation denotes the direction in which the element index positively increments in. (repeated from page 17)

   - **The outputs:** the 1D model predicts the closure terms for all shape functions simultaneously, while the 3D model uses eight MLPs to predict the closure terms of all shape functions. Each MLP predicts the four closure terms of a separate shape function.
   - **The objective function:** the 1D model was trained on an unconstrained problem minimising the L_2 error of the closure terms. The 3D model was trained on a constrained problem that attempts to minimise the L_2 error of the closure terms and the weak-residual of the system on the $\kappa = 0$ iteration, under energy penalties and solver adherence penalties.

- **The training data:** The 1D model was trained on data from 781 L_2 projected time steps ($\Delta t = 0.0064$), while the 3D model was trained on a total of 480 H_0^1 projected time steps ($\Delta t = 0.0072$). The time derivative used in the training data was different as well: the 1D model utilised a three-point first-order implicit approximation, while the 3D model utilised the backwards Euler method.

2) What coarse-scale variables are appropriate for the feature set?

   By observation of the two-scale Navier-Stokes equations Eqs. (3.1–3.2) and its residual-based variant Eqs. (3.3–3.4), appropriate features include the sum of all coarse-scale components, and its individual constituents. The time derivative was not included since the differences in the spatial features across the temporal stencil are indicative of temporal changes. The coarse-scale sum is the only term that includes the time march contribution as temporal effects are required to replicate it correctly. The dependency on the choice of spatiotemporal basis functions used is accepted.

   In addition to the coarse-scale features, the wall-normal distance (from any of the two walls of the channel) was used based on the success seen in the works of Gamahara and Hattori [17].

3) What spatiotemporal stencil is appropriate for the data-driven model to produce strong correlations between predictions and truth?

   Two spatiotemporal stencils were produced in this thesis by taking inspiration from two fine-scale Greens' functions. One stencil is associated with the momentum equations, while the other stencil is associated with the continuity equation. Based on the fine-scale Greens' function of the advection-diffusion problem given by Hughes and Sangalli [12], the momentum spatial stencil was chosen to include four upstream elements to the (i, j, k) element, and one downstream element. The continuity spatial stencil utilises the same elements, in addition to the direct spanwise neighbours, and an element away from the closest wall (relative to the (i, j, k) element). This was based on the fine-scale Greens' function of the Poisson problem, courtesy of Shrestha [15]. The temporal stencil extends from t^n to t^{n-1}, where the same spatial stencils are used across both time levels. No further temporal elements were chosen to avoid an excessively large stencil. The spatial component of the spatiotemporal stencils are visualised in Fig. 3.4.

4) What technique(s) can be used to make the model energy stable?

   This thesis utilises objective function constraints to influence the trained model to be energy stable. The constrained problem was made suitable for TensorFlow via the Augmented Lagrangian method, which transforms the problem into an unconstrained problem. The constraints used are established through the energy equation, which require that each element's predicted energy contribution is smaller than the truth. This is made possible by the additional property of inequalities.

5) How strongly can the data-driven model establish a functional relationship between the coarse-scale variables and the closure terms?

   In the context of network correlations, all eight data-driven models provide Pearson correlation coefficients between prediction and truth above 0.90 for all closure terms, implying a strongly correlated functional relationship between coarse-scale features and closure terms (see Fig. 4.2 and Fig .4.3). The eight networks used can be categorised into two 'supersets' of four networks each, with each set providing similar correlations to one another in the superset. The supersets are described by a strength adjective, denoting how strongly one superset's coarse-scale features correlate to the closure terms relative to the other superset. The adjective serves as a distinction between which superset is being discussed. The 'weak' superset provides correlational coefficients between ~0.92 to ~0.96 for the momentum closure terms, whereas the 'strong' superset provides correlational coefficients between ~0.97 to ~0.98 for the momentum closure terms. Both weak and strong supersets provide correlation coefficients of ~0.92 for the continuity closure term.

   When considering the network outputs as a function of channel height, the weak superset provides correlations coefficients above 0.88 for all closure terms (see Fig. 4.6a), while the strong superset provides coefficients above 0.86 for all closure terms (see Fig. 4.6b). The weak superset predicted average x-momentum closure terms that were between 8% and 22% larger than the truth (with exception at $z < 0.1$, which abruptly provides terms up to -2% larger than the truth), while the rest of the terms were between -5% to -10% larger than the truth. The strong superset predicted on average x-momentum closure terms that were between -8% to 6% larger than the truth (with exception at $z > 1.9$, which provides terms up to -14% larger than the truth), while the rest of the terms were between -2% to -12% larger than the truth.

   Although a priori correlations display that a strong functional has been established, an a posteriori simulation of the TCF test case demonstrates that this functional relationship / data-driven model is excessively energy dissipative. This was illustrated through the rapid reduction of the mean x-velocity when compared to the projected DNS solution after several time steps. The mean profiles of the y- and z-velocity are better represented, staying closer to the truth over the simulation period. In terms of fluctuation statistics, the functional produces smaller fluctuation magnitudes than the projected DNS fluctuations (see Figs.(5.2–5.4).

6.2 Recommendations

Given the limited scope of this thesis, there are several points of interest for future research, given below in a list:

- **Feature Set Reduction** – A total of 217 features are currently used for the neural networks, which poses a memory storage issue for larger problems in comparison to this thesis's test case. A parameter study aimed at removing features that marginally improve the

networks' closure term prediction capabilities both alleviate memory storage issues and aid in understanding the terms that drive the closure terms.

- **Spatiotemporal Stencil Sensitivity Analysis** – A sensitivity analysis into the spatiotemporal stencil selection allows for the reduction of the feature set size. Additionally, a better understanding on the locality of information which drives the closure terms is attainable.
- **Different Penalties** – Currently a mean-square error metric implemented through the Augmented Lagrangian method is used to measure the violation of the constraints when training the eight networks. The updates to the Augmented Langrangian variables are ad hoc and does not adhere to the standard algorithm associated with the method provided by Nocedal and Wright [20]. Alternative formulations of the penalties may better enforce energy conservation and solver adherence than the current method used.
- **Further Data Augmentation** – Currently the full data set only considers a simple case of data augmentation via the wall-normal distance feature. This is meant to prevent the choice of wall influencing the closure term predictions. Other forms of data augmentation can serve to teach the network of various properties of the flow. Of particular interest is teaching the solver of prediction invariance under z-axis rotations – the current network is unable to handle a flow in which the coordinate frames are unaligned with the x-axis being the streamwise direction, y-axis being the spawnwise direction, and z-axis being the wall-normal direction.
- **Superset Convergence** – It is hypothesised that the aforementioned data augmentation will merge the superset correlations together (and therefore all sets within the two supersets). This would allow a single network to predict the closure terms associated with any shape function as the range of outputs that the network network needs to account for will be invariant to the choice of (linear) shape functions. This can be seen as a consequence of teaching rotational invariance to the network. Rotational invariance manifests in the requirement of each network needing to predict what all other networks are able to predict, to prevent association based on shape functions (which skew the integrating quantity in a given direction). It may be possible that a second network will be required due to differing flow statistics along the wall-normal direction, thus the two networks would predict the closure terms associated with the shape functions at a different z-level in an element.
- **Question of Closure Term Uniqueness** – The true closure terms are realisation of a single fine-scale velocity and pressure field, regardless of the shape function considered. Predictions that differ from the truth are realisations of different fine-scale fields, and imply that they are different for each shape function. A method to enforce closure term predictions coming from the same fine-scale fields may improve a posteriori simulations.
- **Physics-Informed Neural Networks** – Currently the network is informed of the solver physics by regressing for the element-local weak residual, which relates the t^{n+1} closure term and t^n coarse-scale sum for each of the shape functions. The solver adherence penalty was used to explicitly relate the closure term prediction and weak residual prediction. It may be possible to improve the network's knowledge on the Navier-Stokes equations by considering a loss function measuring the violation of the weak-form two-scale Navier-Stokes equations at t^{n+1}. This approach would be of interest when predicting the closure terms across multiple elements simultaneously as the weak-form PDEs are satisfied across the weighting function and not across the shape function.

- **Local Non-Dimensionalisation** – Network generalisation is possible by training across a larger range of flow conditions, meaning that a larger training data set is required. The cost associated with this training can become prohibitive even with better hardware used than in this thesis. Circumvention of this problem may be achieved by data set non-dimensionalisation, allowing for outputs to be dimensionalised according to the flow condition currently experienced after the prediction process.
- **Convolutional Neural Networks** – The spatial component of the spatiotemporal stencil could be reproduced through the use of a convolutional model in the network. The input can be a local region to the (i, j, k)-element instead of the entire flow field. This may be important when considering rotational invariance: the current spatial stencil is unsuitable as it is skewed in the x-direction. A y-dominant flow requires a spatial stencil that will be skewed in the y-direction, and similarly for flows with any heading in the plane defined by the z-axis unit vector.
- **Recurrent Neural Networks** – The temporal component of the spatiotemporal stencil could be reproduced through the use of a recurrent model in the network. Coupling a recurrent model to the input layer would be representative of the inputs time history, whereas coupling to the output layer would be representative of the outputs time history. Both are relevant as there are clear temporal correlations between inputs and outputs (see App. A), and strong autocorrelations (e.g. Fig. A.223).

Works Cited

[1] Thomas J. R. Hughes, Assad A. Oberai, and Luca Mazzei. "Large eddy simulation of turbulent channel flows by the variational multiscale method." In: *Physics of Fluids* 13.6 (June 2001), pp. 1784–1799. DOI: 10.1063/1.1367868.

[2] Bernard J Geurts and Darryl D Holm. "Commutator errors in large-eddy simulation." In: *Journal of Physics A: Mathematical and General* 39.9 (Feb. 2006), pp. 2213–2229. DOI: 10.1088/0305-4470/39/9/015.

[3] Martin Janssens. "Machine Learning of Atmospheric Turbulence in a Variational Multiscale Model." MA thesis. Delft University of Technology, Aug. 2019.

[4] Michel Robijns. "A Machine Learning Approach to Unresolved Scale Modeling for Burgers' Equation." MA thesis. Delft University of Technology, Aug. 2019.

[5] M. Janssens and S. J. Hulshoff. "Advancing Artificial Neural Network Parameterization for Atmospheric Turbulence Using a Variational Multiscale Model." In: *Journal of Advances in Modeling Earth Systems* 14.1 (Jan. 2022). DOI: 10.1029/2021ms002490.

[6] Abhinand Pusuluri. "Noise-augmented offline training of ANN unresolved scale models." MA thesis. Delft University of Technology, Jan. 2020.

[7] Sharath Rajampeta. "Stabilization Strategies for a Variational Multiscale Method Coupled with an Artificial Neural Network." MA thesis. Delft University of Technology, Oct. 2021.

[8] Mohith Kanala. "Variational Multiscale Method coupled with an Artificial Neural Network for 3D turbulent channel flow." MA thesis. Delft University of Technology, Apr. 2023.

[9] Zhideng Zhou et al. "Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network." In: *Computers & Fluids* 195 (Dec. 2019), p. 104319. DOI: 10.1016/j.compfluid.2019.104319.

[10] Karthik Duraisamy. "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence." In: *Physical Review Fluids* 6.5 (May 2021). DOI: 10.1103/physrevfluids.6.050504.

[11] Andrea Beck and Marius Kurz. "A perspective on machine learning methods in turbulence modeling." In: *GAMM-Mitteilungen* 44.1 (Mar. 2021). DOI: 10.1002/gamm.202100002.

[12] T. J. R. Hughes and G. Sangalli. "Variational Multiscale Analysis: the Fine-scale Green's Function, Projection, Optimization, Localization, and Stabilized Methods." In: *SIAM Journal on Numerical Analysis* 45.2 (Jan. 2007), pp. 539–557. DOI: 10.1137/050645646.

[13] Thomas J. R. Hughes, Guglielmo Scovazzi, and Leopoldo P. Franca. *Multiscale and Stabilized Methods*. Online Encyclopedia Chapter. Nov. 2017. DOI: 10.1002/9781119176817.ecm2051.

[14] Y. Bazilevs et al. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows." In: *Computer Methods in Applied Mechanics and Engineering* 197.1-4 (Dec. 2007), pp. 173–201. DOI: 10.1016/j.cma.2007.07.016.

[15] Suyash Shrestha. "Hybridised Mimetic Discretisation and Variational Multiscale Theory for Advection-Dominated Problems." MA thesis. Delft University of Technology, Nov. 2022.

[16] Isabelle Guyon and André Elisseef. "An Introduction to Variable and Feature Selection." In: *Journal of Machine Learning* 3 (2003), pp. 1157–1182. URL: https://ei.is.mpg.de/publications/2685.

[17] Masataka Gamahara and Yuji Hattori. "Searching for turbulence models by artificial neural network." In: *Physical Review Fluids* 2.5 (May 2017). DOI: 10.1103/physrevfluids.2.054604.

[18] Andrea Beck, David Flad, and Claus-Dieter Munz. "Deep neural networks for data-driven LES closure models." In: *Journal of Computational Physics* 398 (Dec. 2019), p. 108910. DOI: 10.1016/j.jcp.2019.108910.

[19] Marius Kurz and Andrea Beck. "A machine learning framework for LES closure terms." In: *ETNA - Electronic Transactions on Numerical Analysis* 56 (2022), pp. 117–137. DOI: 10.1553/etna_vol56s117.

[20] Jorge Nocedal and Stephen J. Wright. "Numerical Optimization." en. In: 2nd ed. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer, July 2006. Chap. 17.

[21] Leslie N. Smith. "Cyclical Learning Rates for Training Neural Networks." In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2017. DOI: 10.1109/wacv.2017.58. URL: https://doi.org/10.1109/wacv.2017.58.

[22] Charles A. Taylor, Thomas J.R. Hughes, and Christopher K. Zarins. "Finite element modeling of blood flow in arteries." In: *Computer Methods in Applied Mechanics and Engineering* 158.1-2 (May 1998), pp. 155–196. DOI: 10.1016/s0045-7825(98)80008-x.

Appendices: Energy-Conservative Data-Driven Modelling for the two-scale Navier-Stokes Equations

Master of Science Thesis

by

Andrea Bettini

to obtain the degree of Master of Science
in Aerospace Engineering at Delft University of Technology,
to be defended publicly on August 29th, 2023 at 10:00

Student number:	4656970	
Project duration:	January 9, 2023 – August 29, 2023	
Thesis committee:	Dr. S. Hulshoff,	TU Delft, Responsible Supervisor
	Dr. R. P. Dwight,	TU Delft, Chair
	Dr. B. Chen,	TU Delft, External Examiner

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Appendix A H_0^1 TCF Dataset Histograms

A.1 t^{n+1} - t^{n+1} Histograms 63
A.1.1 (i, j, k) Histograms 63
A.1.2 $(i-1, j, k)$ Histograms 72
A.2 t^{n} - t^{n+1} Histograms 81
A.2.1 (i, j, k) Histograms 81
A.2.2 $(i-1, j, k)$ Histograms 91
A.2.3 $(i-2, j, k)$ Histograms 101
A.2.4 $(i-3, j, k)$ Histograms 111
A.2.5 $(i-4, j, k)$ Histograms 121
A.2.6 $(i+1, j, k)$ Histograms 131
A.2.7 $(i, j-1, k)$ Histograms 141
A.2.8 $(i, j+1, k)$ Histograms 151
A.2.9 $(i, j, k \pm 1)$ Histograms 161
A.3 t^{n-1} - t^{n+1} Histograms 171
A.3.1 (i, j, k) Histograms 171
A.3.2 $(i-1, j, k)$ Histograms 181
A.3.3 $(i-2, j, k)$ Histograms 191
A.3.4 $(i-3, j, k)$ Histograms 201
A.3.5 $(i-4, j, k)$ Histograms 211
A.3.6 $(i+1, j, k)$ Histograms 221
A.3.7 $(i, j-1, k)$ Histograms 231
A.3.8 $(i, j+1, k)$ Histograms 241
A.3.9 $(i, j, k \pm 1)$ Histograms 251

A.1 t^{n+1} - t^{n+1} Histograms

A.1.1 (i, j, k) Histograms

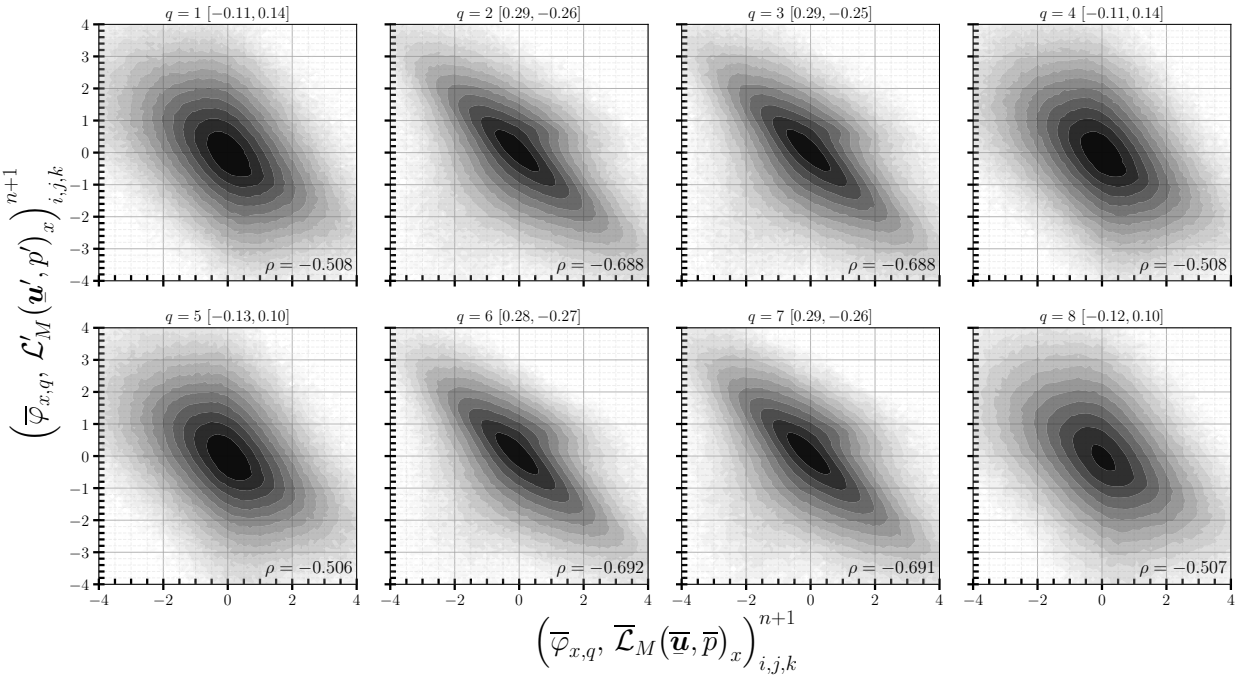


FIGURE A.1 Histograms of the t^{n+1} (i, j, k)-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

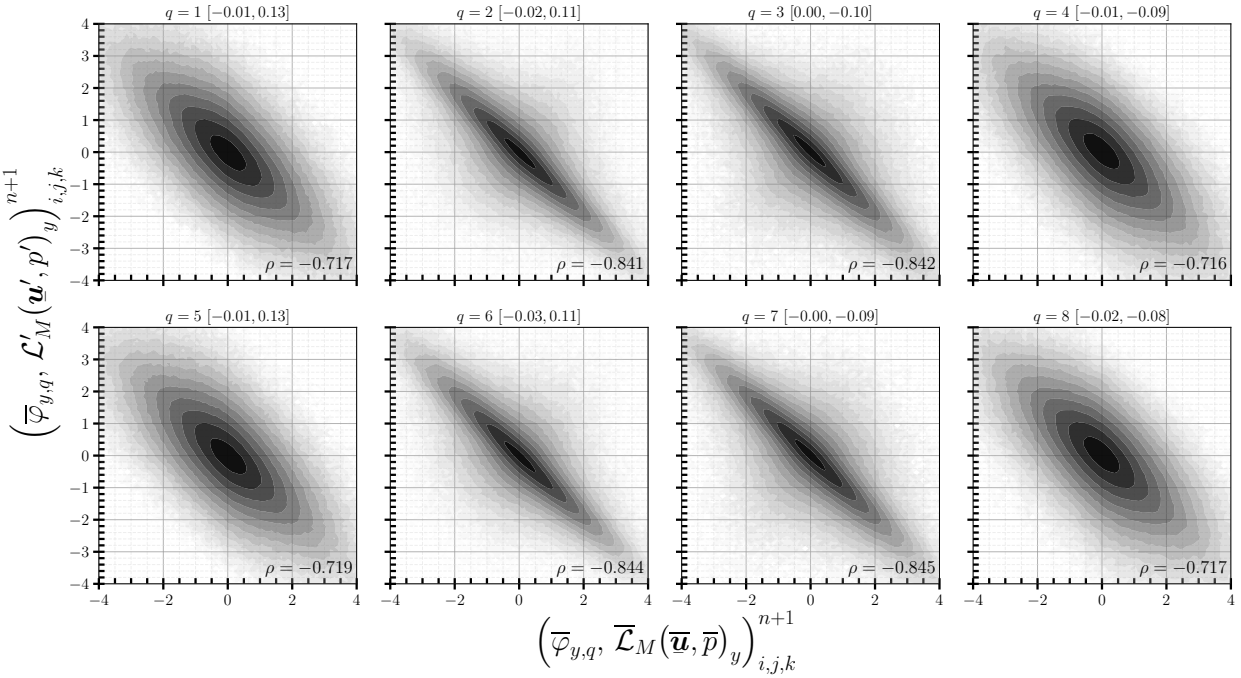


FIGURE A.2 Histograms of the t^{n+1} (i, j, k)-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

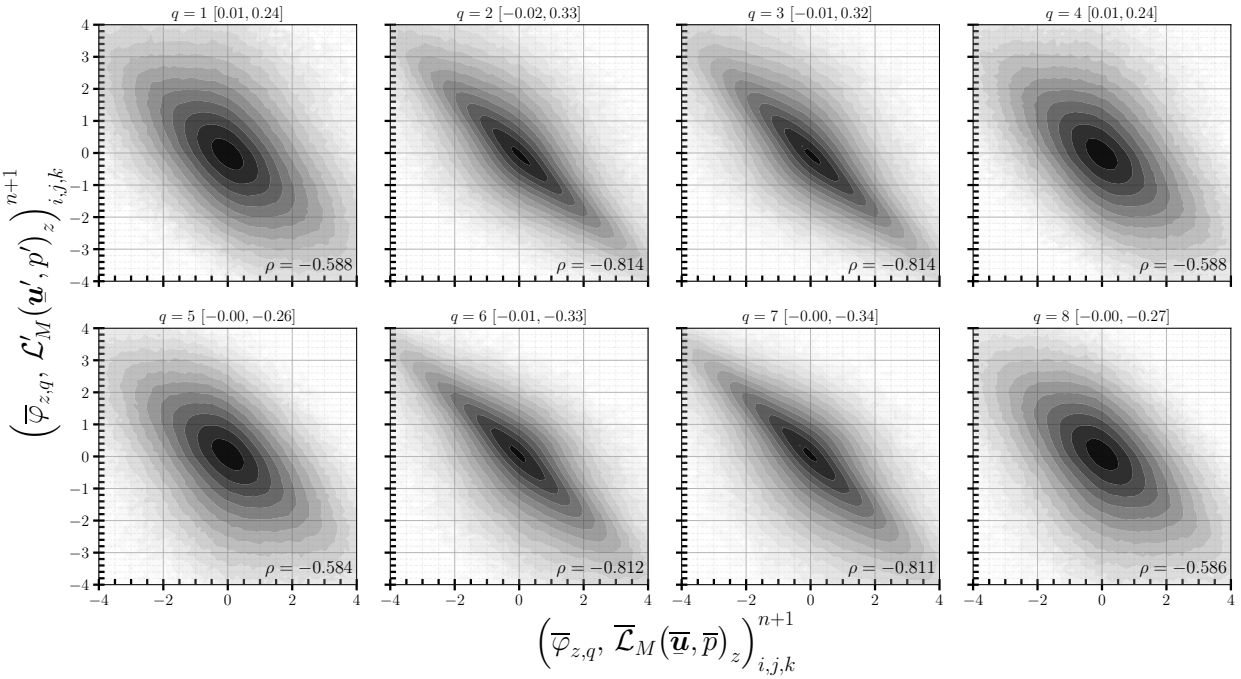


FIGURE A.3 Histograms of the t^{n+1} (i, j, k)-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

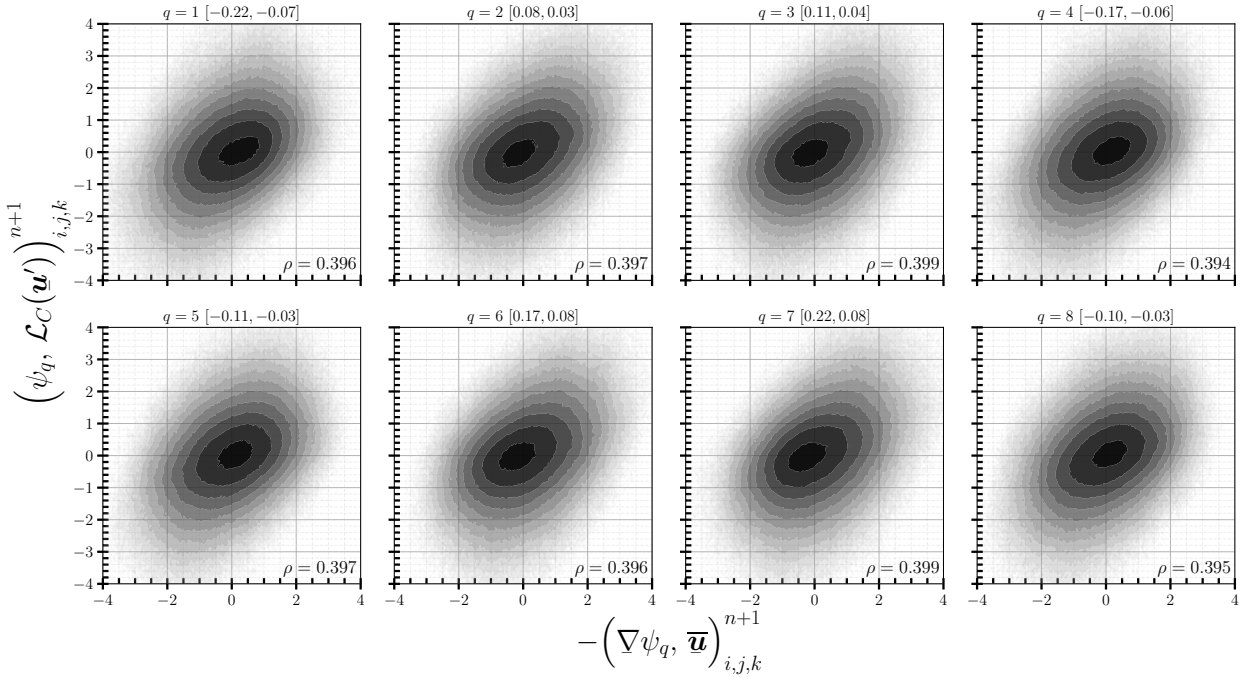


FIGURE A.4 Histograms of the t^{n+1} (i, j, k)-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

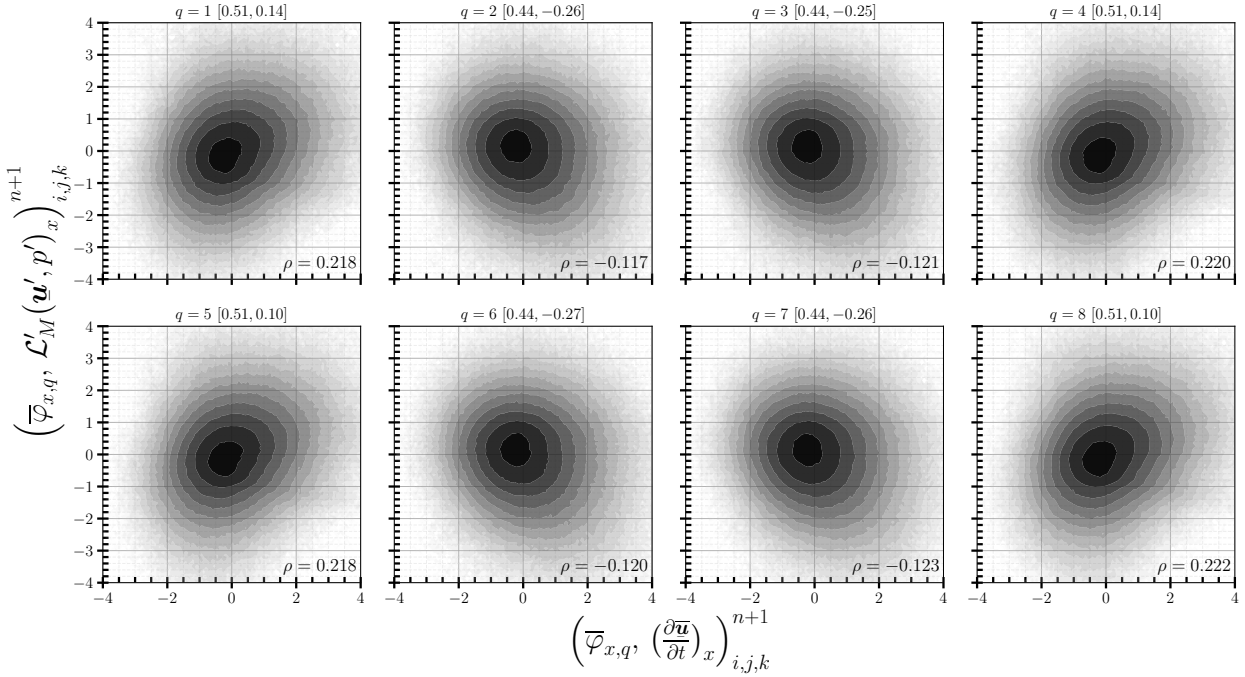


FIGURE A.5 Histograms of the t^{n+1} (i, j, k)-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

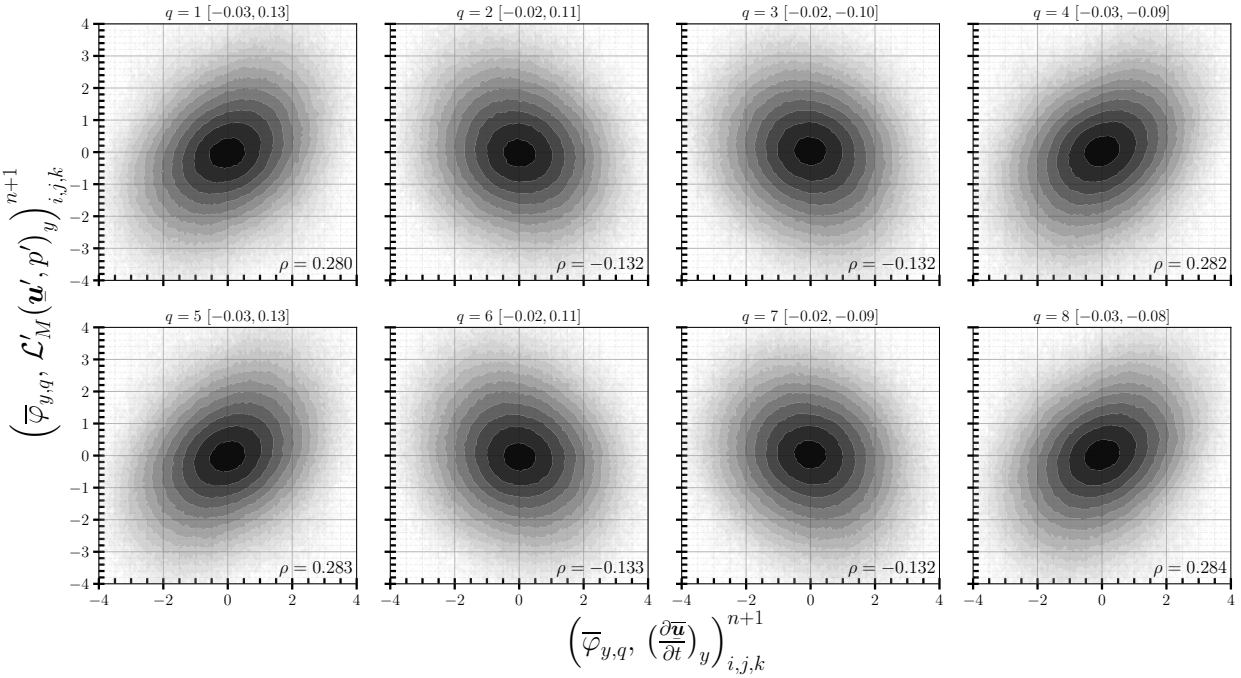


FIGURE A.6 Histograms of the t^{n+1} (i, j, k)-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

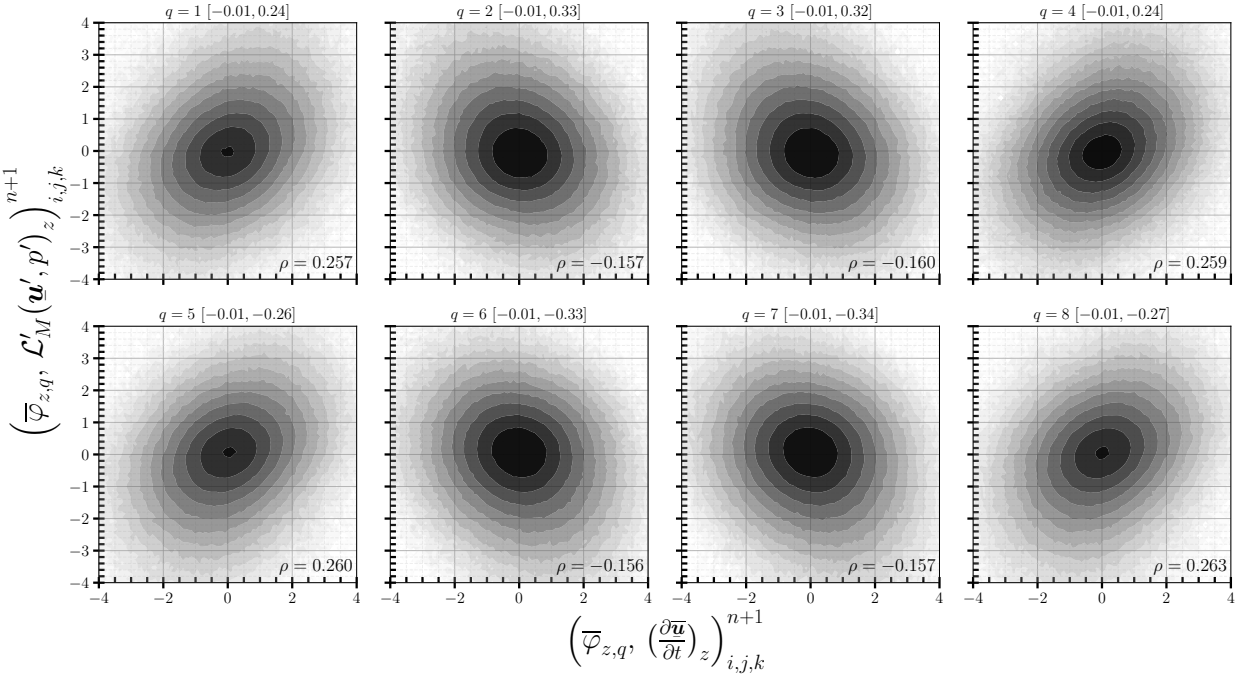


FIGURE A.7 Histograms of the t^{n+1} (i, j, k)-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

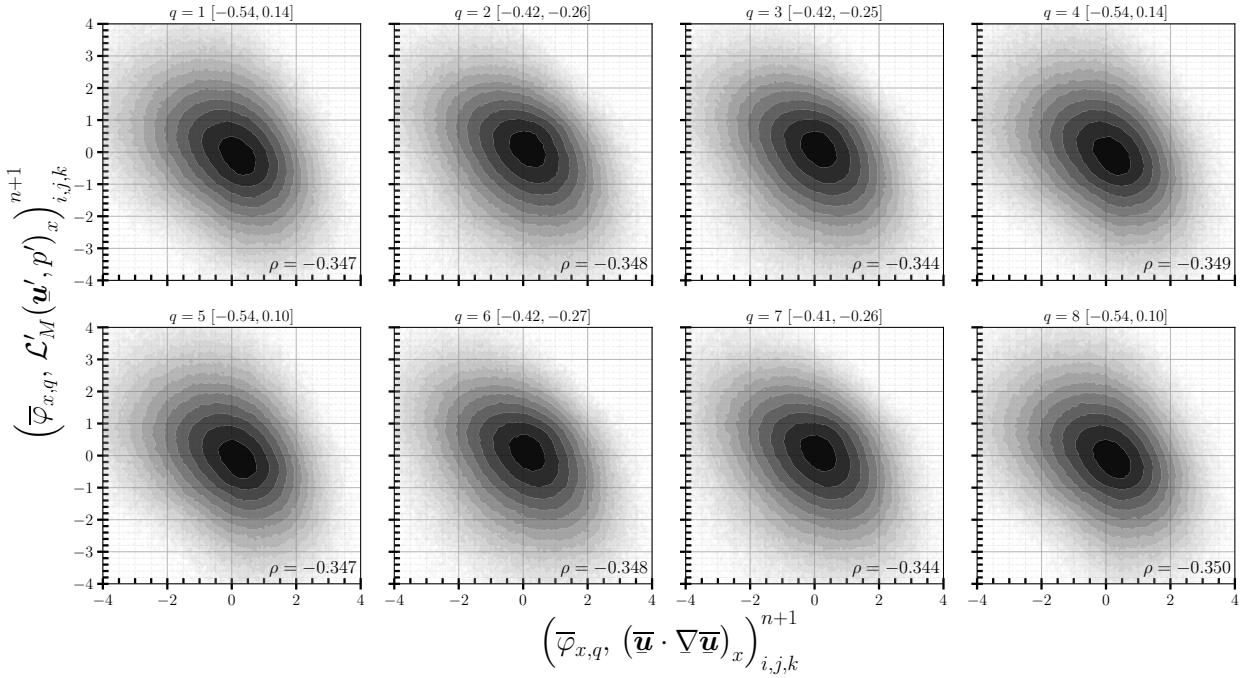


FIGURE A.8 Histograms of the t^{n+1} (i, j, k)-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

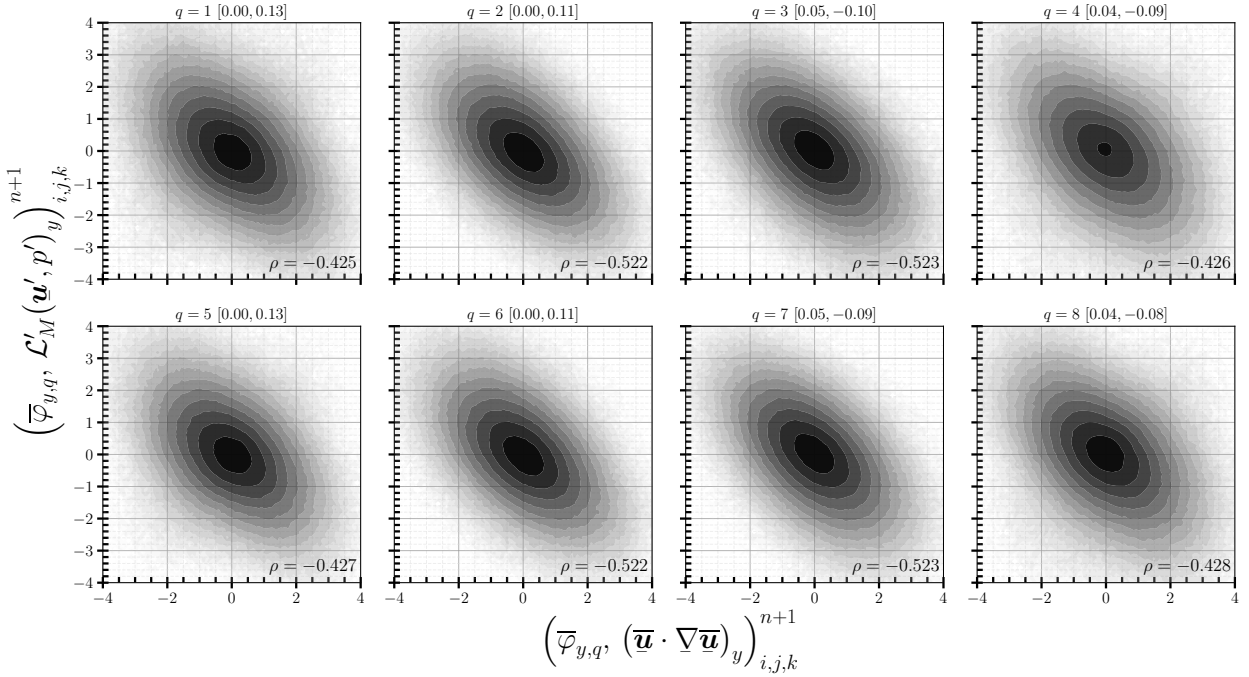


FIGURE A.9 Histograms of the t^{n+1} (i, j, k)-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

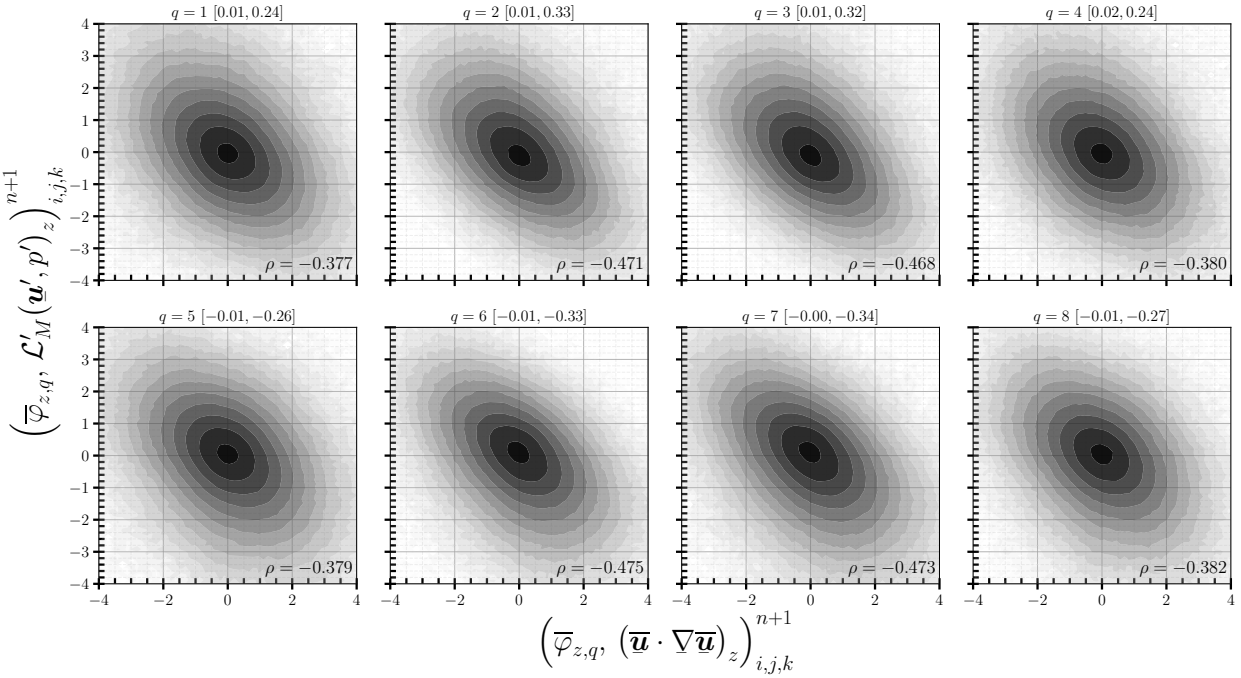


FIGURE A.10 Histograms of the t^{n+1} (i, j, k)-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

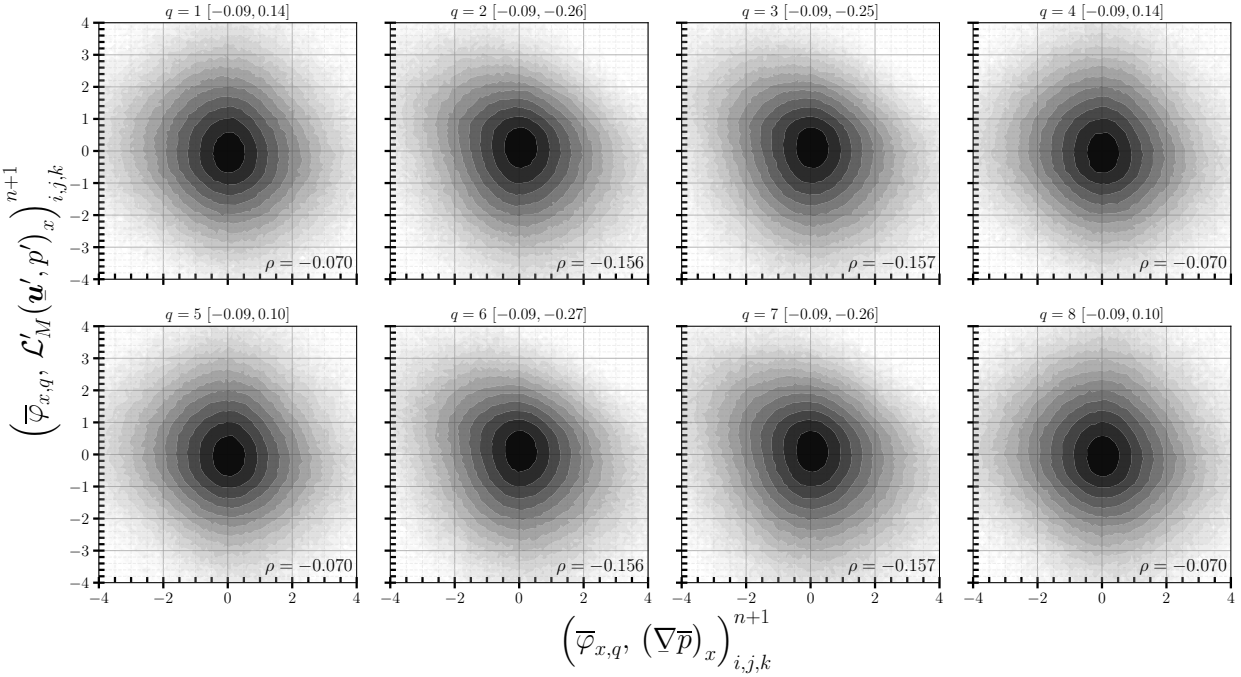


FIGURE A.11 Histograms of the t^{n+1} (i, j, k)-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

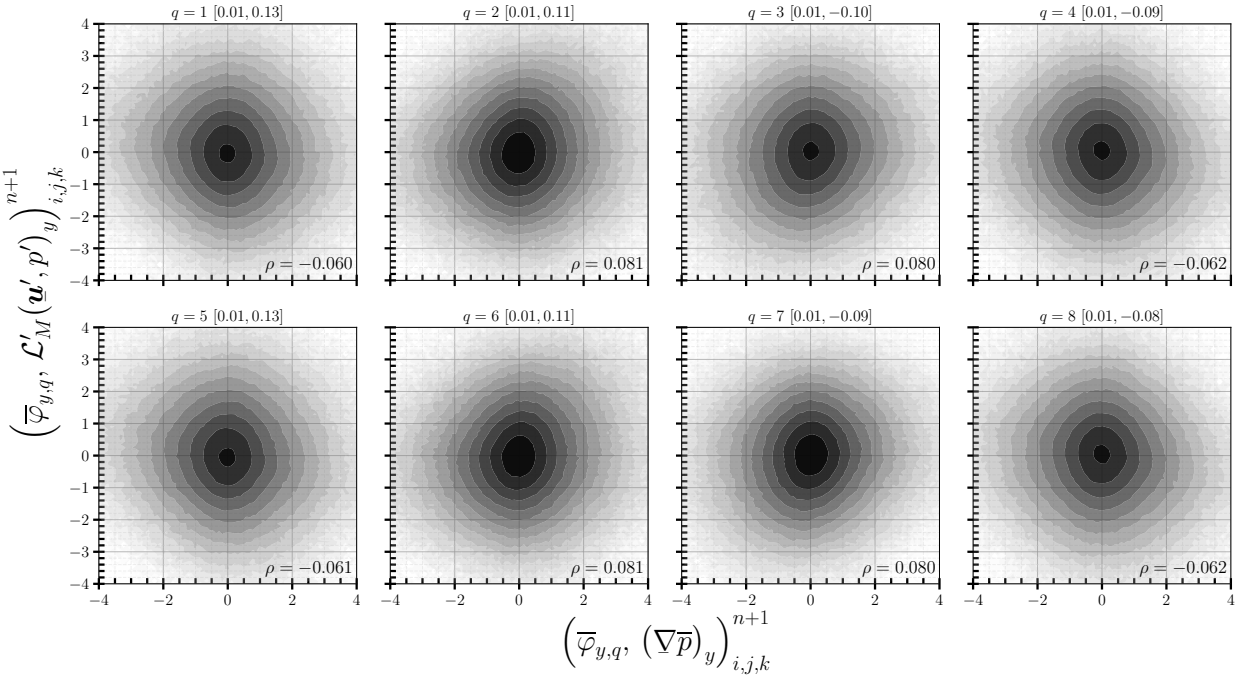


FIGURE A.12 Histograms of the t^{n+1} (i, j, k)-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

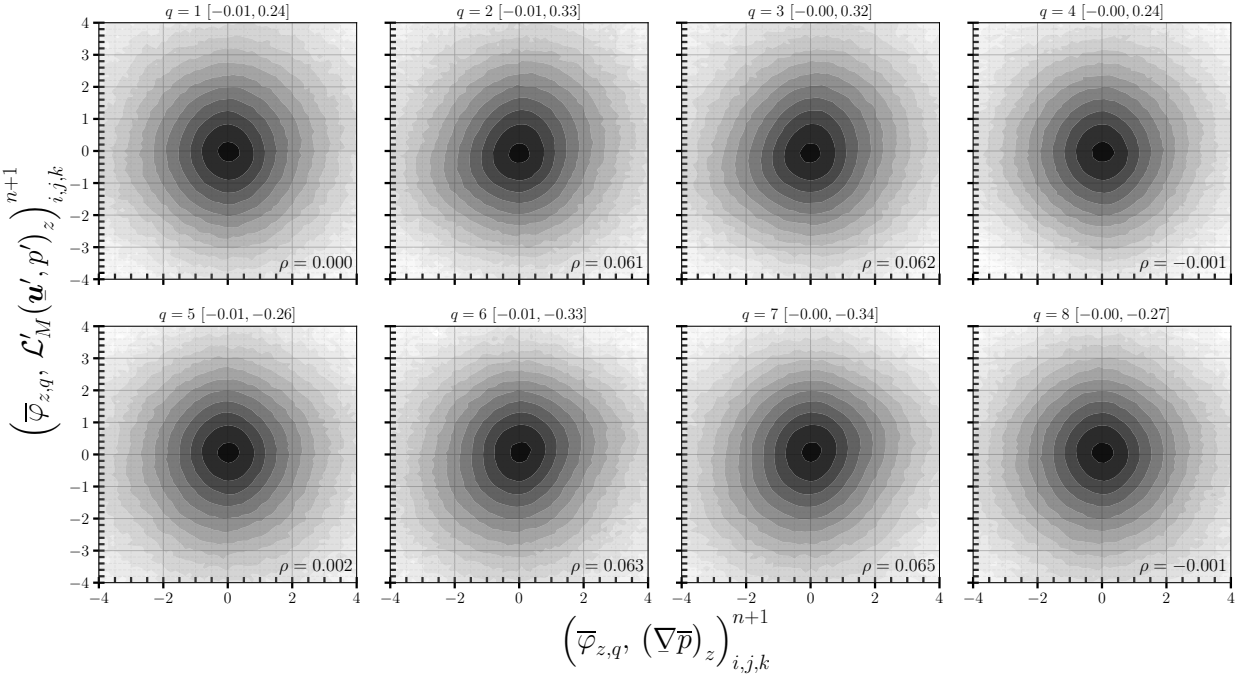


FIGURE A.13 Histograms of the t^{n+1} (i, j, k)-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

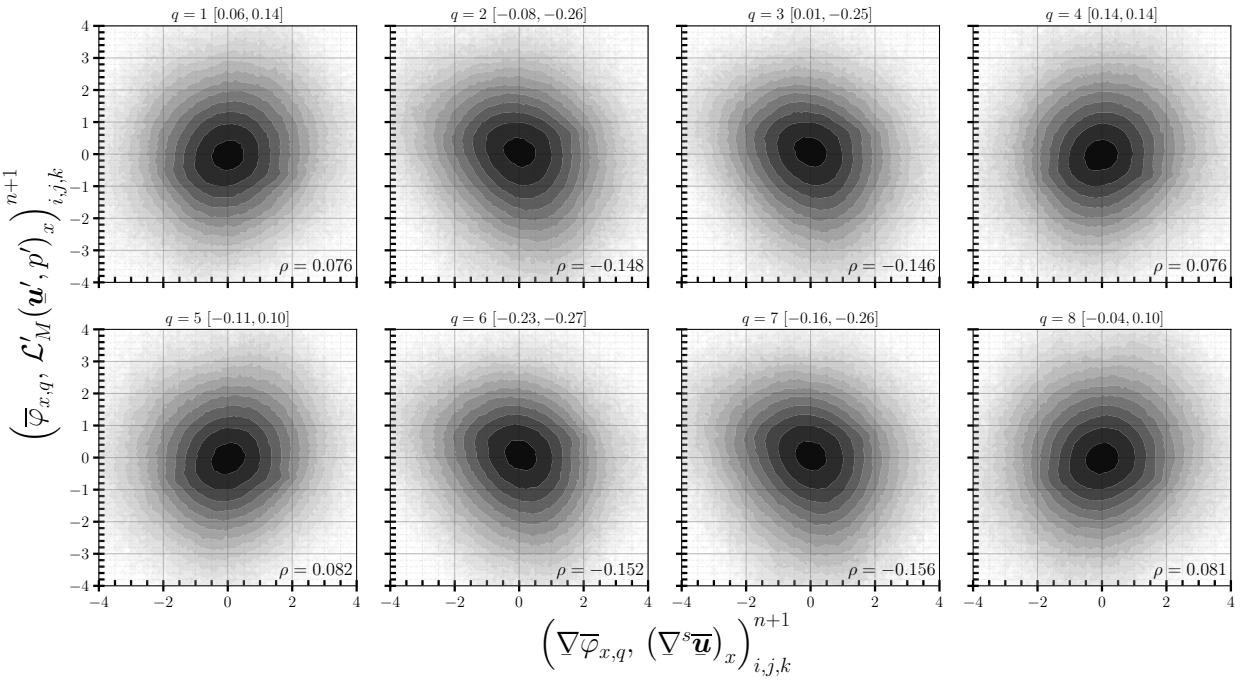


FIGURE A.14 Histograms of the t^{n+1} (i, j, k)-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

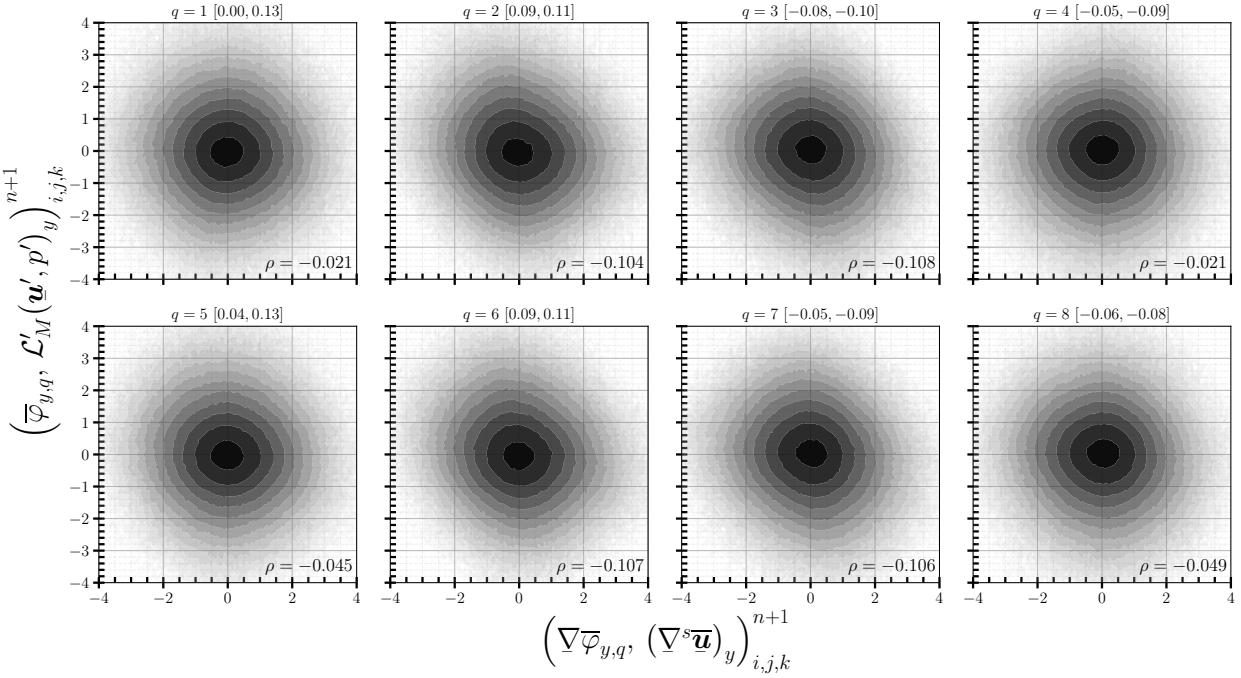


FIGURE A.15 Histograms of the t^{n+1} (i, j, k)-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

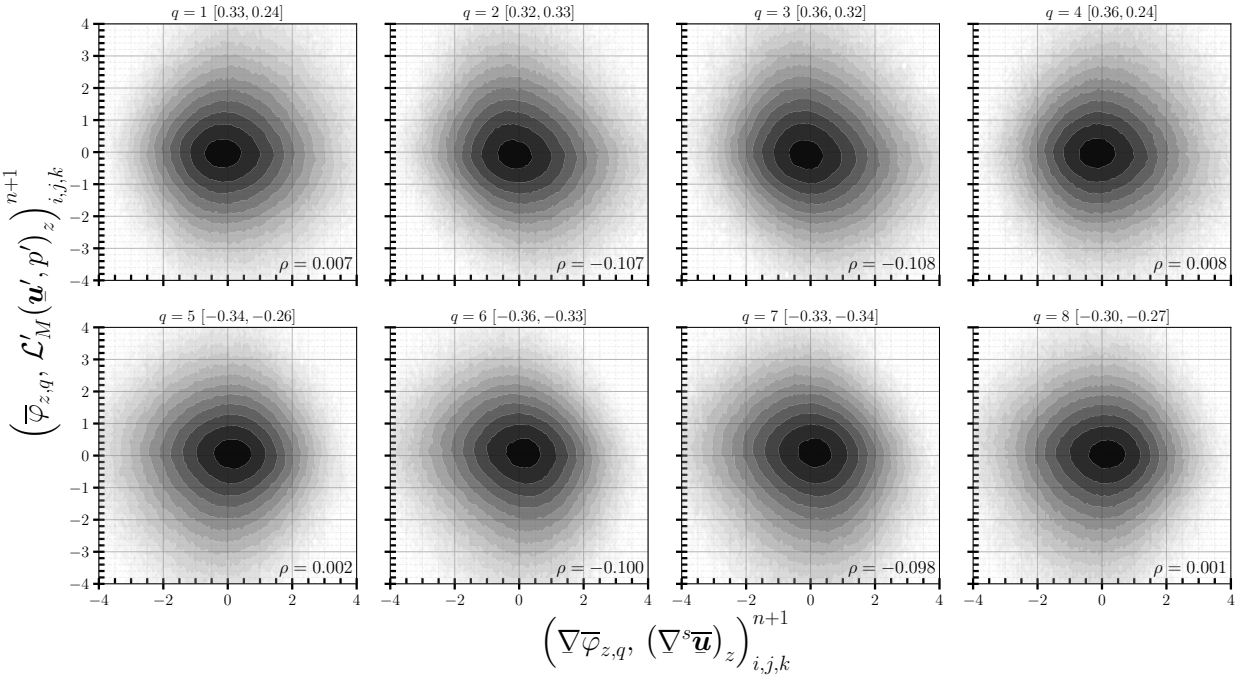


FIGURE A.16 Histograms of the t^{n+1} (i, j, k)-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.1.2 $(i-1, j, k)$ Histograms

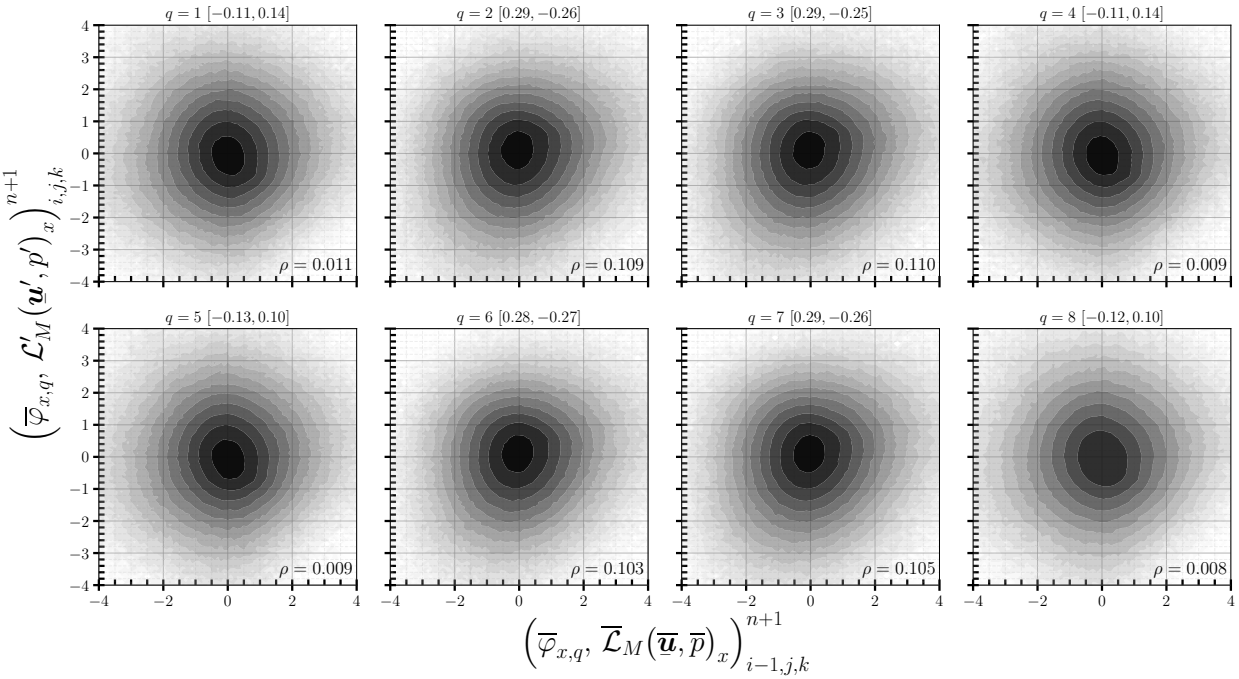


Figure A.17 Histograms of the t^{n+1} $(i-1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

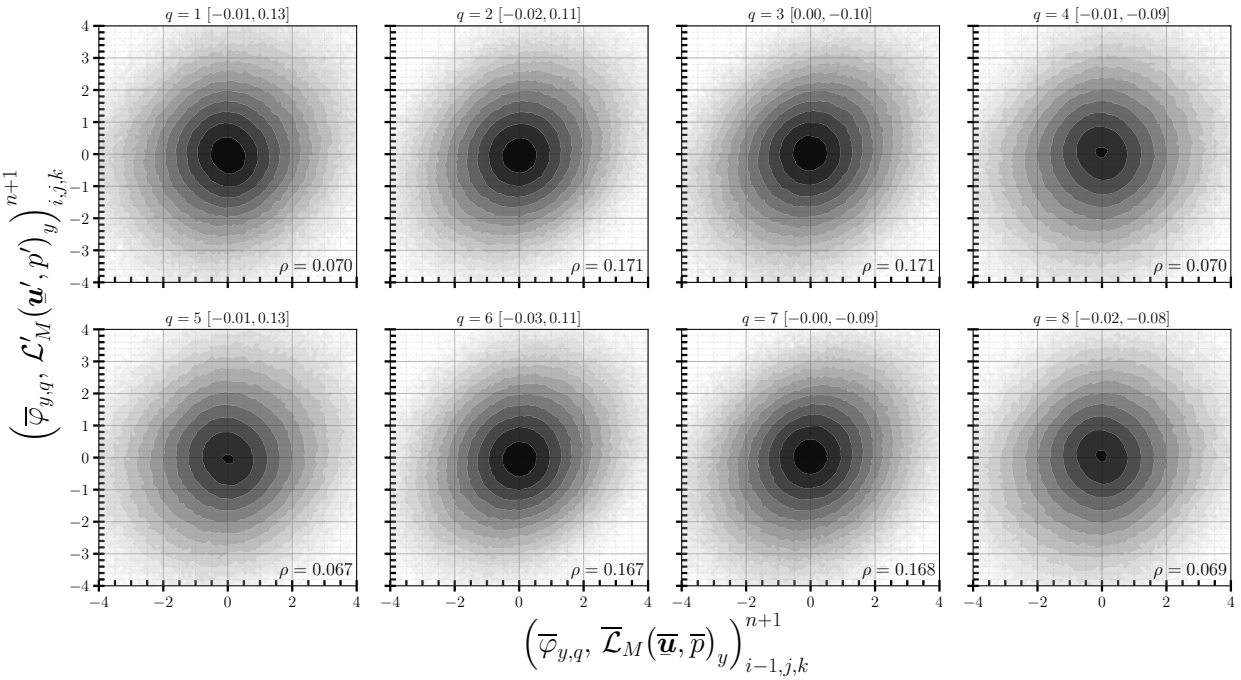


FIGURE A.18 Histograms of the t^{n+1} $(i-1, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

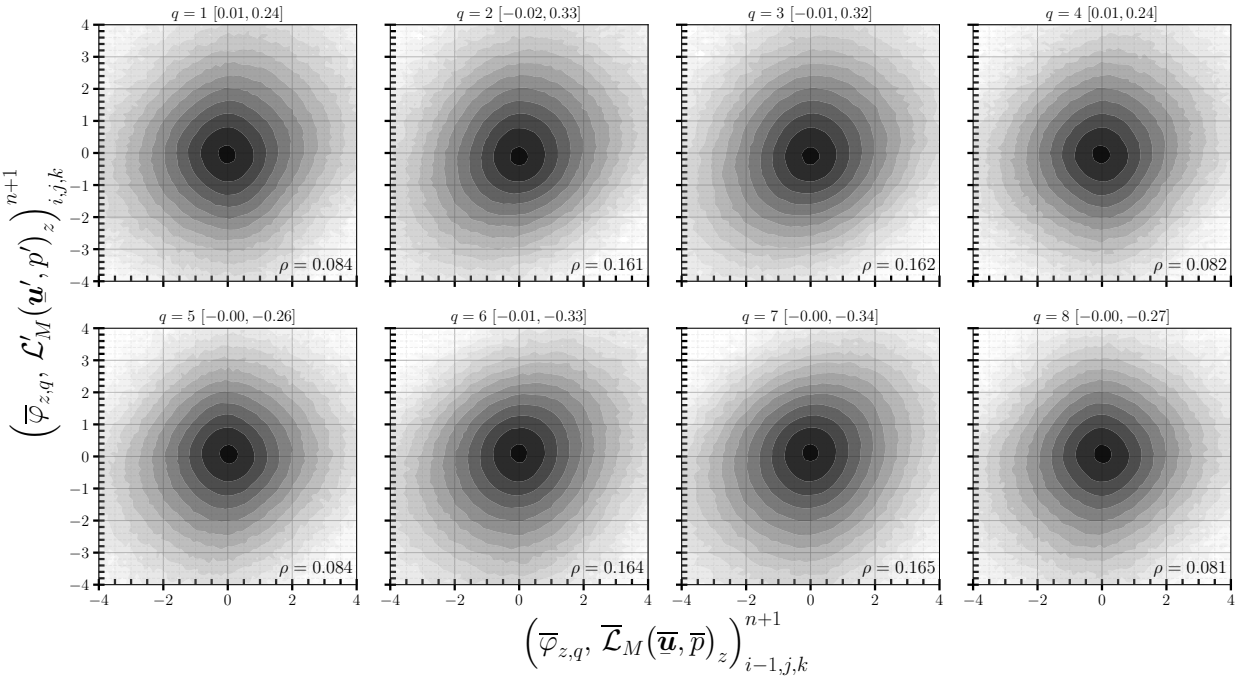


FIGURE A.19 Histograms of the t^{n+1} $(i-1, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

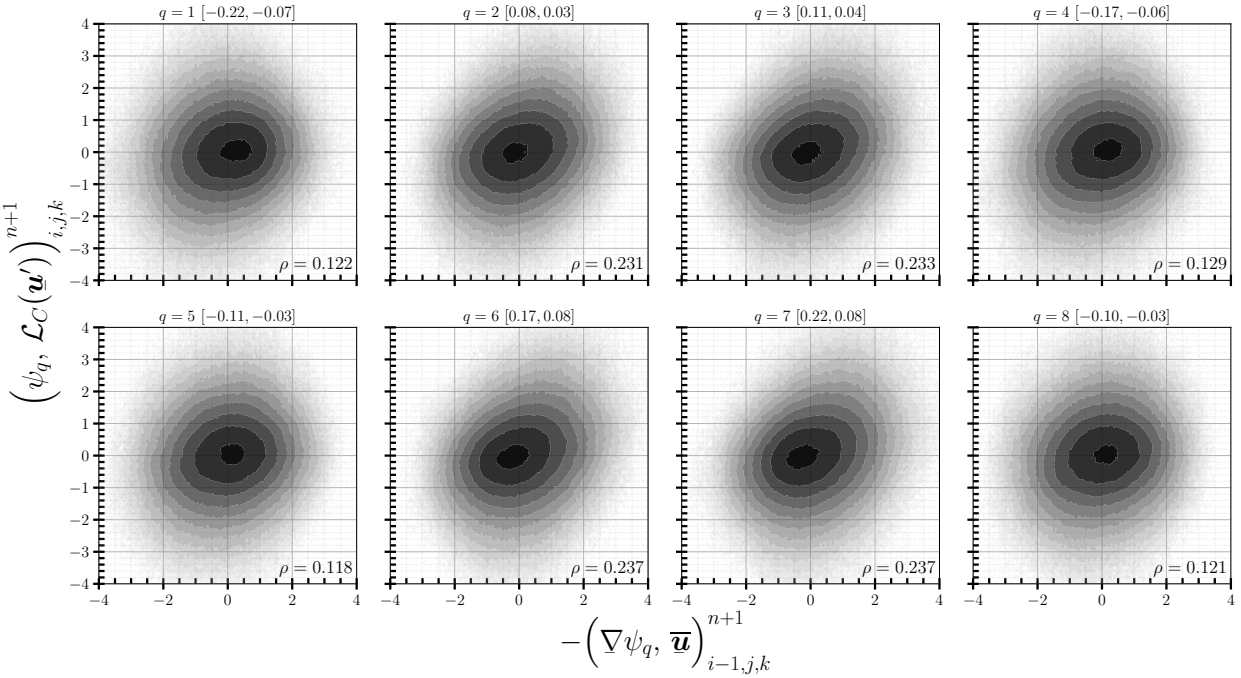


FIGURE A.20 Histograms of the t^{n+1} $(i-1, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

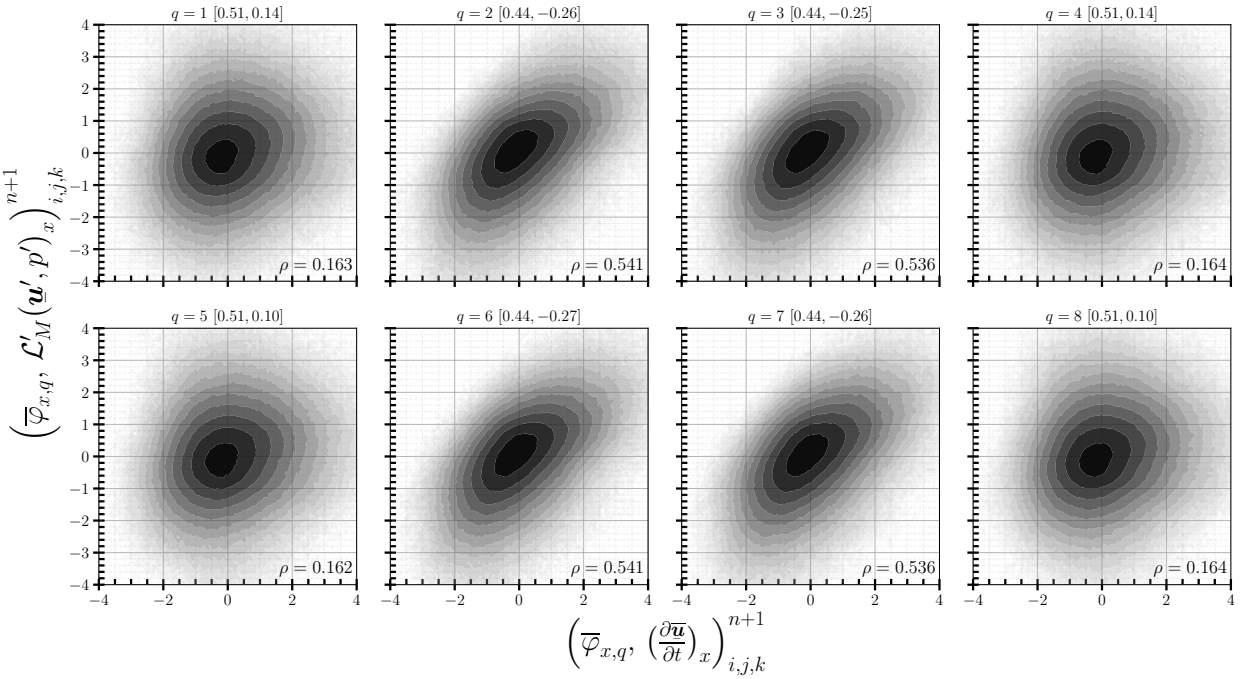


FIGURE A.21 Histograms of the t^{n+1} $(i-1, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

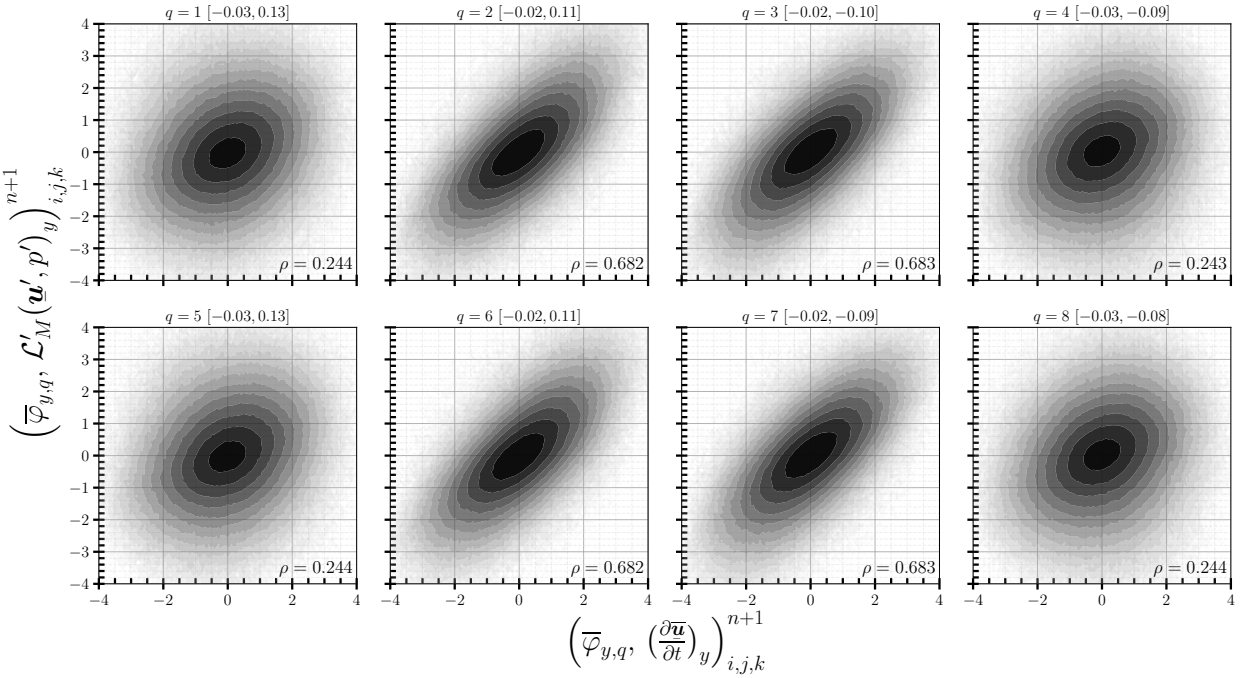


FIGURE A.22 Histograms of the t^{n+1} $(i-1, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

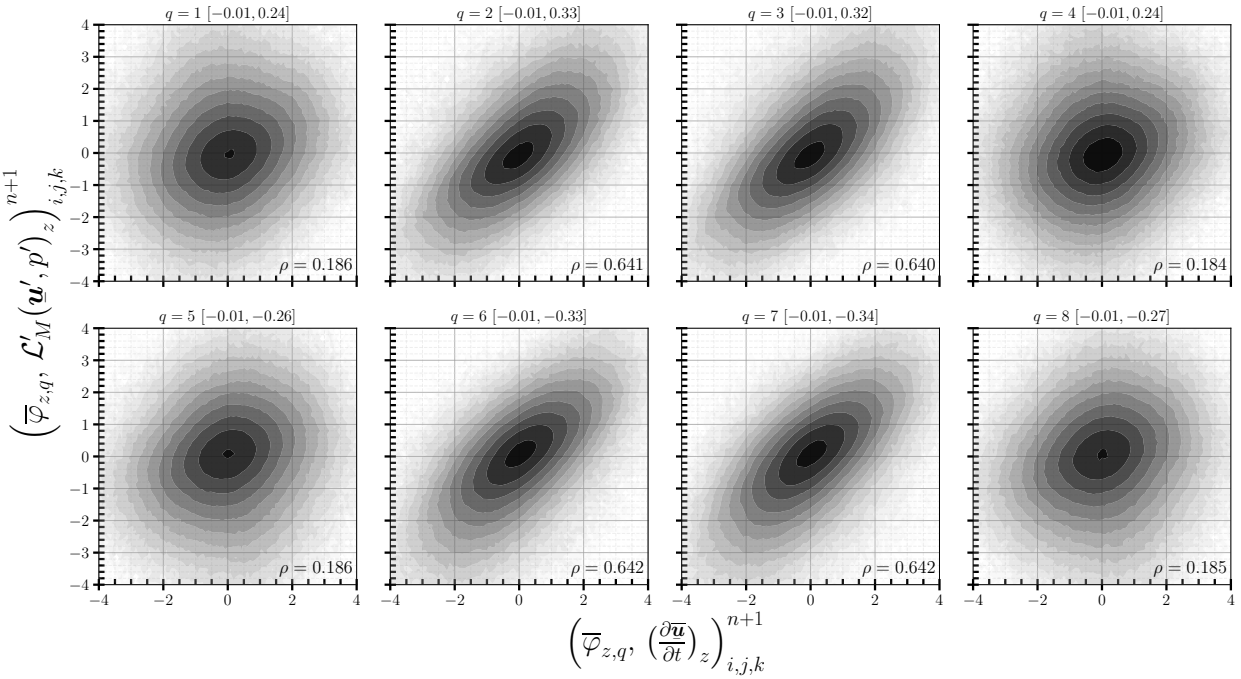


FIGURE A.23 Histograms of the t^{n+1} $(i-1, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

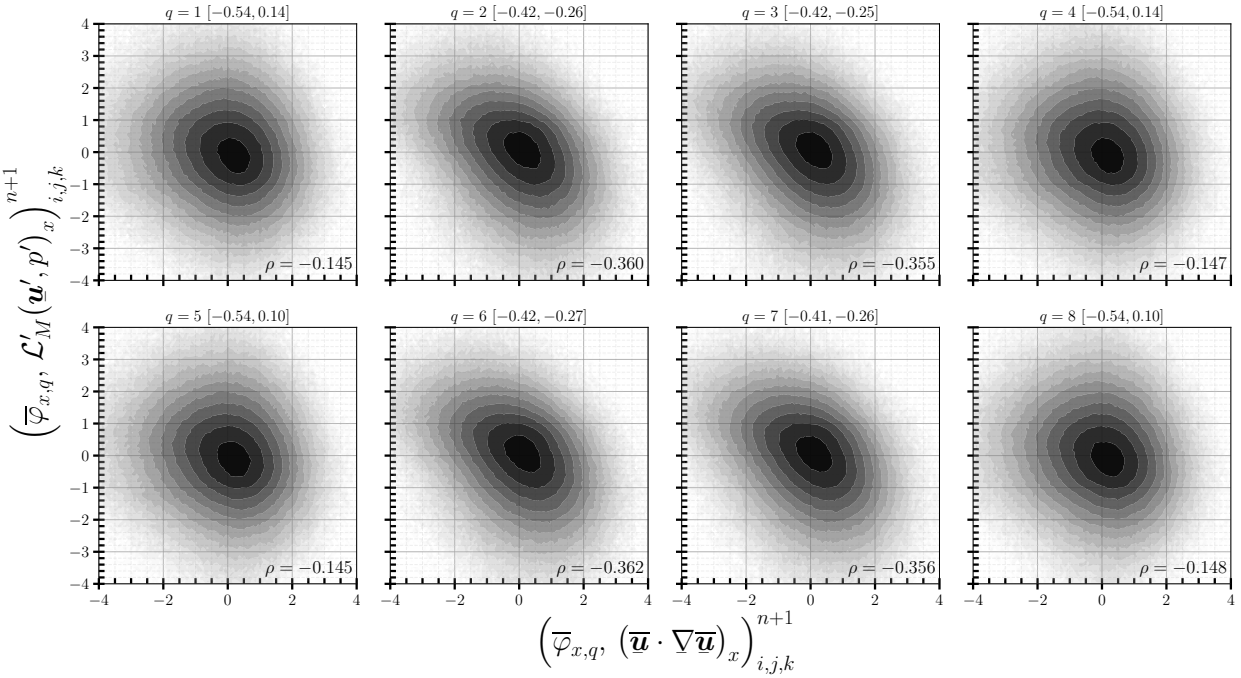


FIGURE A.24 Histograms of the t^{n+1} $(i-1, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

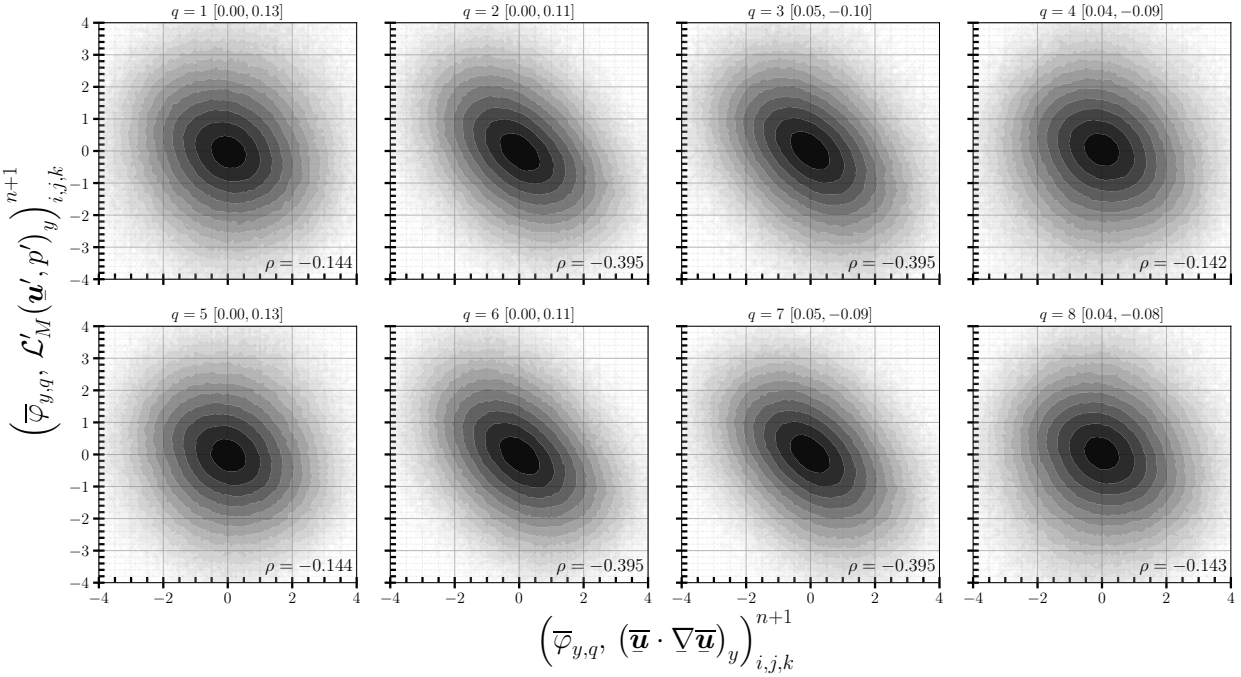


FIGURE A.25 Histograms of the t^{n+1} $(i-1, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

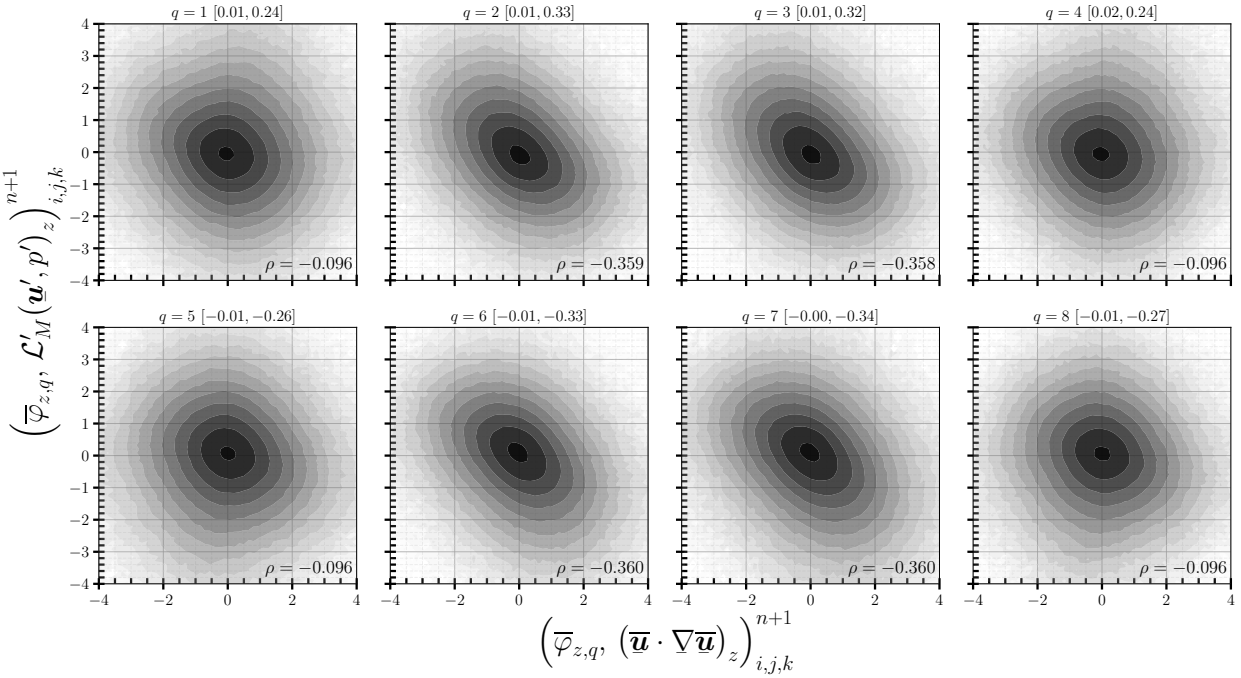


FIGURE A.26 Histograms of the t^{n+1} $(i-1, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

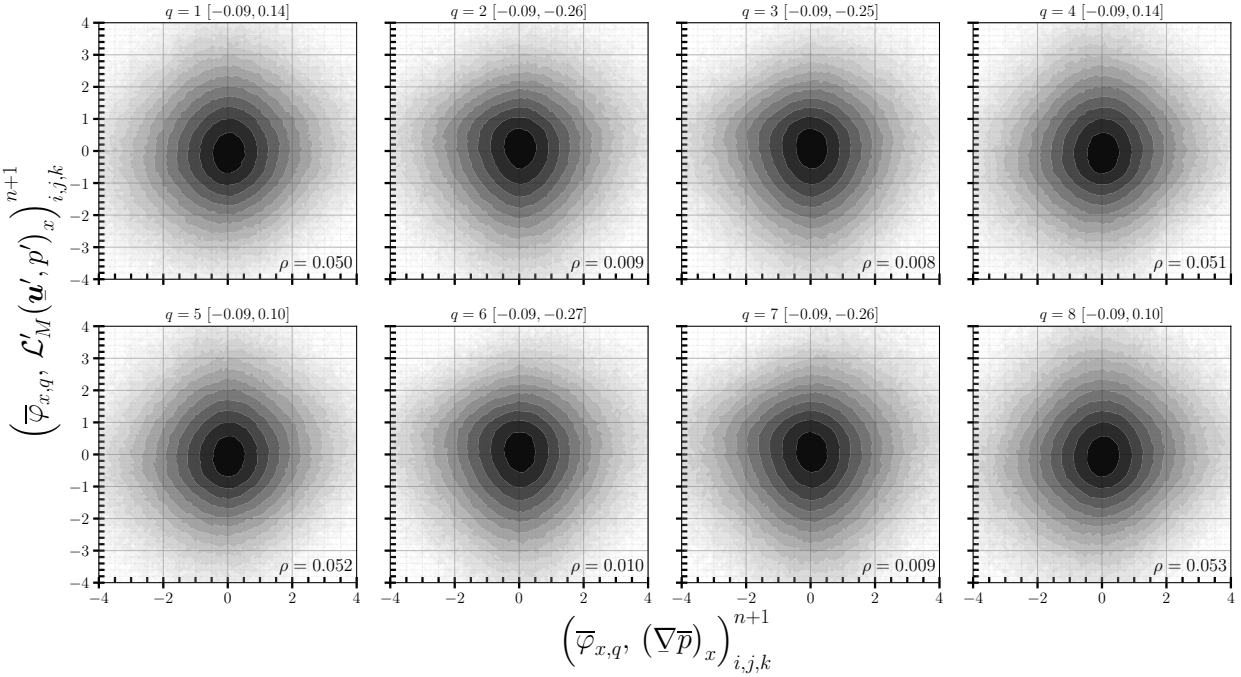


FIGURE A.27 Histograms of the t^{n+1} $(i-1, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

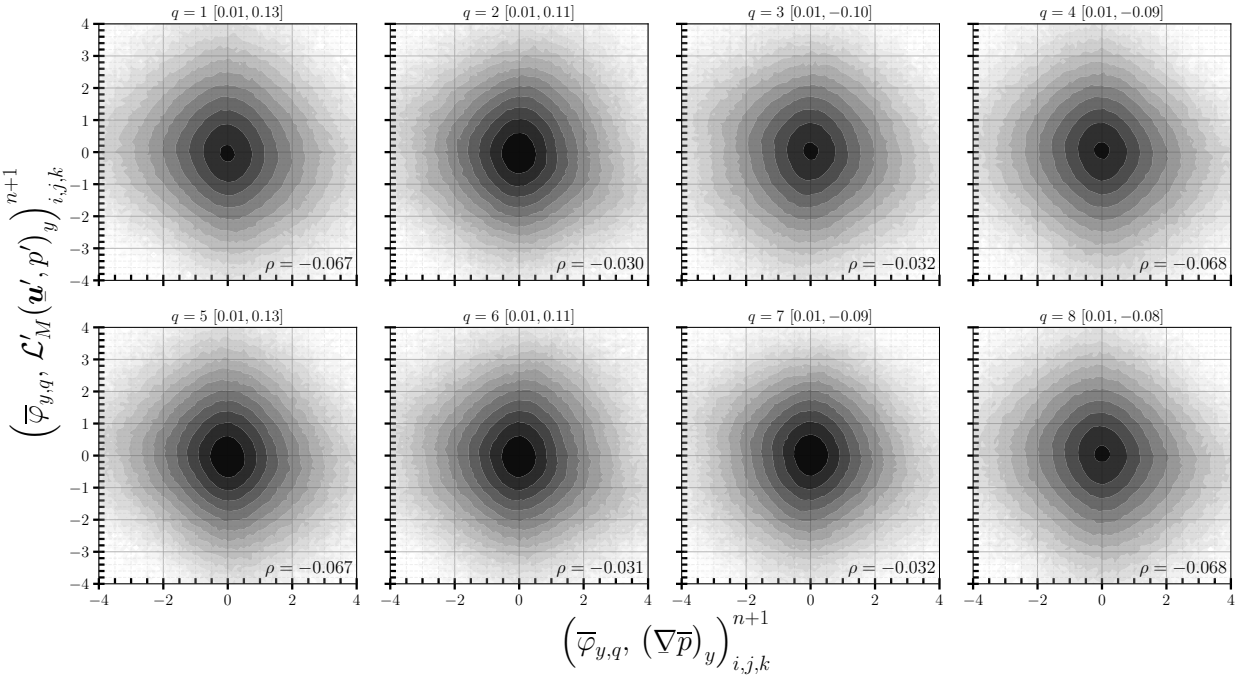


FIGURE A.28 Histograms of the t^{n+1} $(i-1, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

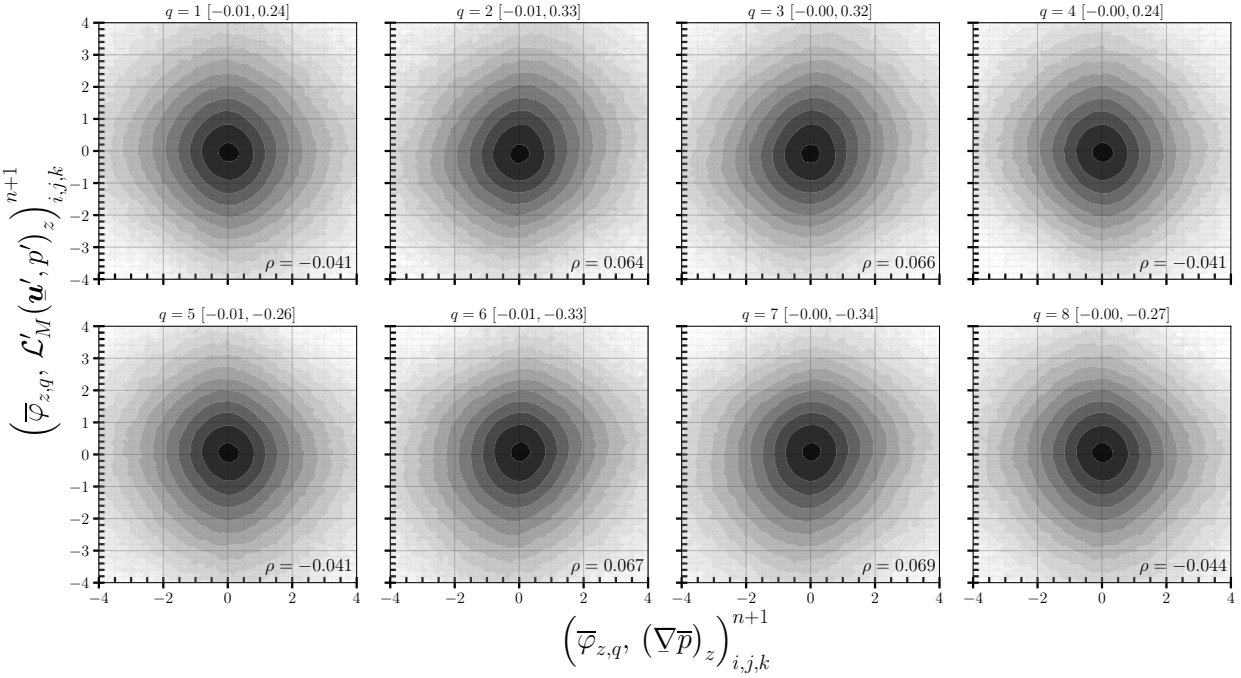


FIGURE A.29 Histograms of the t^{n+1} $(i-1, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

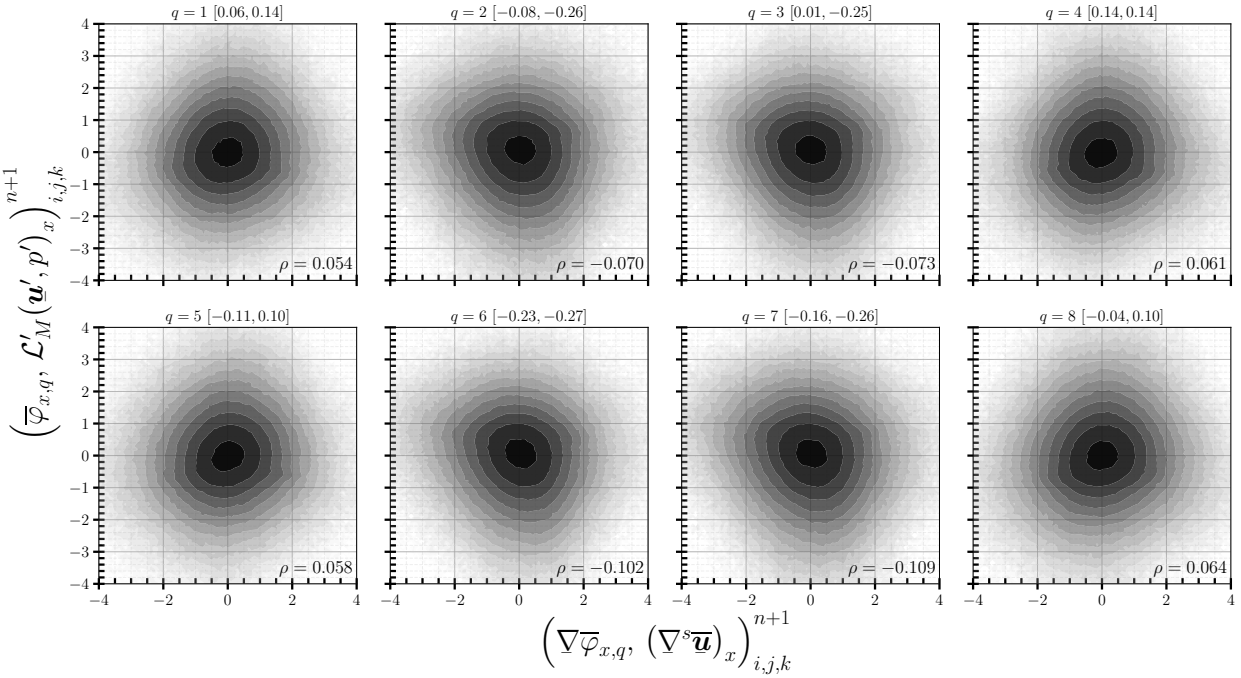


FIGURE A.30 Histograms of the t^{n+1} $(i-1, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

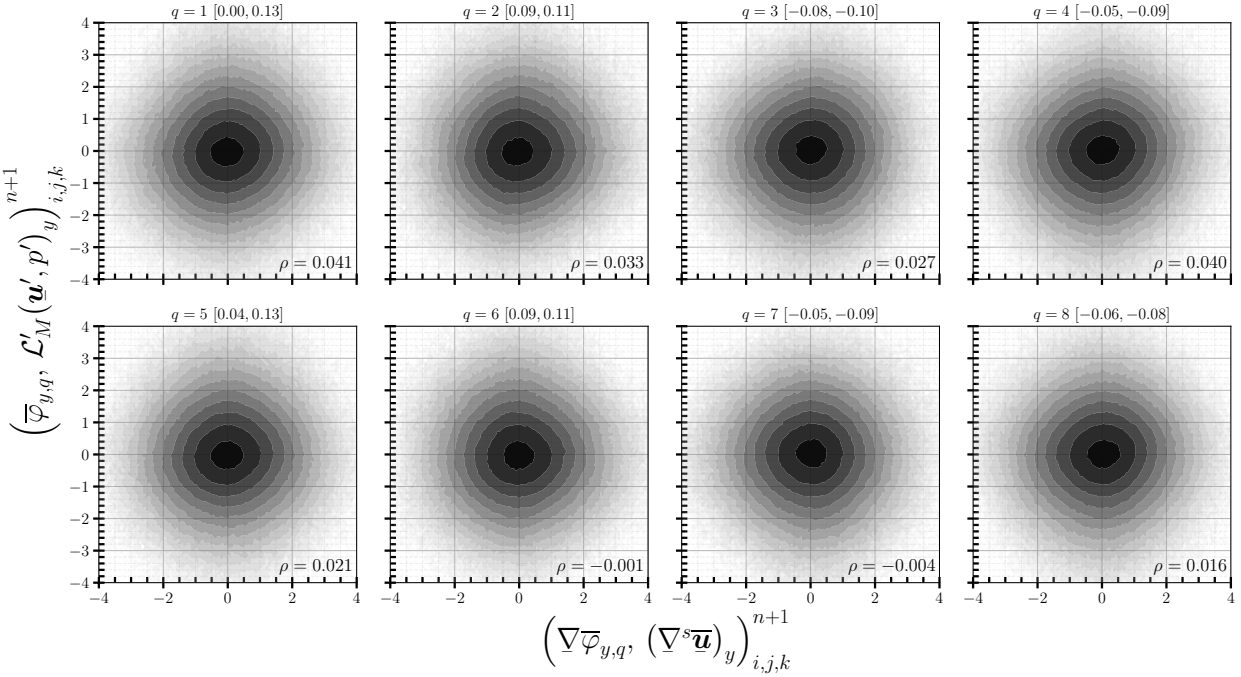


FIGURE A.31 Histograms of the t^{n+1} $(i-1, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

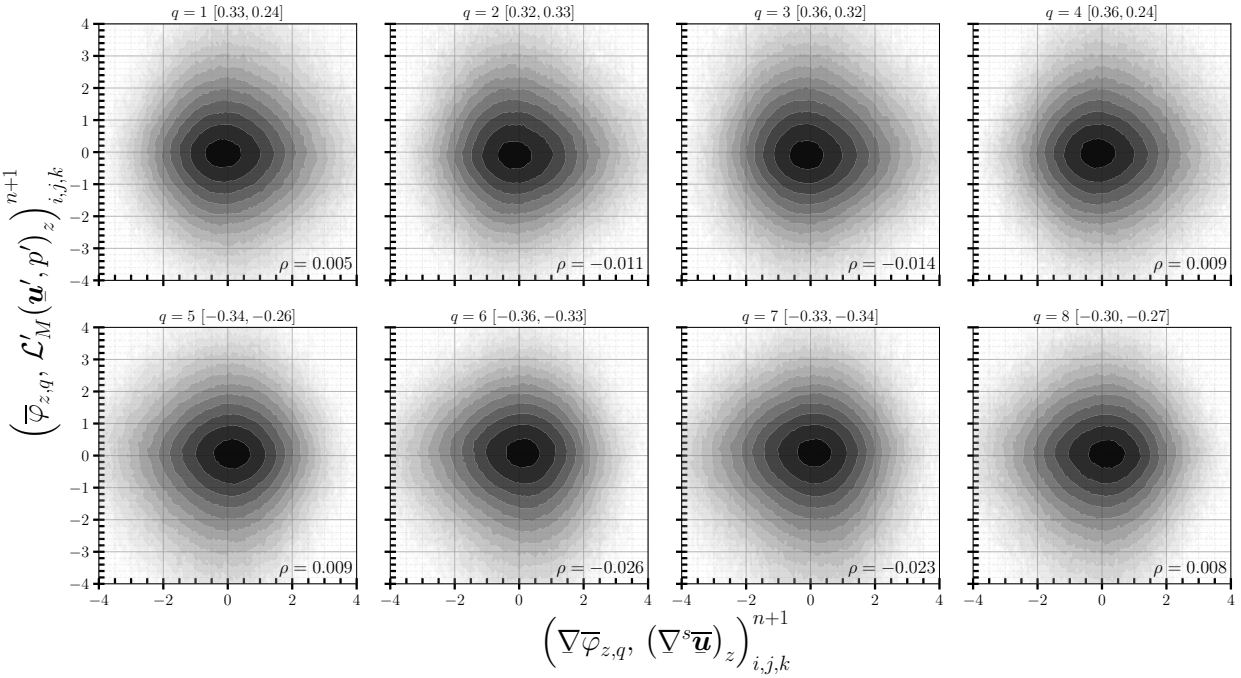


FIGURE A.32 Histograms of the t^{n+1} $(i-1, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2 t^n - t^{n+1} Histograms

A.2.1 (i, j, k) Histograms

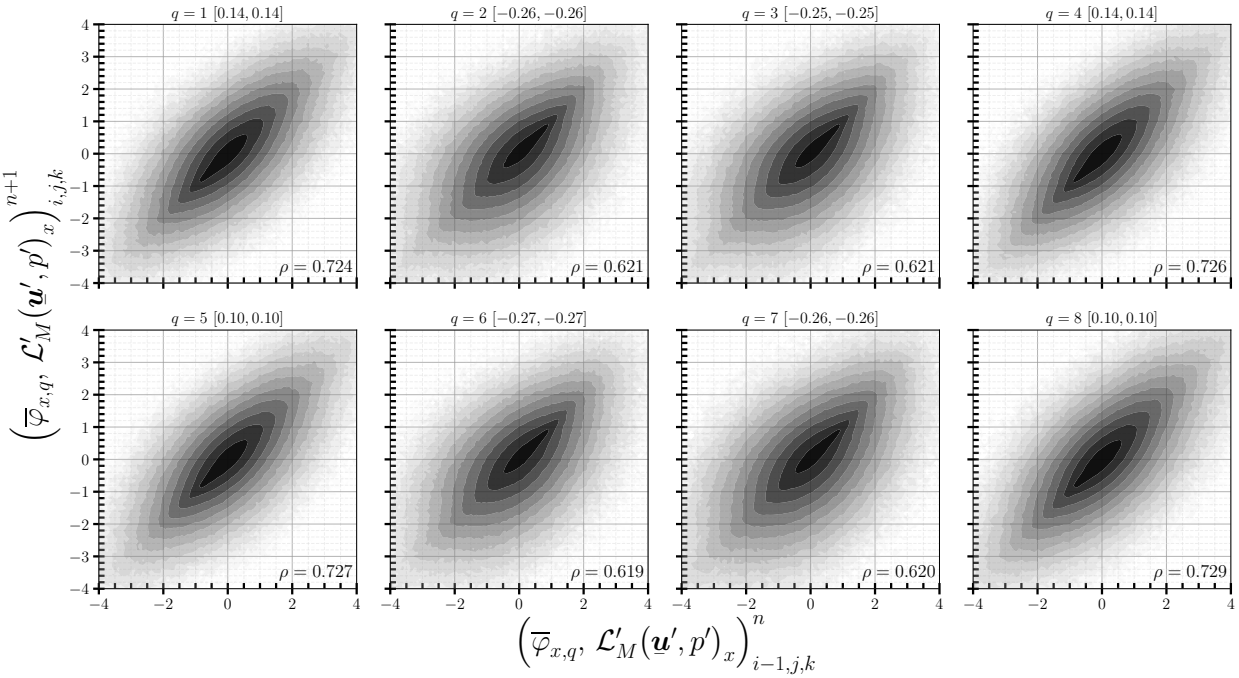


FIGURE A.33 Histograms of the t^n (i, j, k)-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

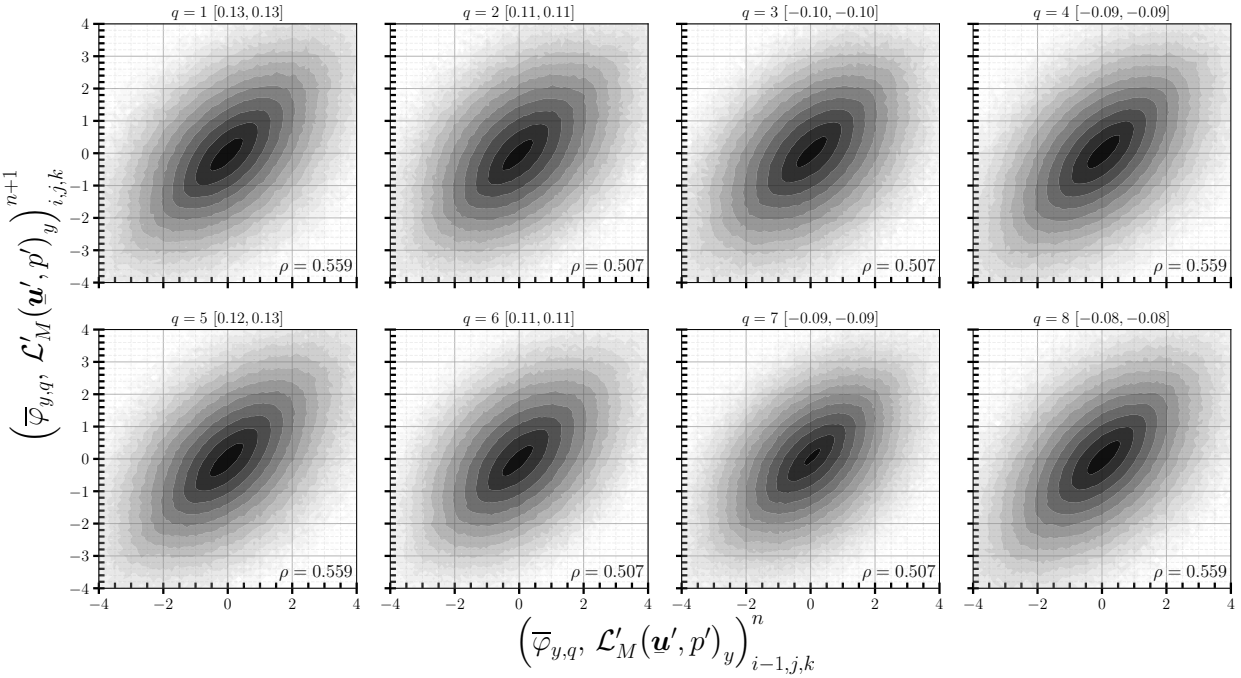


FIGURE A.34 Histograms of the t^n (i, j, k)-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

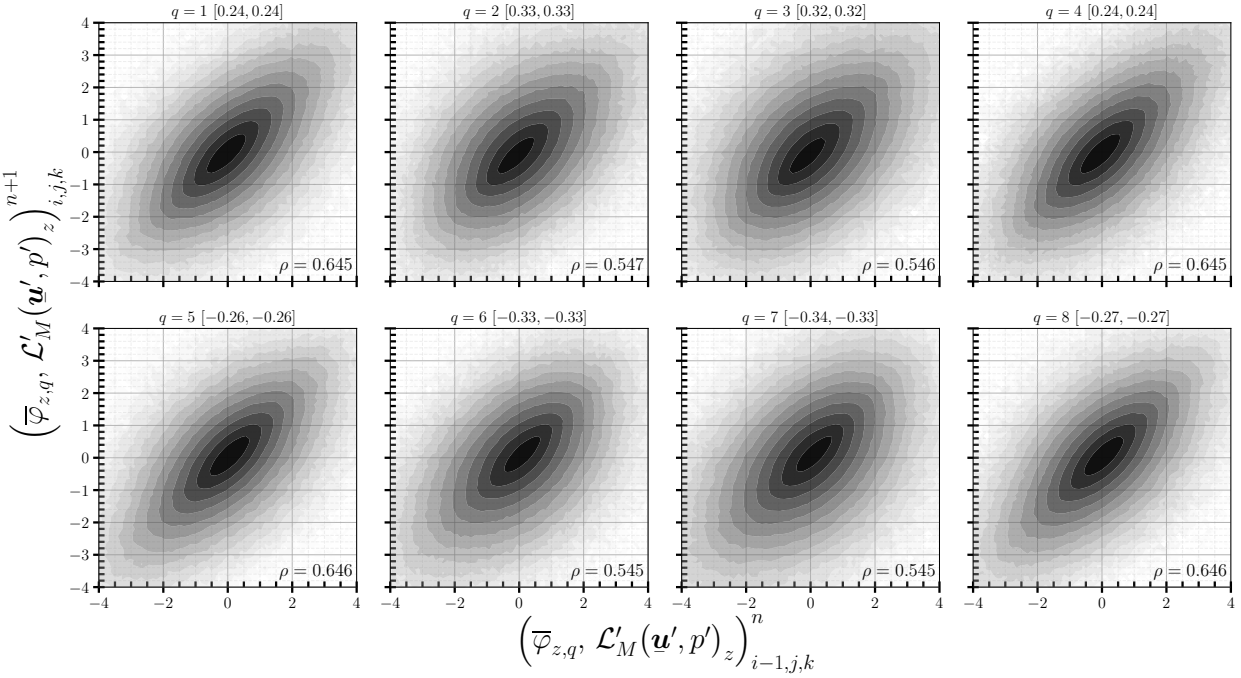


FIGURE A.35 Histograms of the t^n (i, j, k)-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

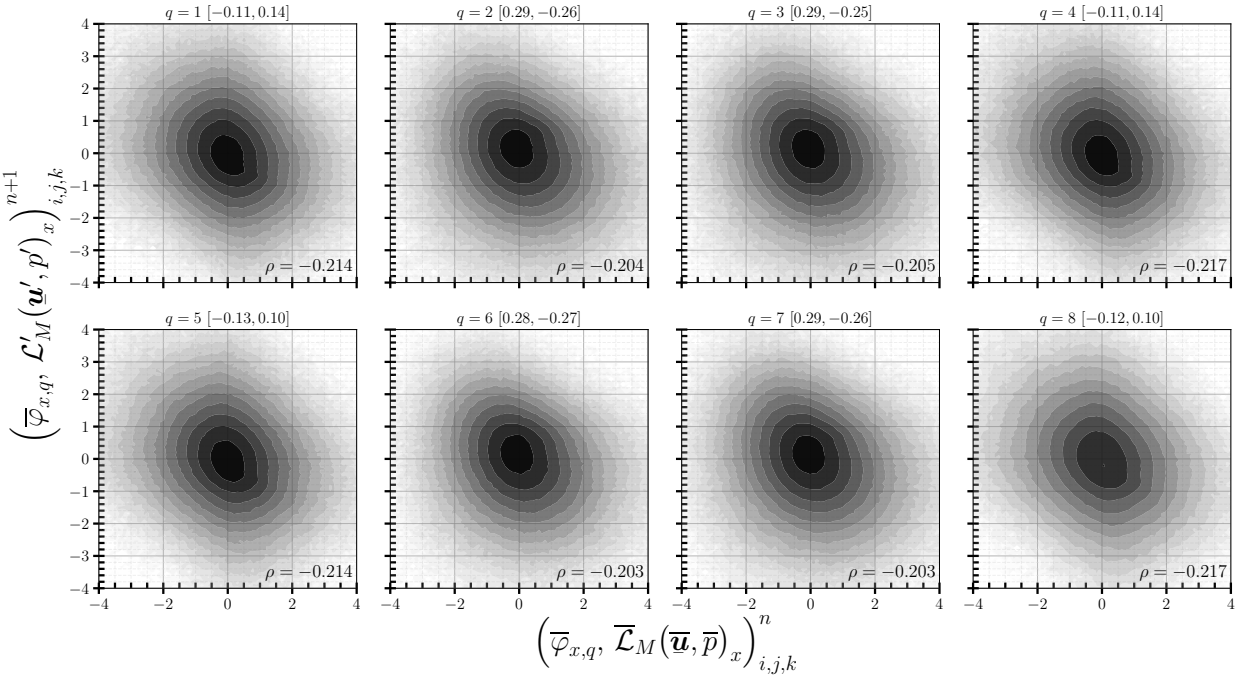


FIGURE A.36 Histograms of the t^n (i, j, k)-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

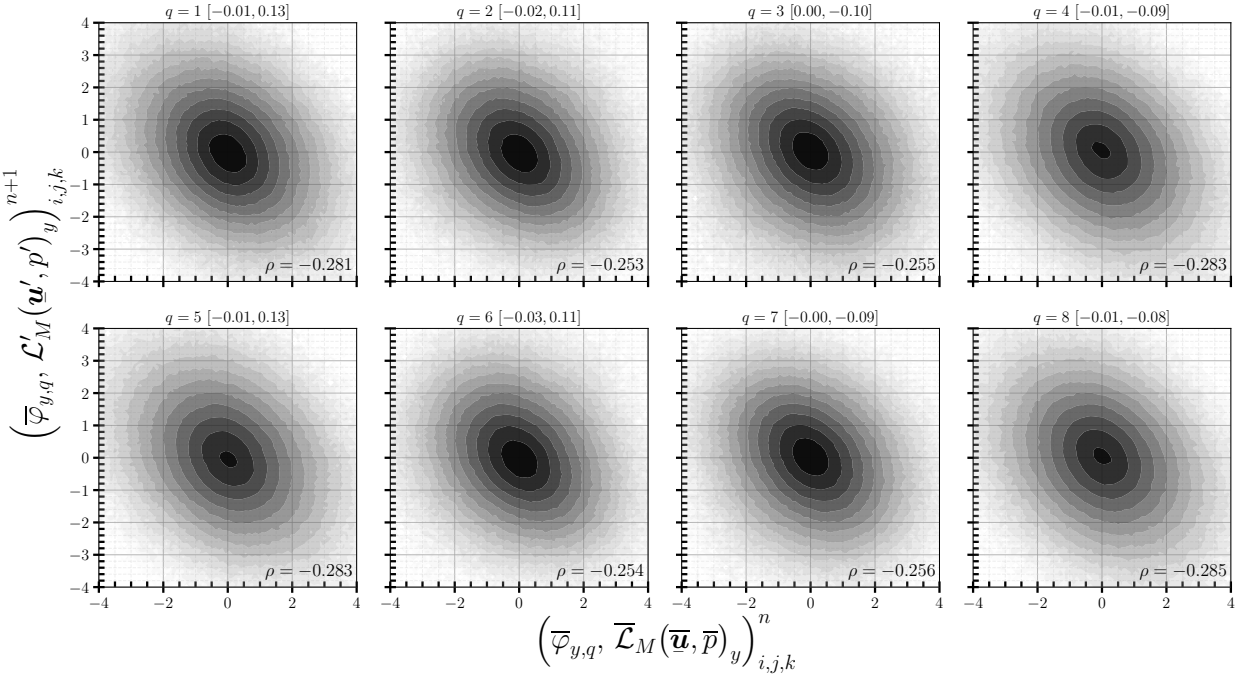


FIGURE A.37 Histograms of the t^n (i, j, k)-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

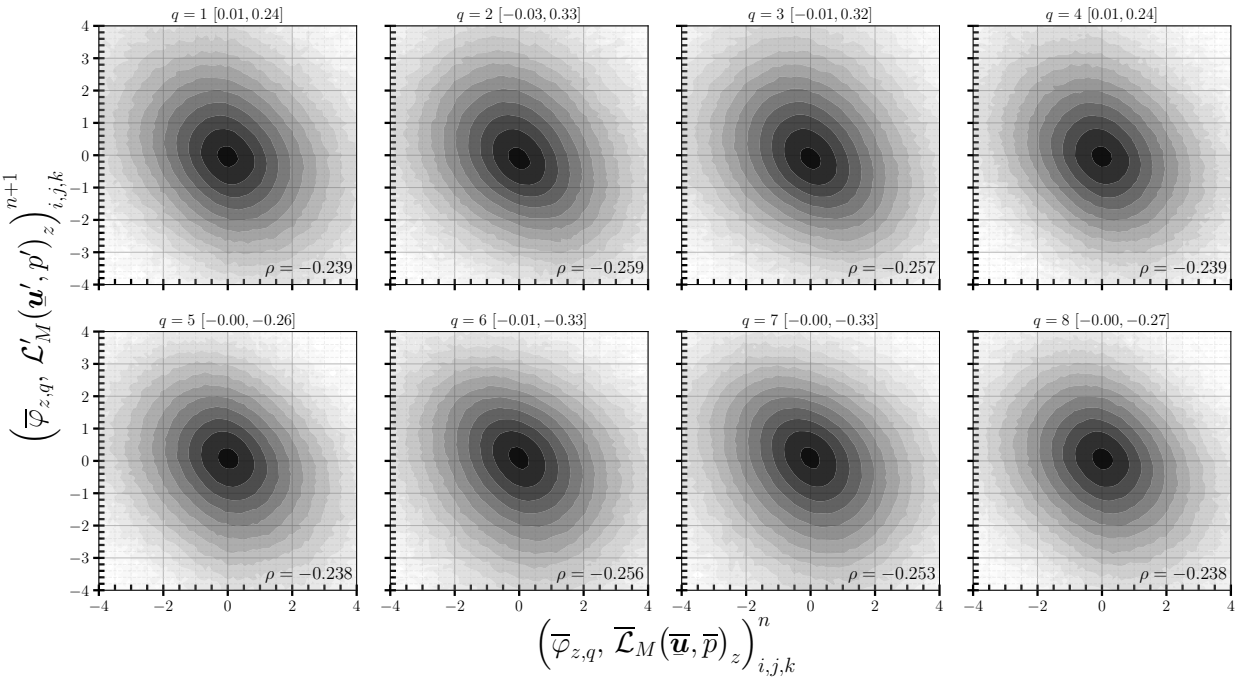


FIGURE A.38 Histograms of the t^n (i, j, k)-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

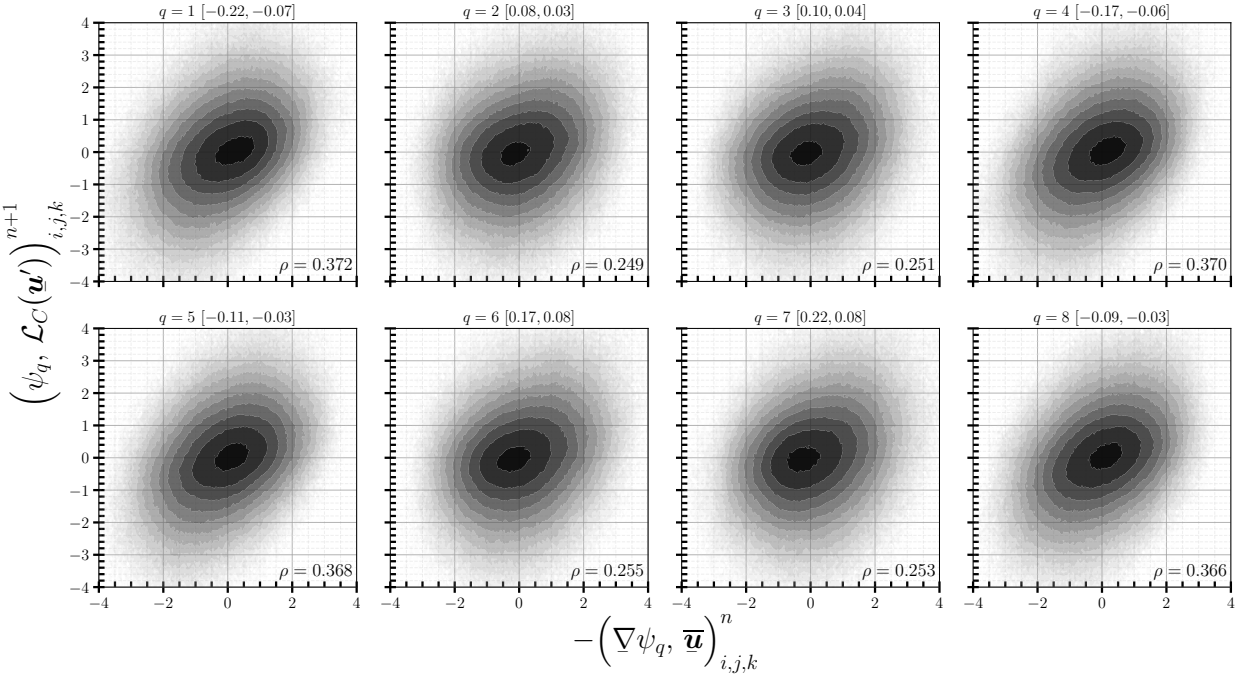


FIGURE A.39 Histograms of the t^n (i, j, k)-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

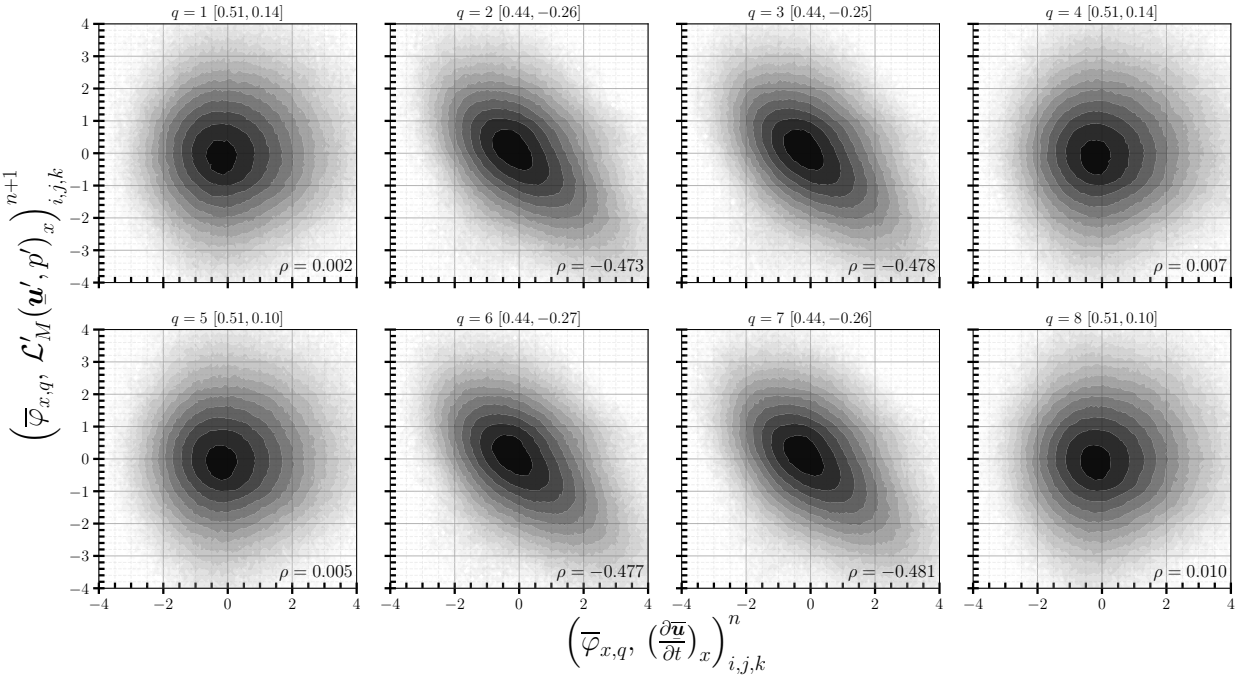


FIGURE A.40 Histograms of the t^n (i, j, k)-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

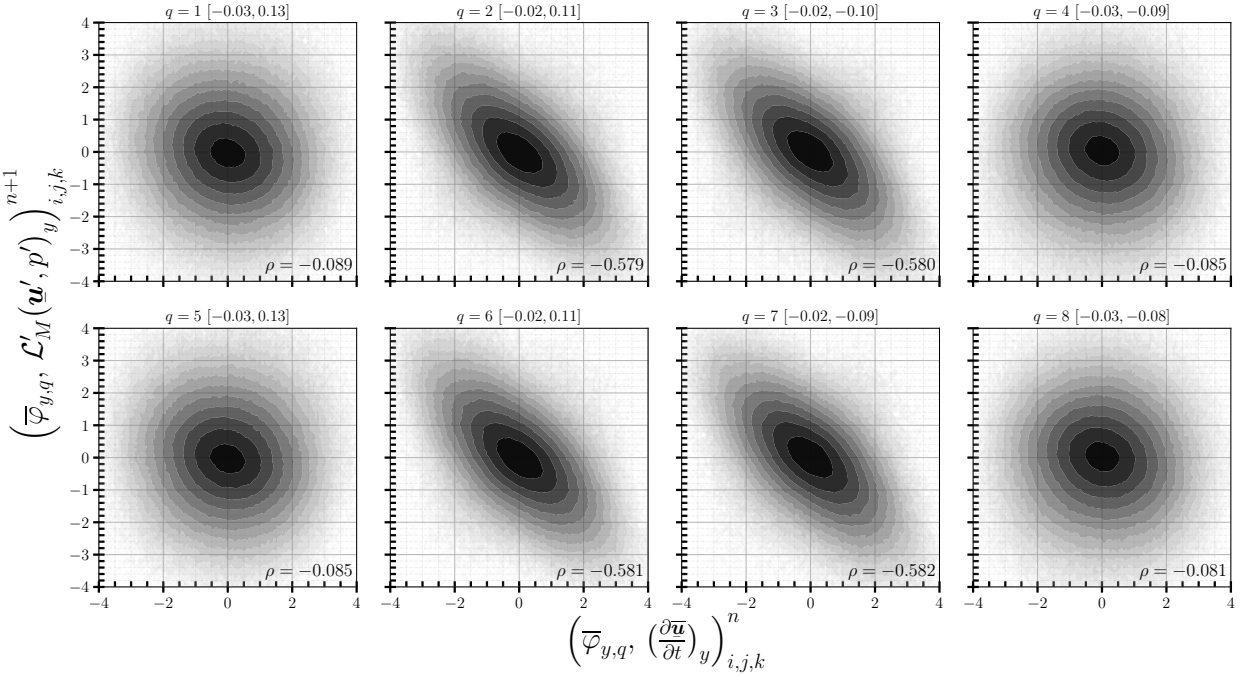


FIGURE A.41 Histograms of the t^n (i, j, k)-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

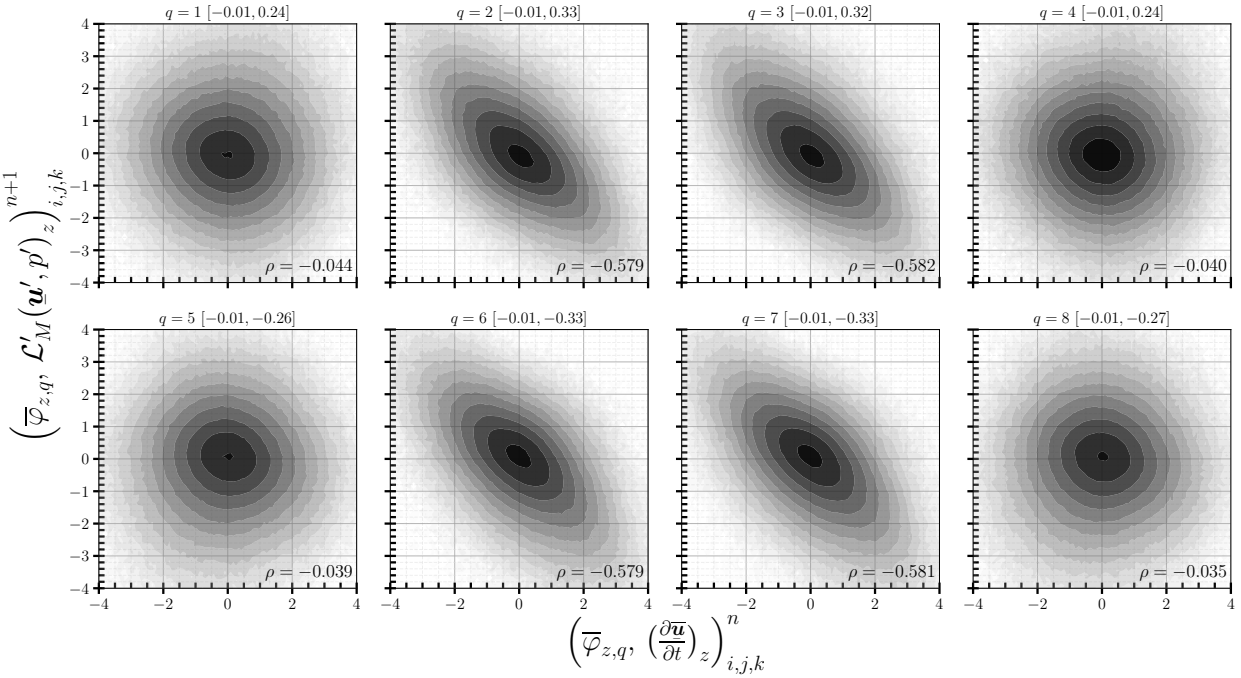


FIGURE A.42 Histograms of the t^n (i, j, k)-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

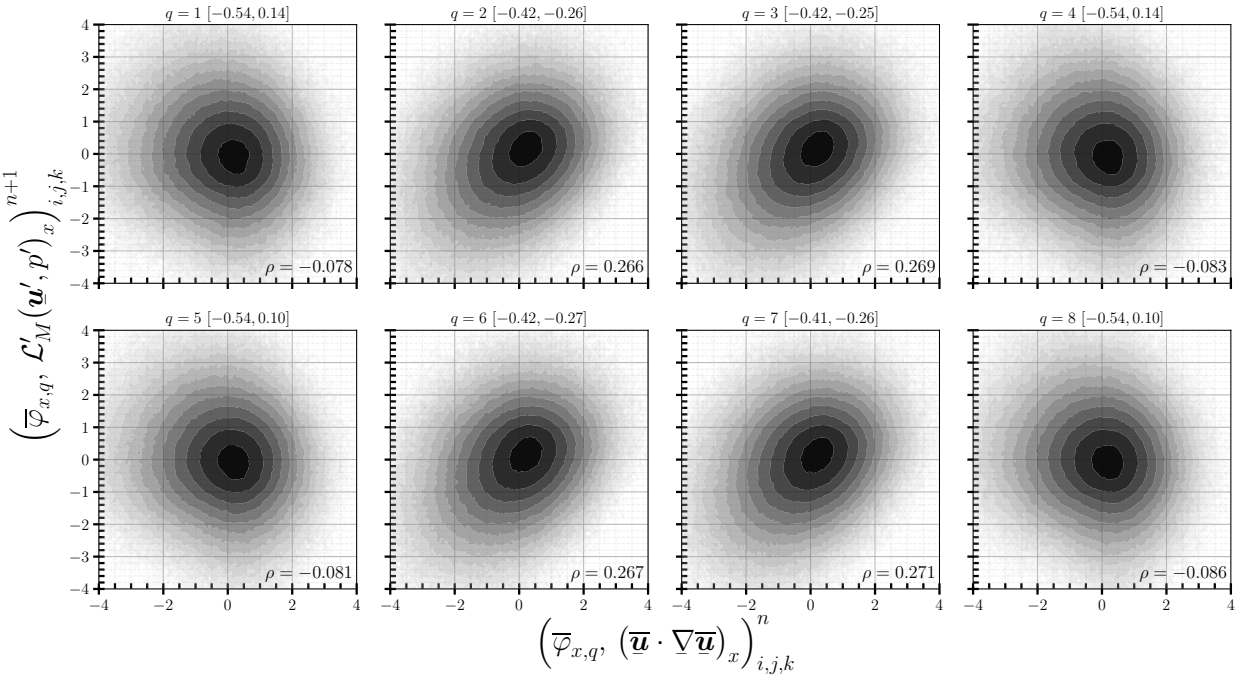


FIGURE A.43 Histograms of the t^n (i, j, k)-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

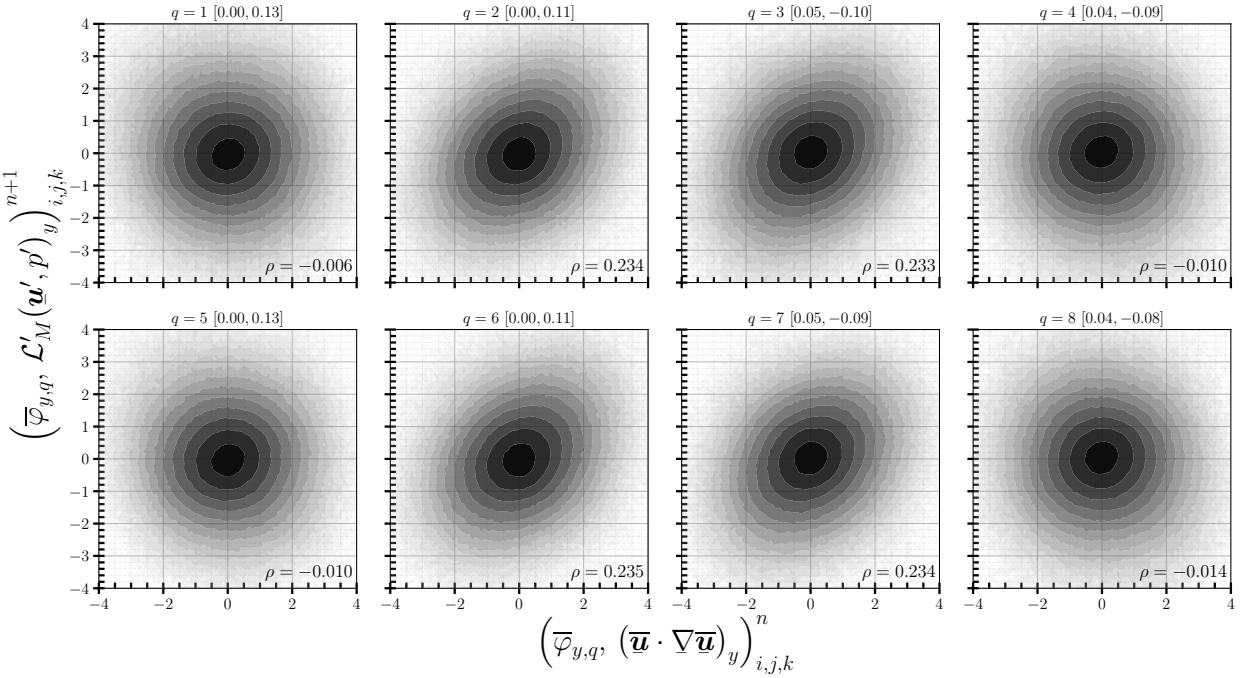


FIGURE A.44 Histograms of the t^n (i, j, k)-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

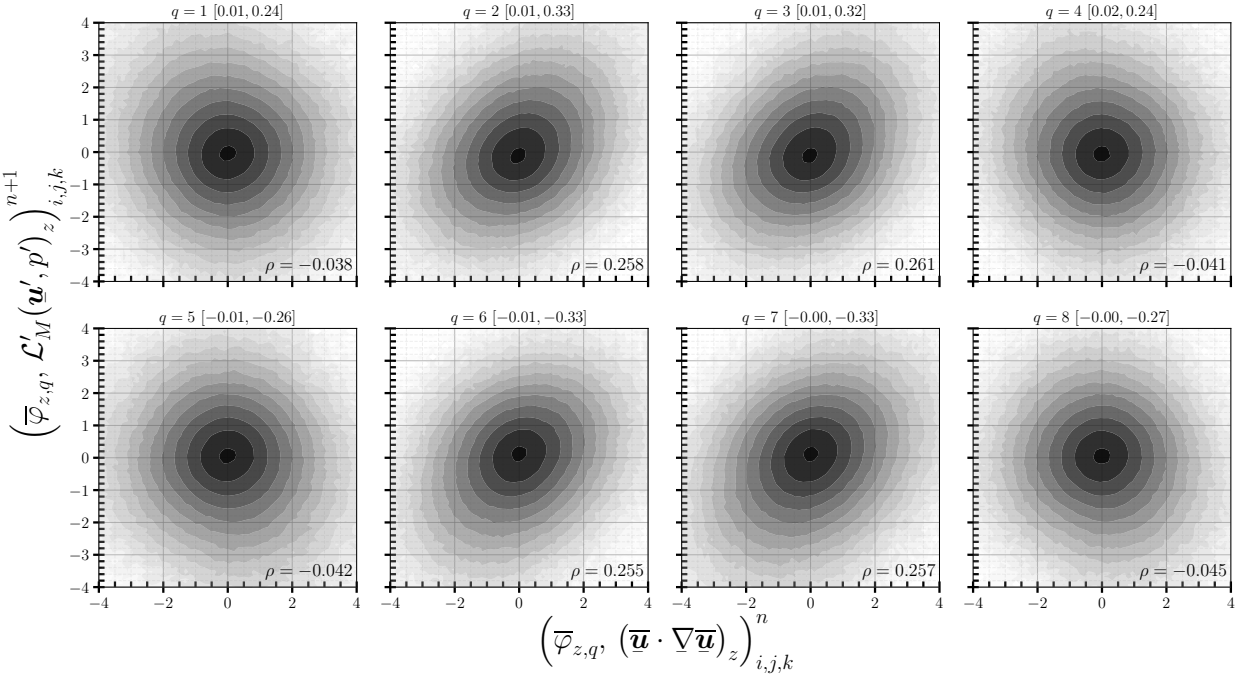


FIGURE A.45 Histograms of the t^n (i, j, k)-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

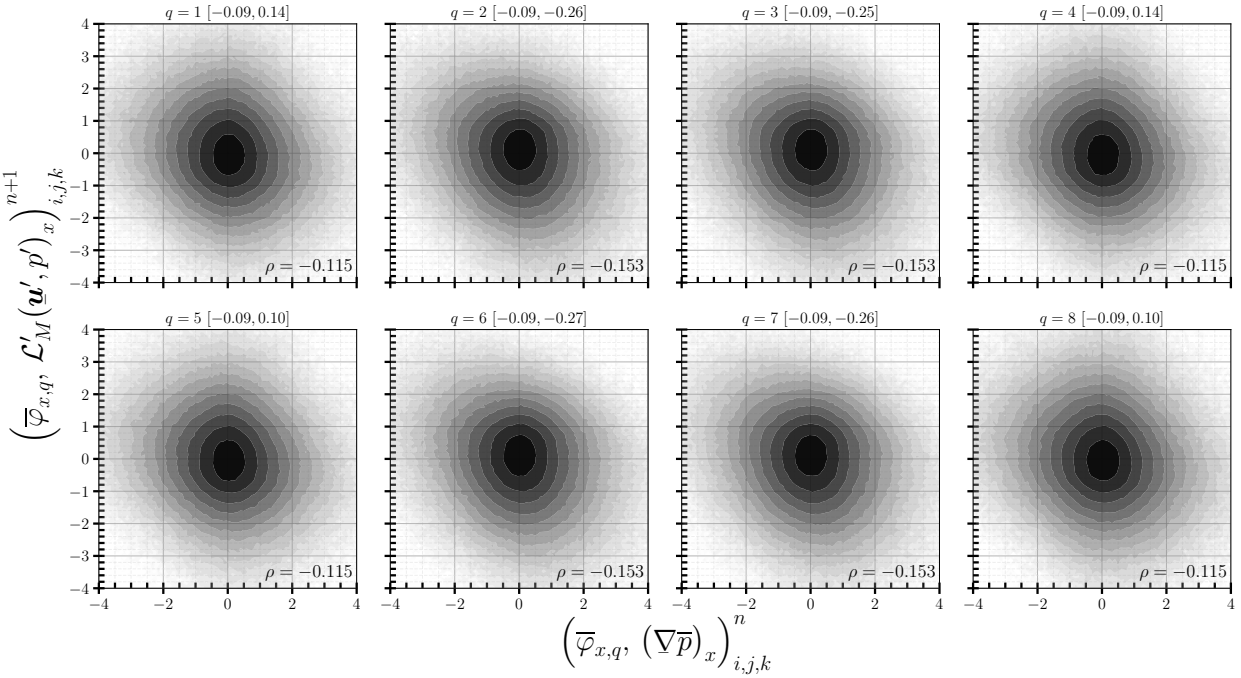


FIGURE A.46 Histograms of the t^n (i, j, k)-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

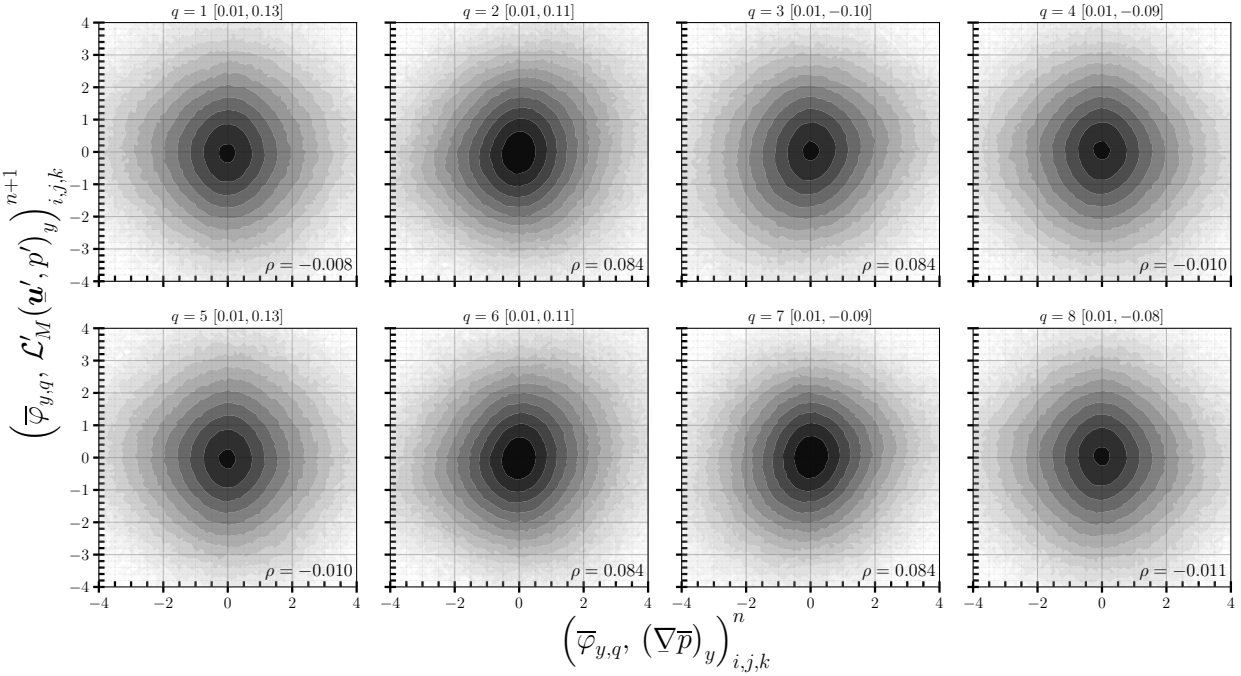


FIGURE A.47 Histograms of the t^n (i, j, k)-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

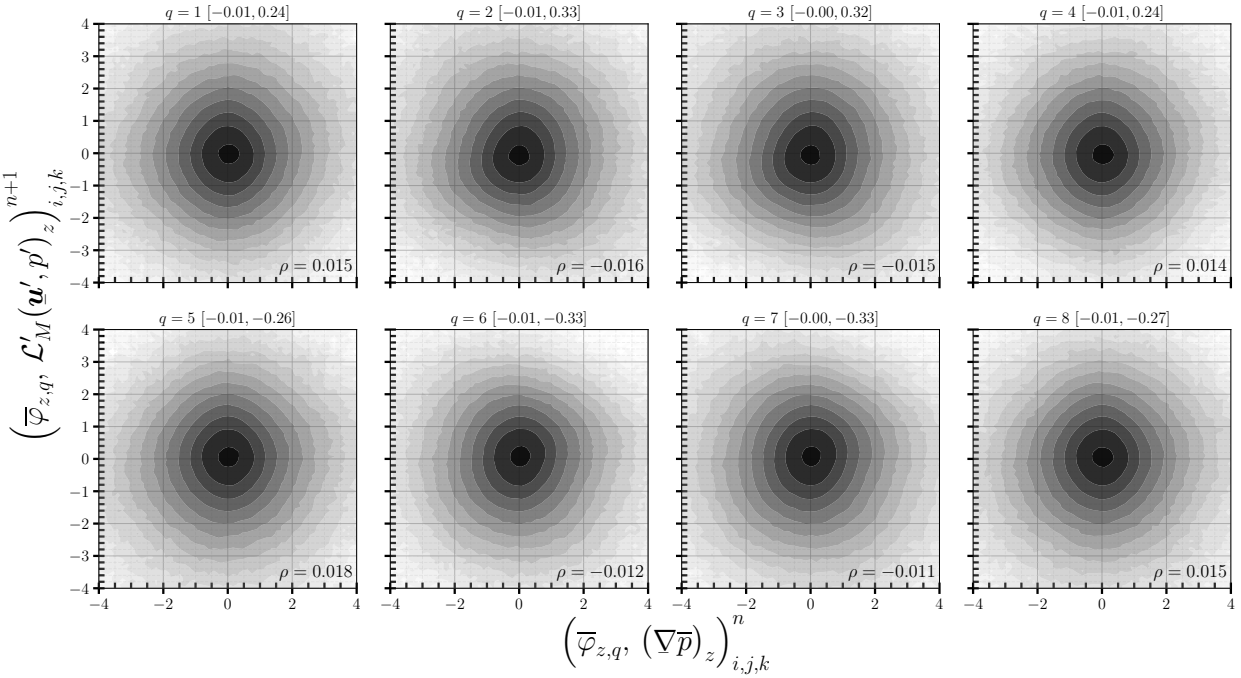


FIGURE A.48 Histograms of the t^n (i, j, k)-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

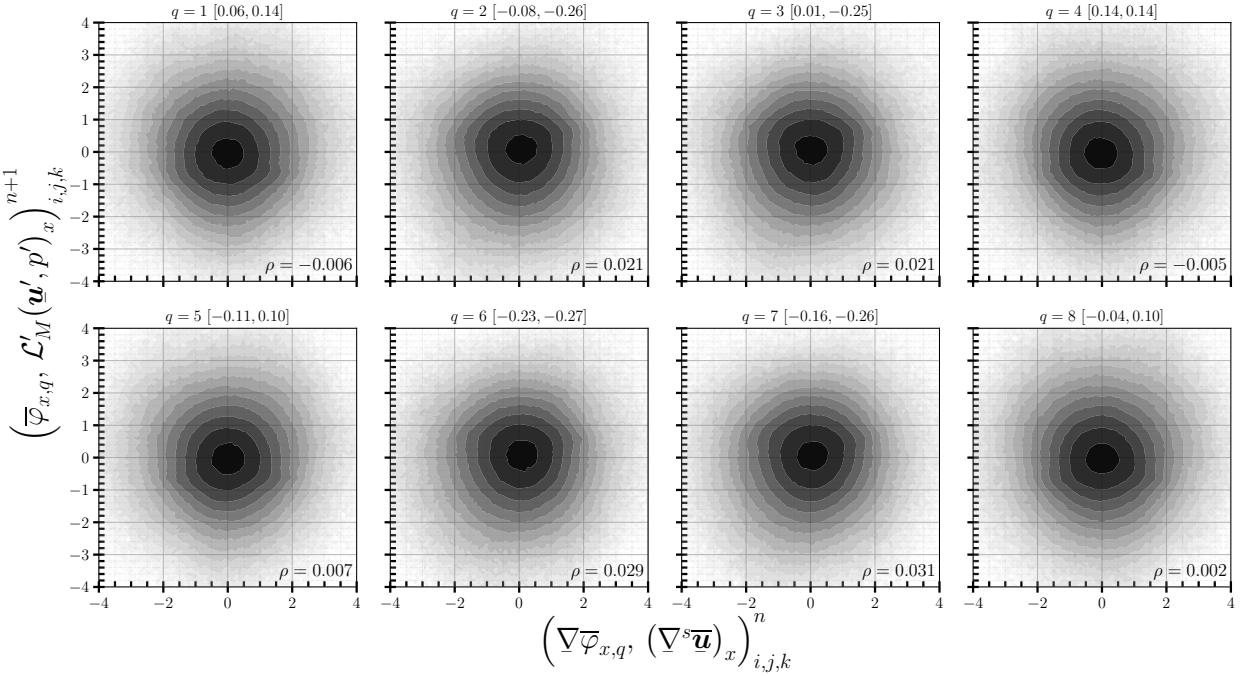


FIGURE A.49 Histograms of the t^n (i, j, k)-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

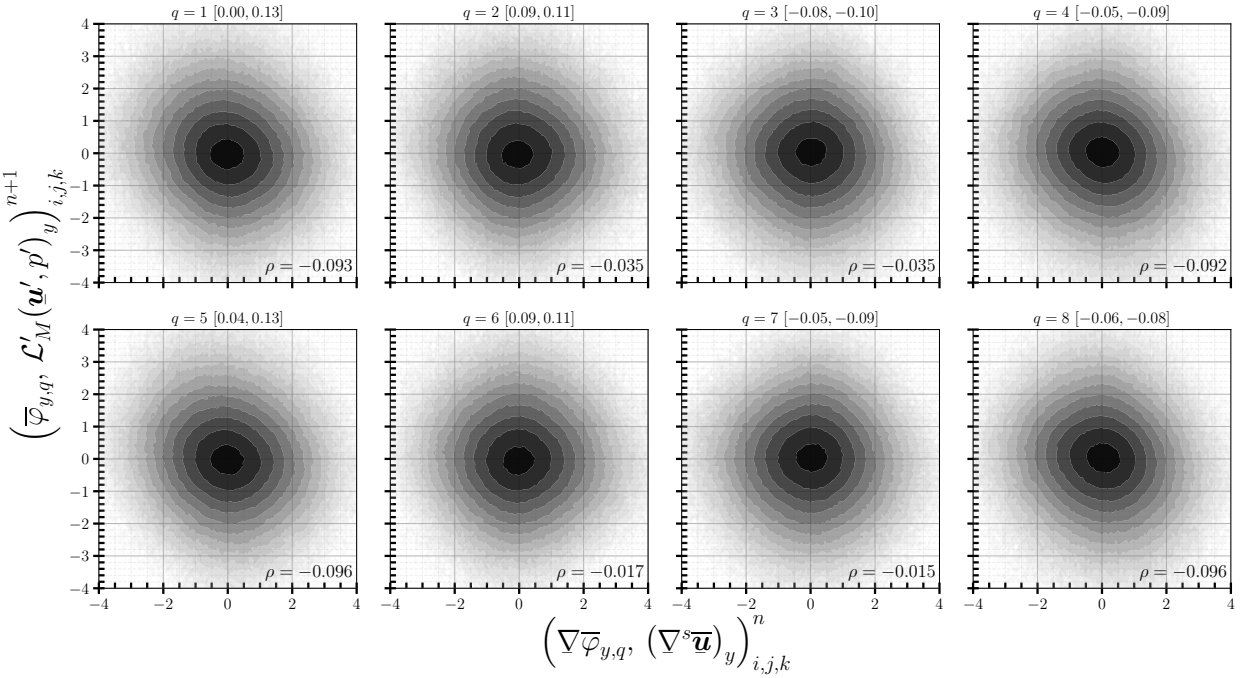


FIGURE A.50 Histograms of the t^n (i, j, k)-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

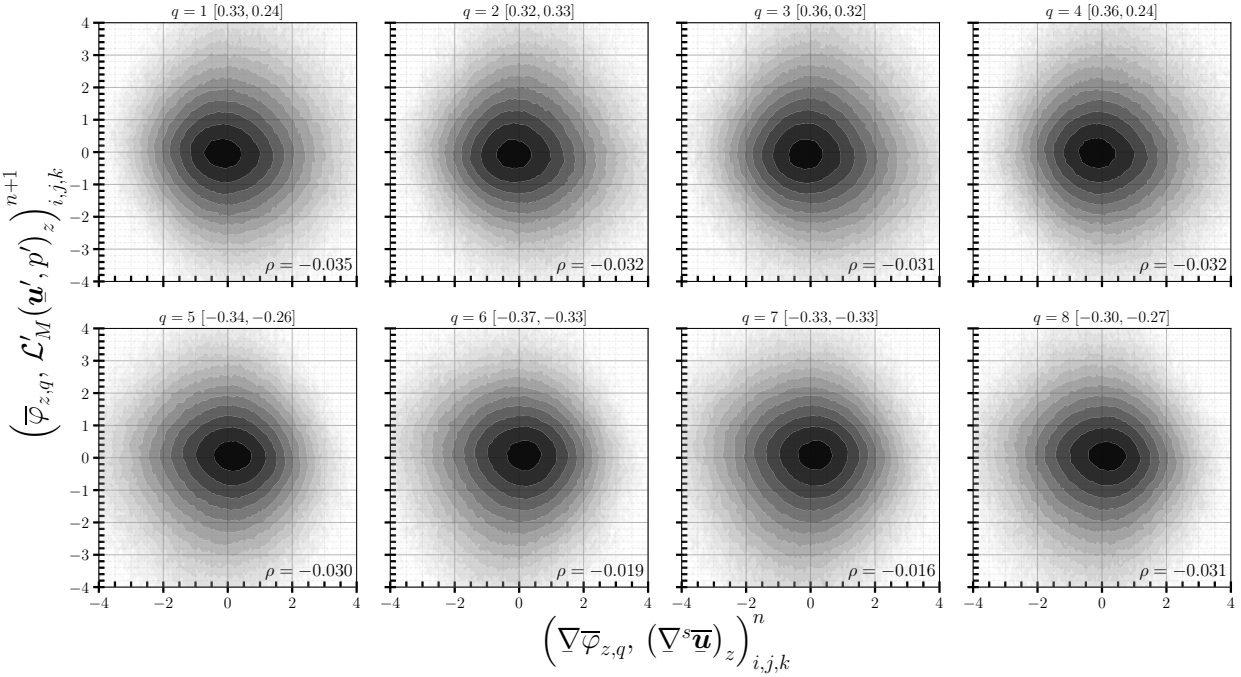


FIGURE A.51 Histograms of the t^n (i, j, k)-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.2 $(i-1, j, k)$ Histograms

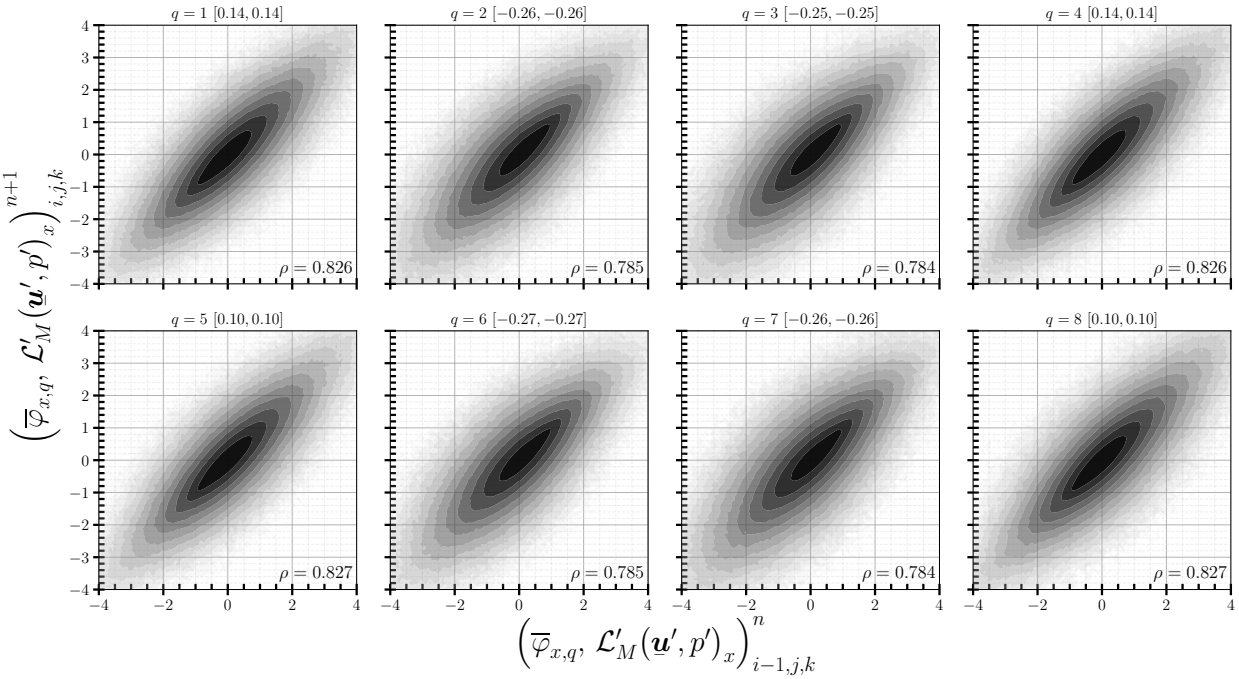


FIGURE A.52 Histograms of the t^n $(i-1, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

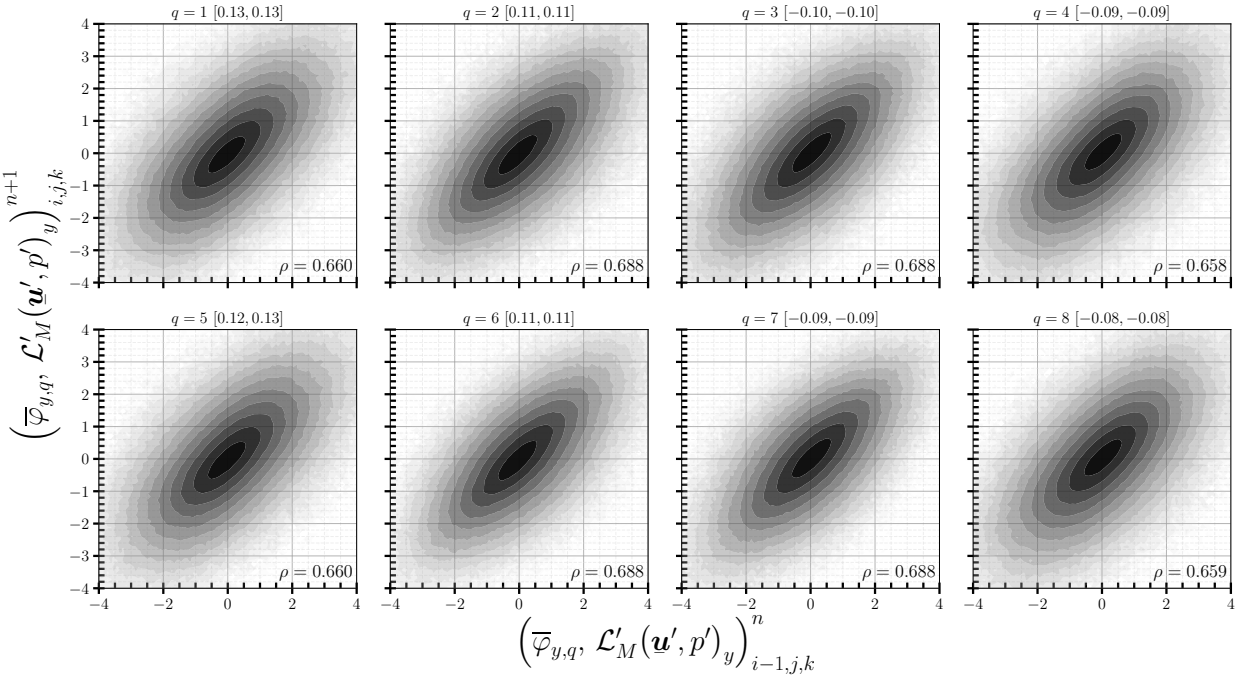


FIGURE A.53 Histograms of the t^n $(i-1, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

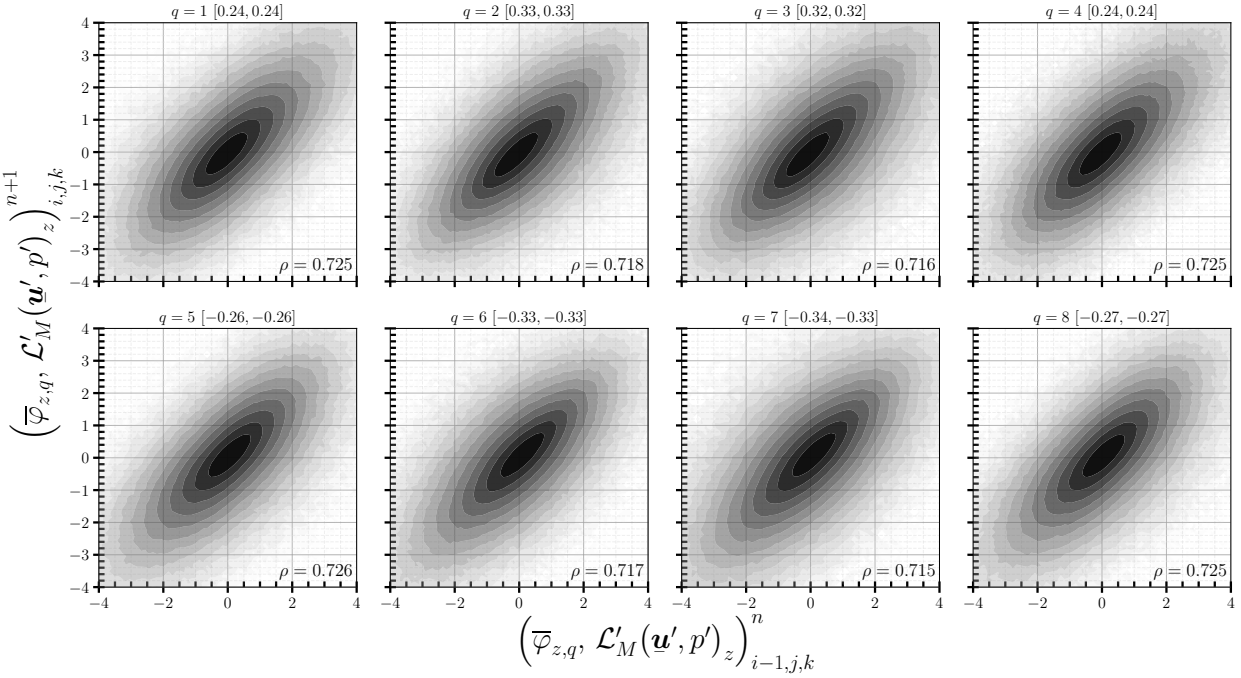


FIGURE A.54 Histograms of the t^n $(i-1, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

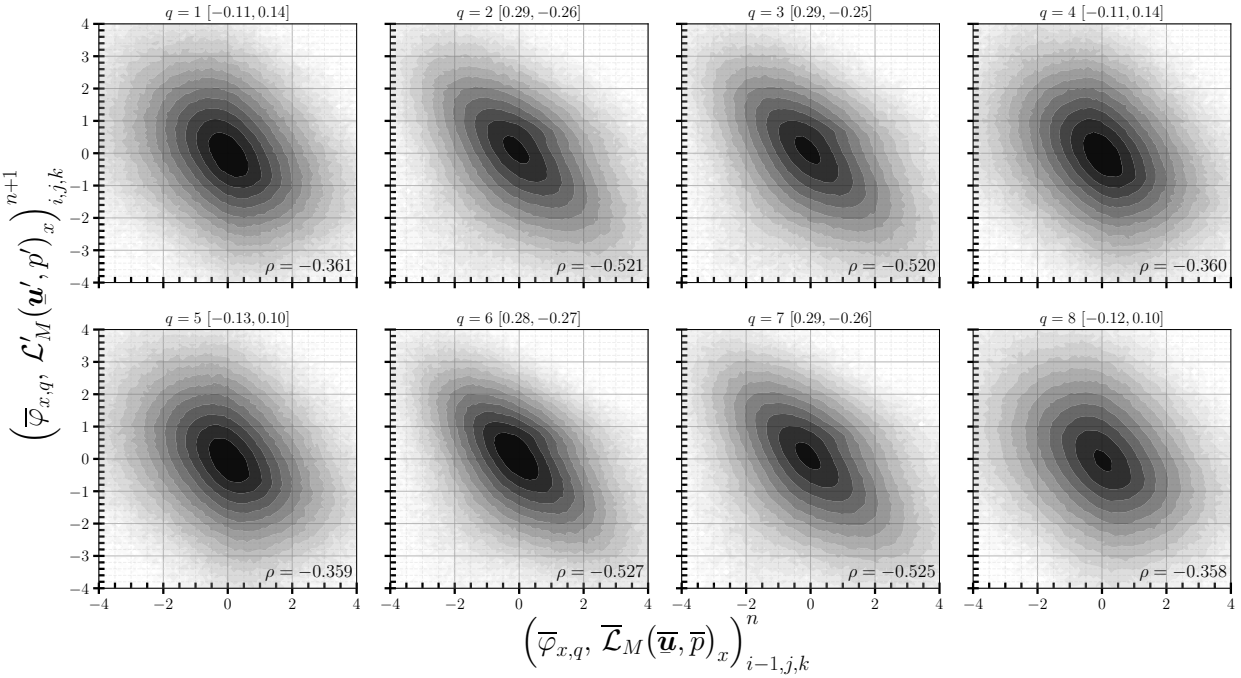


FIGURE A.55 Histograms of the t^n $(i-1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

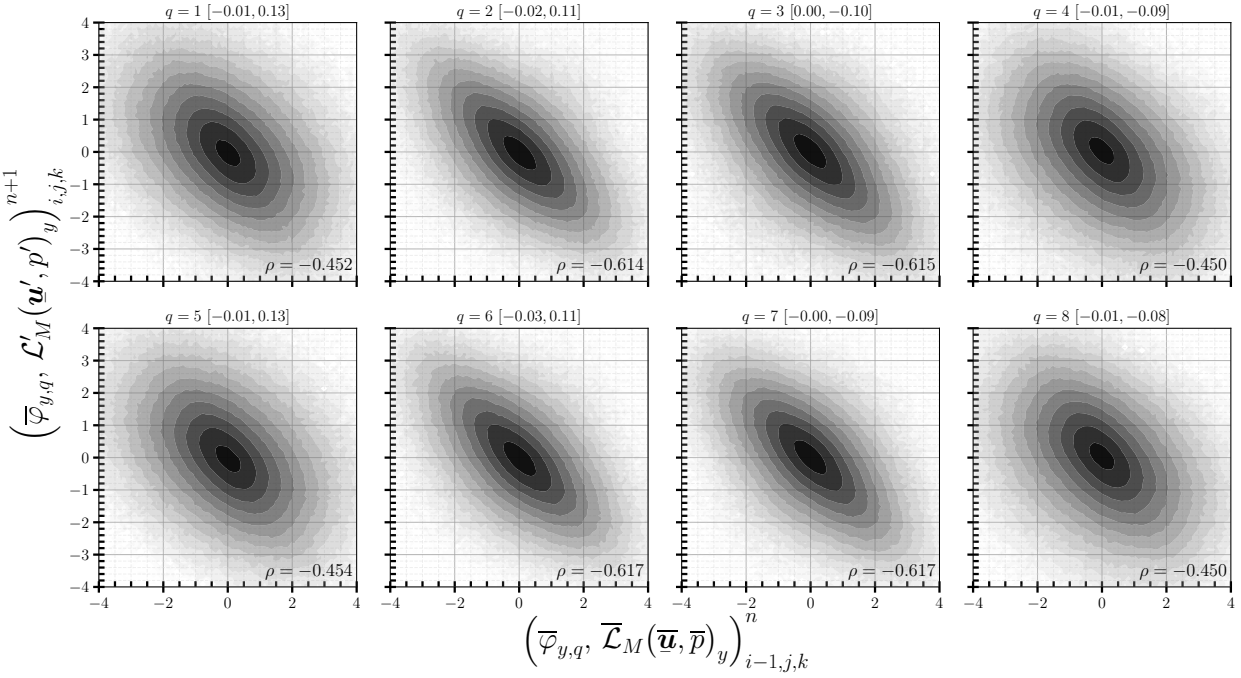


FIGURE A.56 Histograms of the t^n $(i-1, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

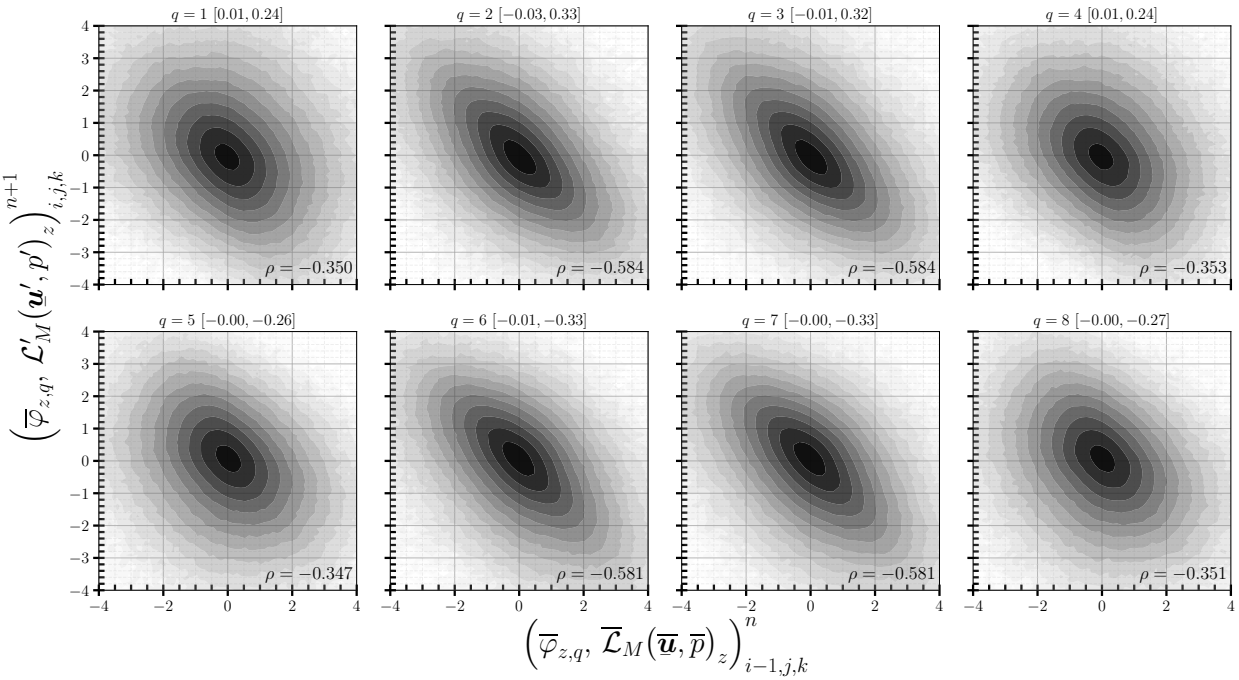


FIGURE A.57 Histograms of the t^n $(i-1, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

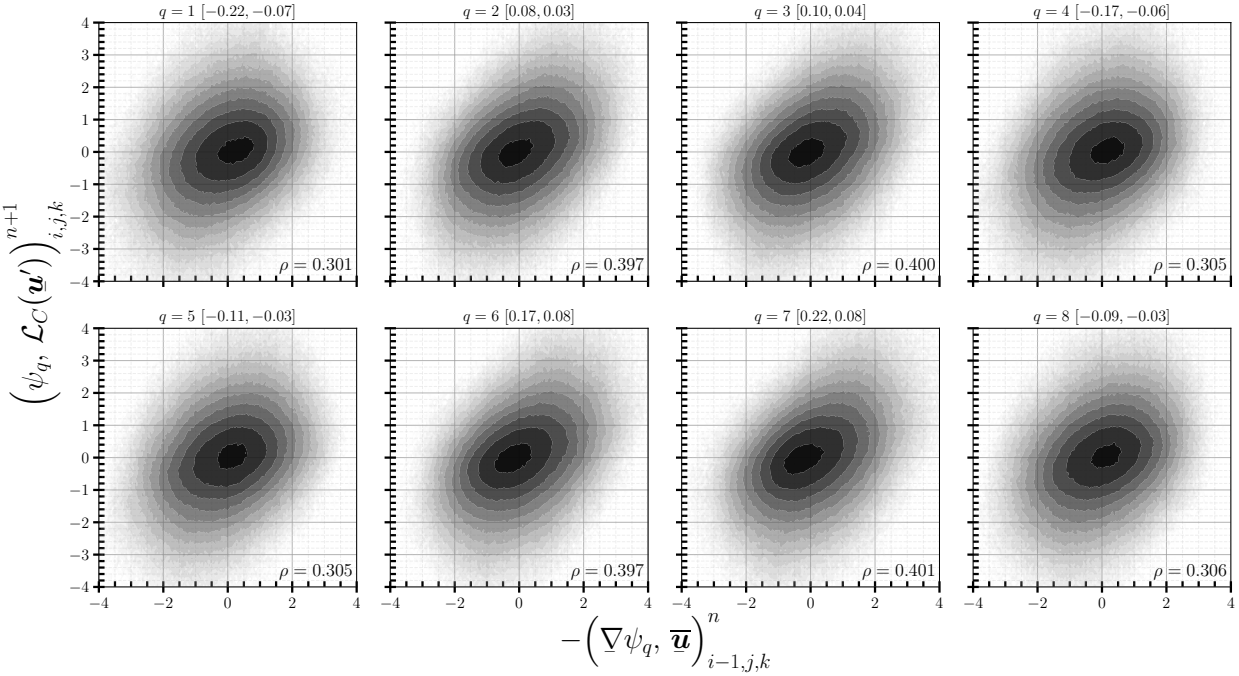


FIGURE A.58 Histograms of the t^n $(i-1, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

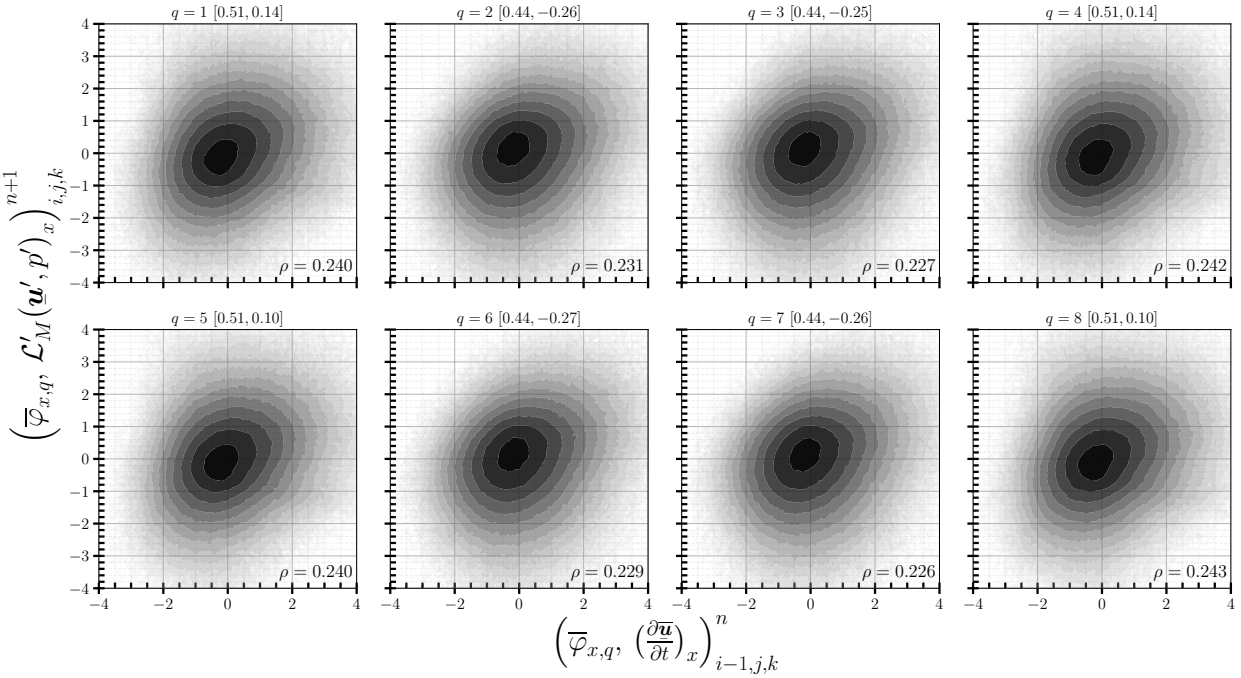


FIGURE A.59 Histograms of the t^n $(i-1, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

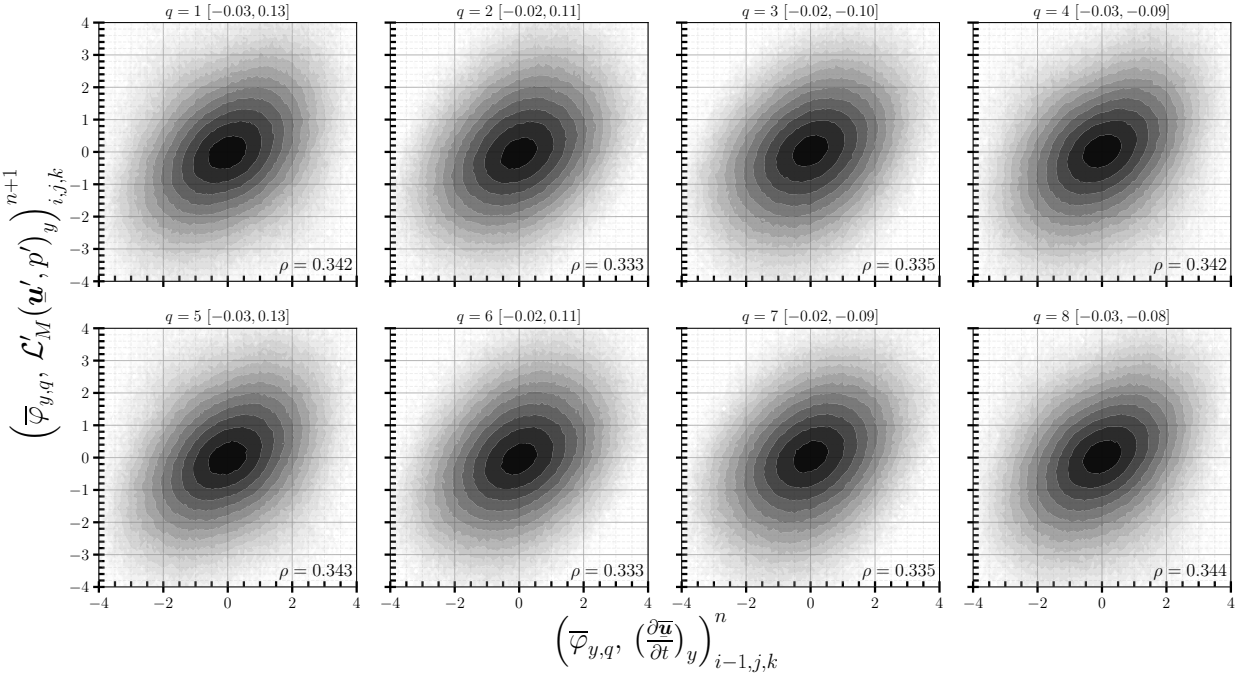


FIGURE A.60 Histograms of the t^n $(i-1, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

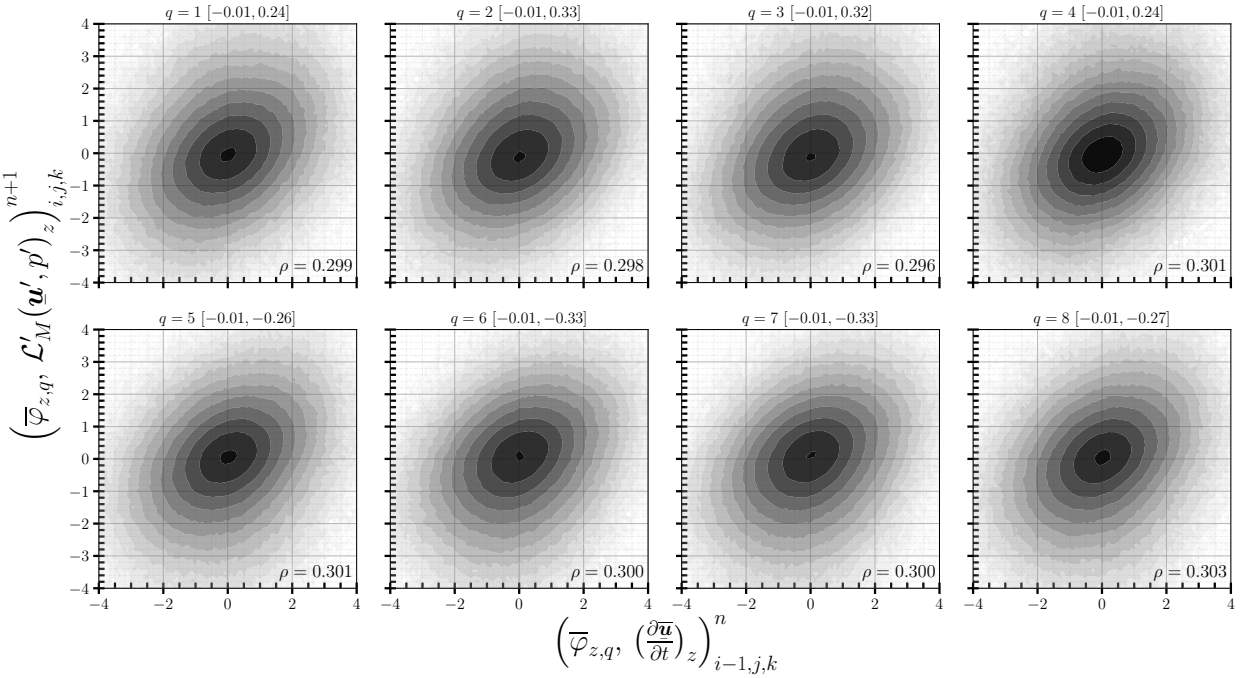


FIGURE A.61 Histograms of the t^n $(i-1, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

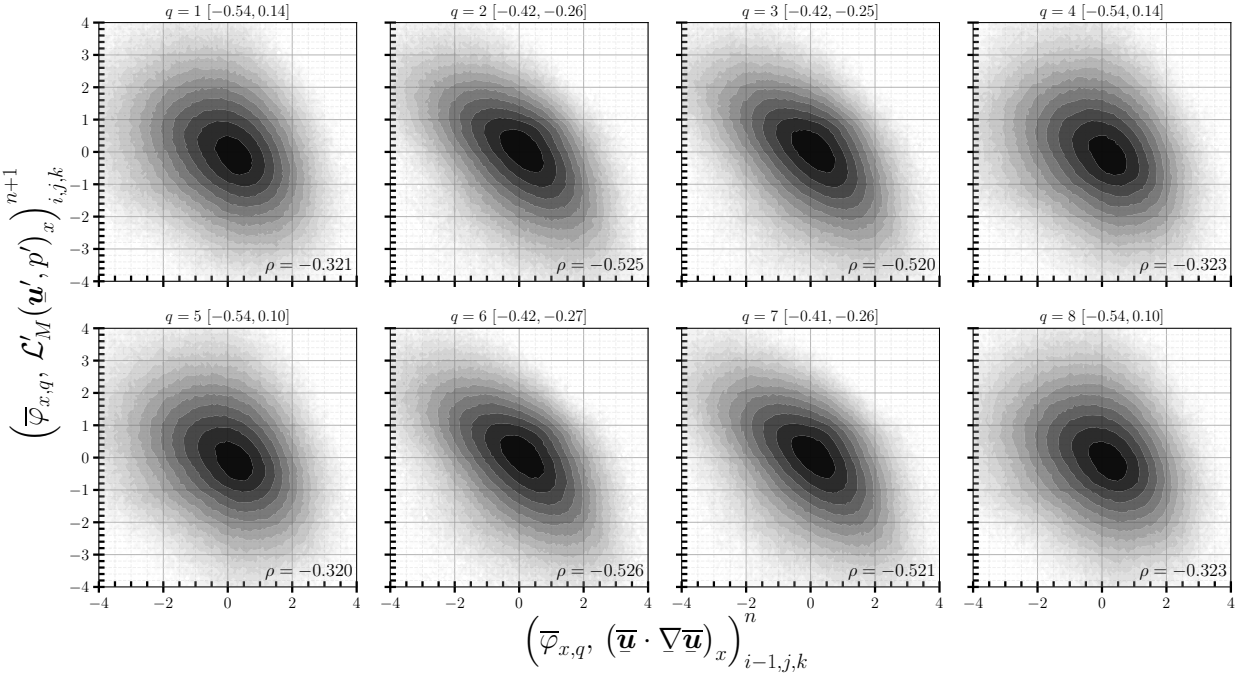


FIGURE A.62 Histograms of the t^n $(i-1, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

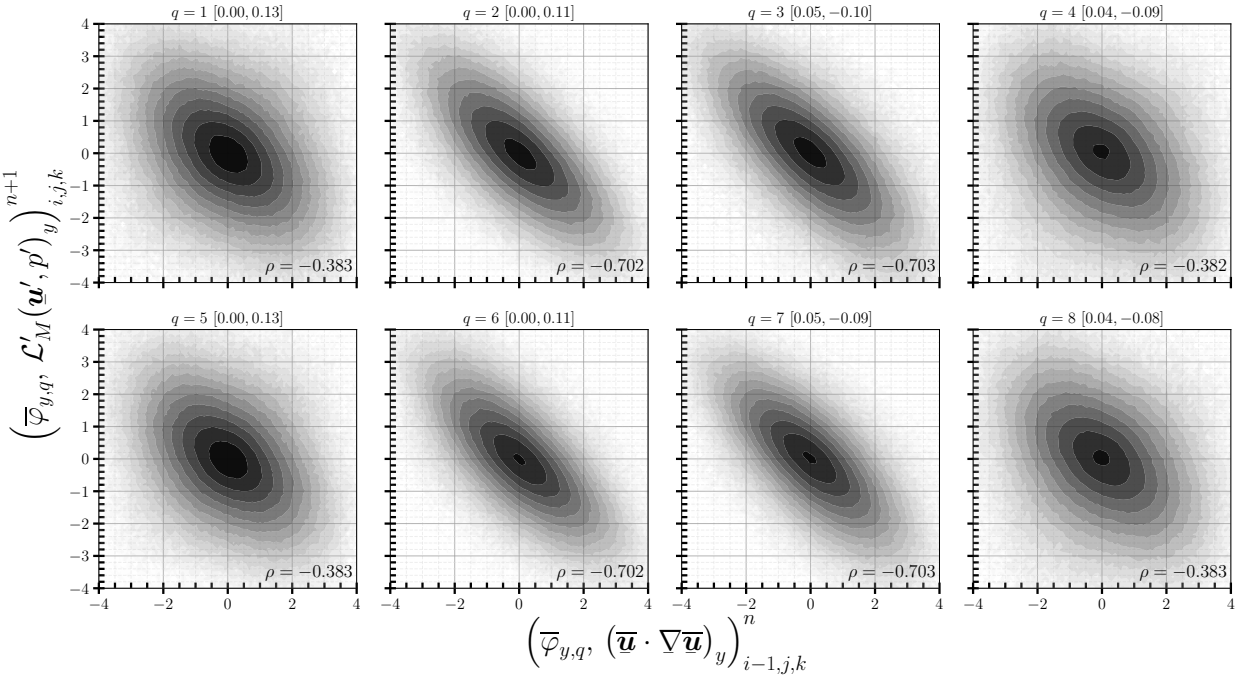


FIGURE A.63 Histograms of the t^n $(i-1, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

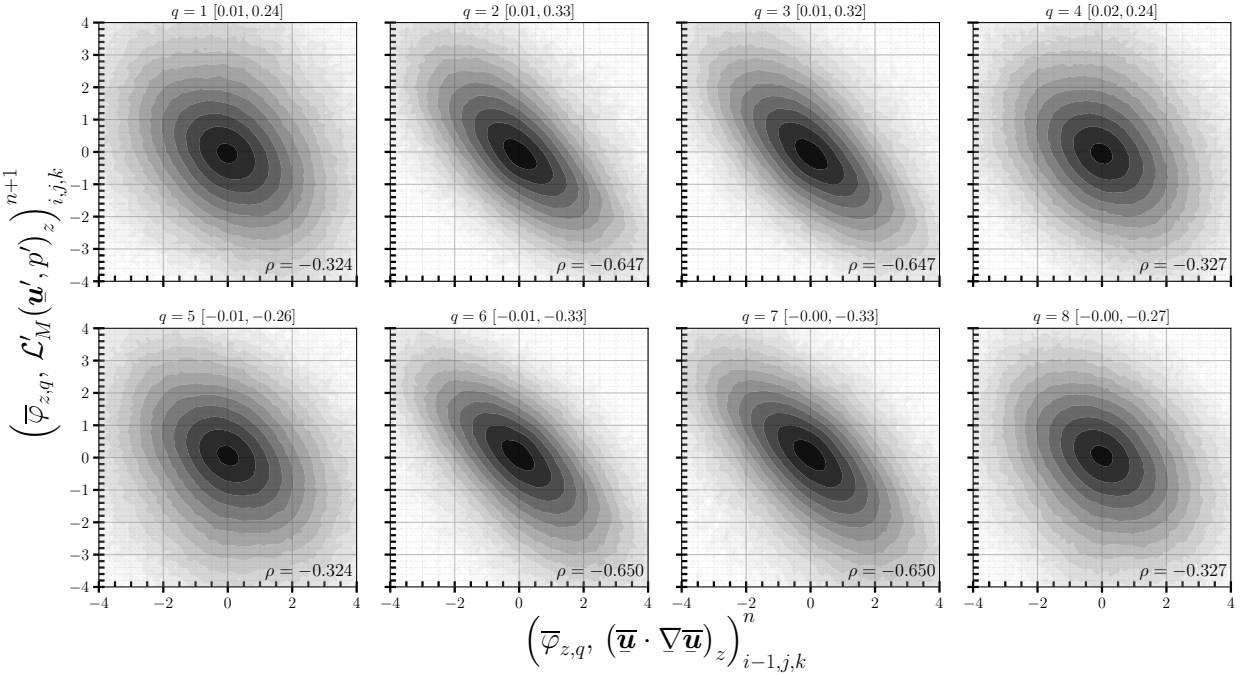


FIGURE A.64 Histograms of the t^n $(i-1, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

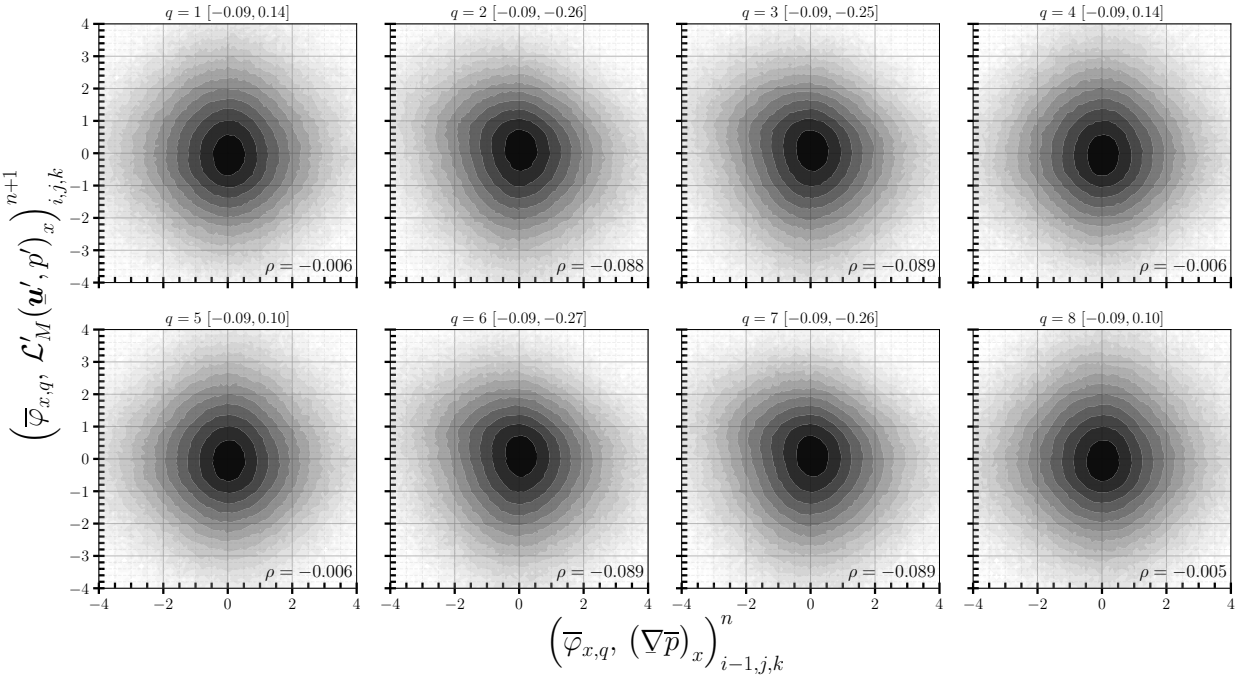


FIGURE A.65 Histograms of the t^n $(i-1, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

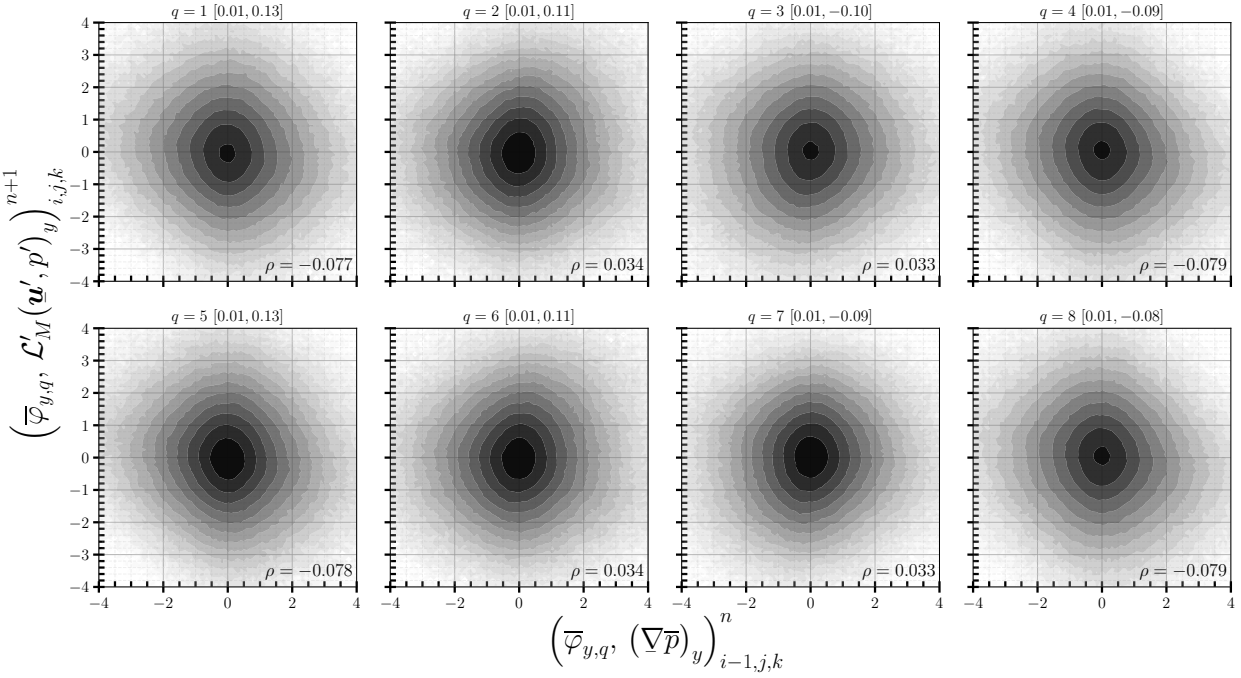


FIGURE A.66 Histograms of the t^n $(i-1, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

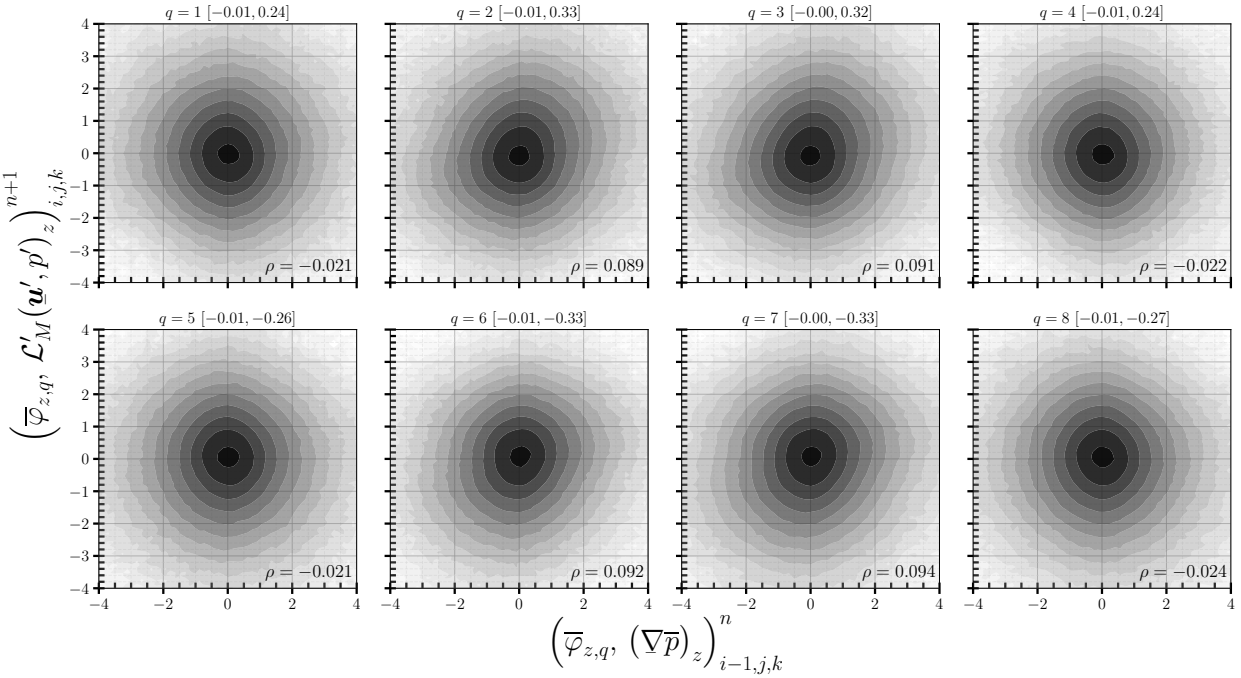


FIGURE A.67 Histograms of the t^n $(i-1, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

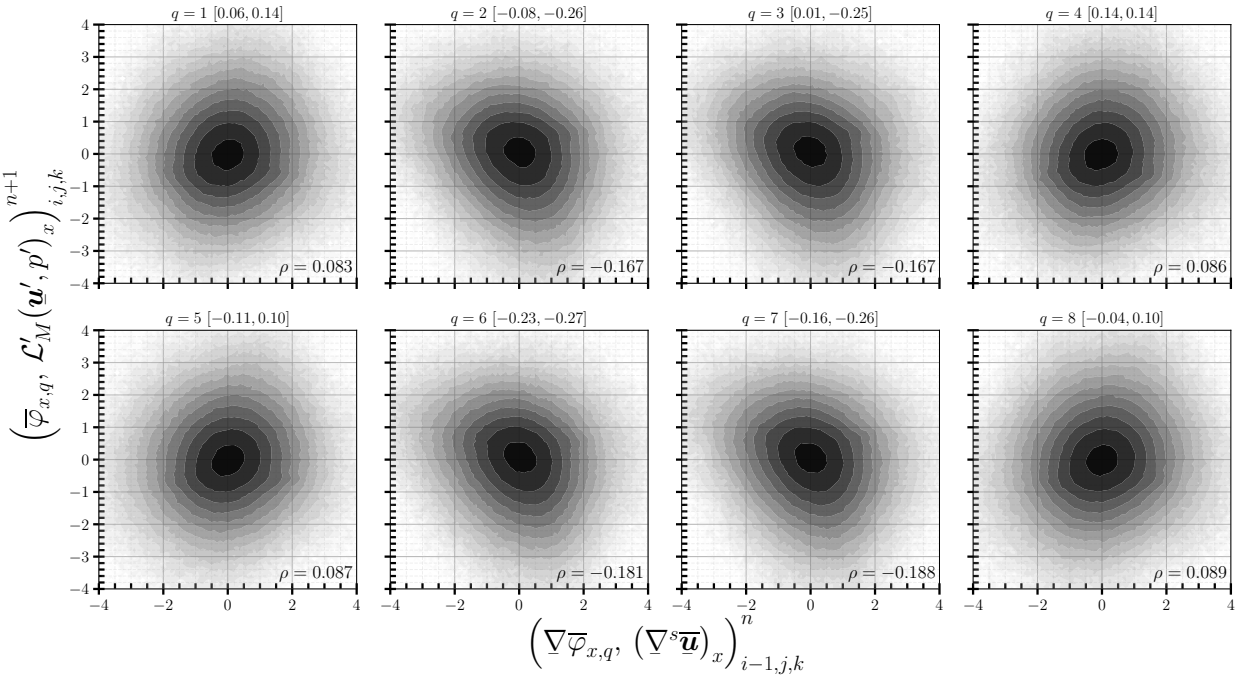


FIGURE A.68 Histograms of the t^n $(i-1, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

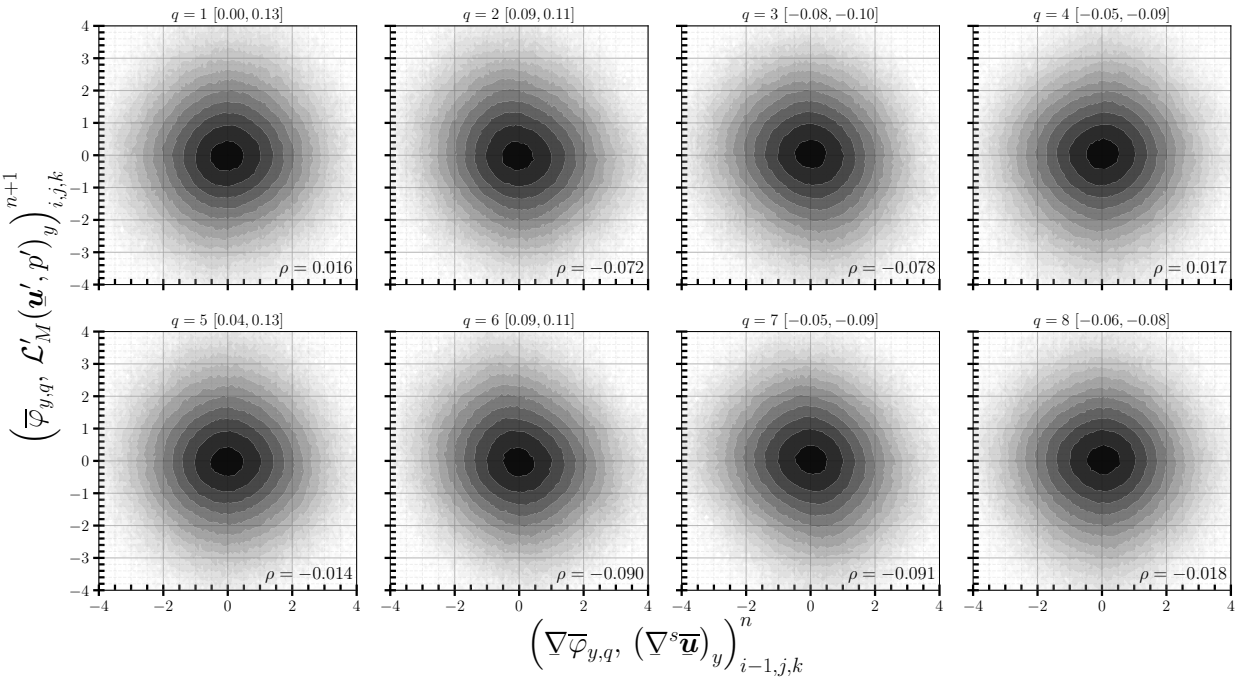


FIGURE A.69 Histograms of the t^n $(i-1, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

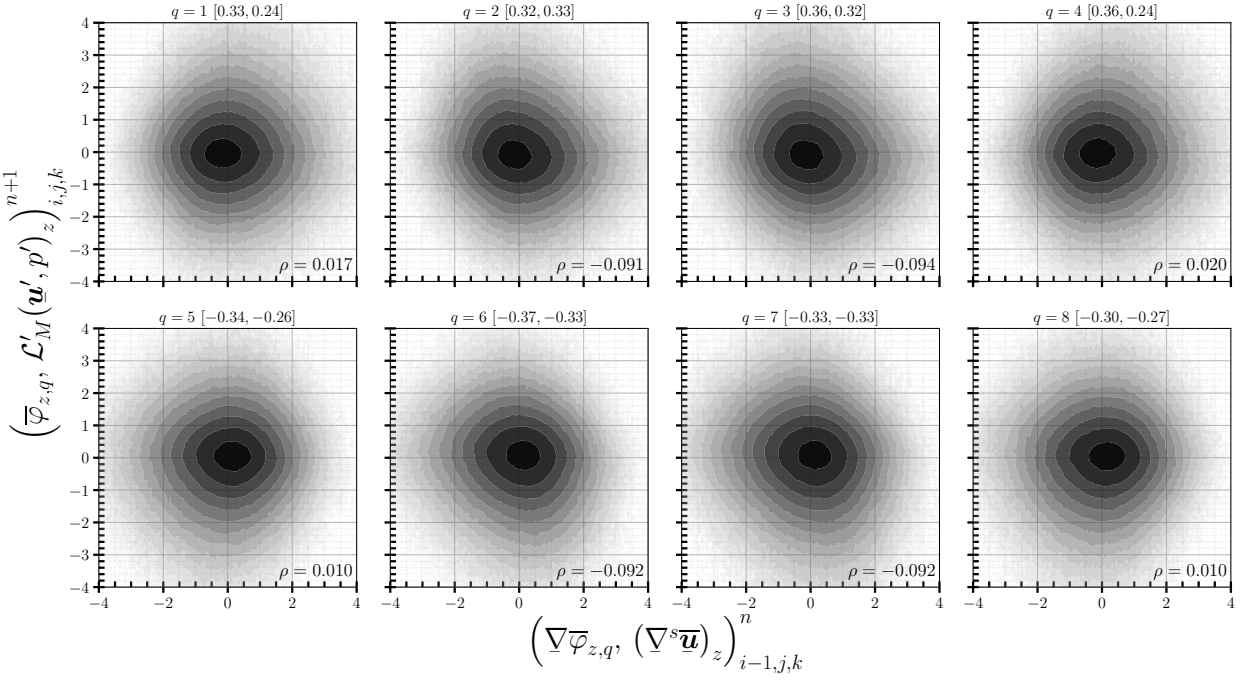


FIGURE A.70 Histograms of the t^n $(i-1, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.3 $(i-2, j, k)$ Histograms

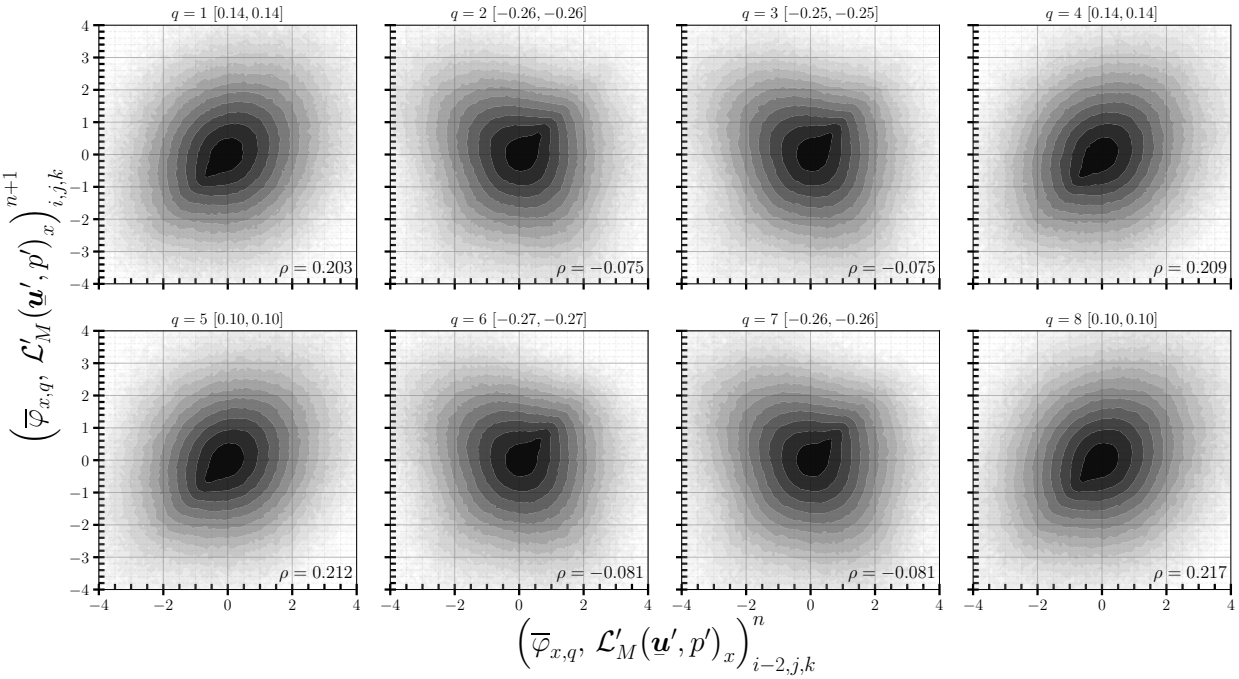


Figure A.71 Histograms of the t^n $(i-2, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

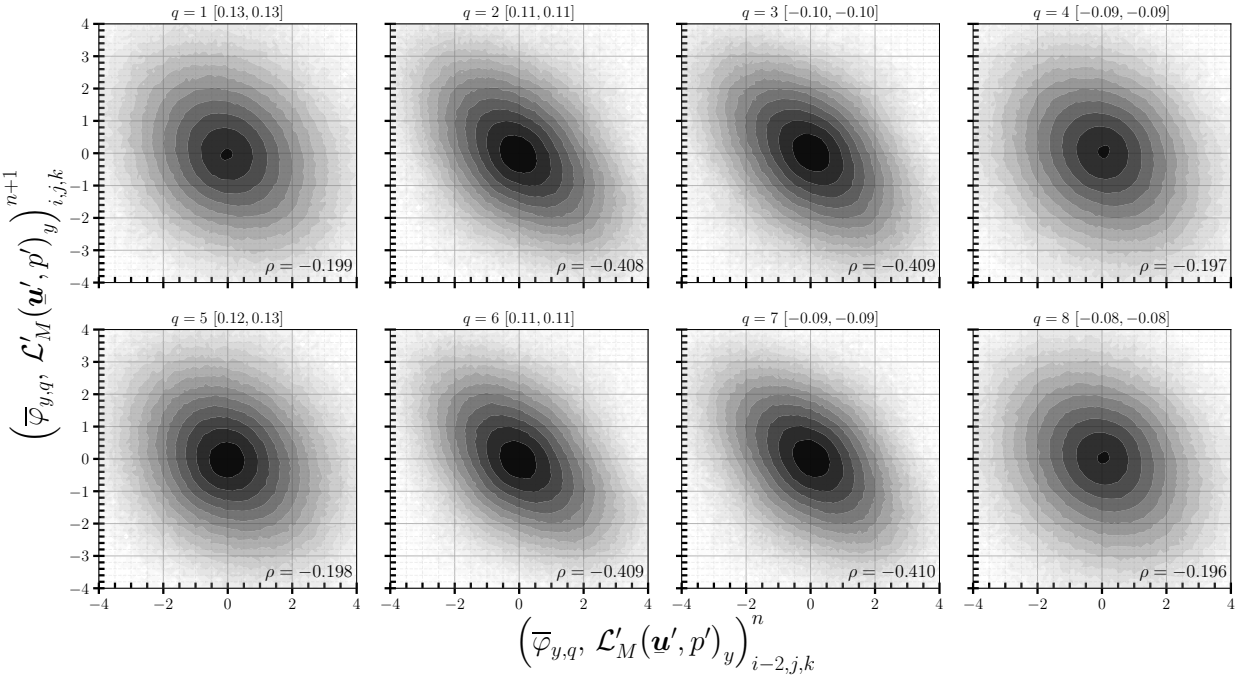


FIGURE A.72 Histograms of the t^n $(i-2, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

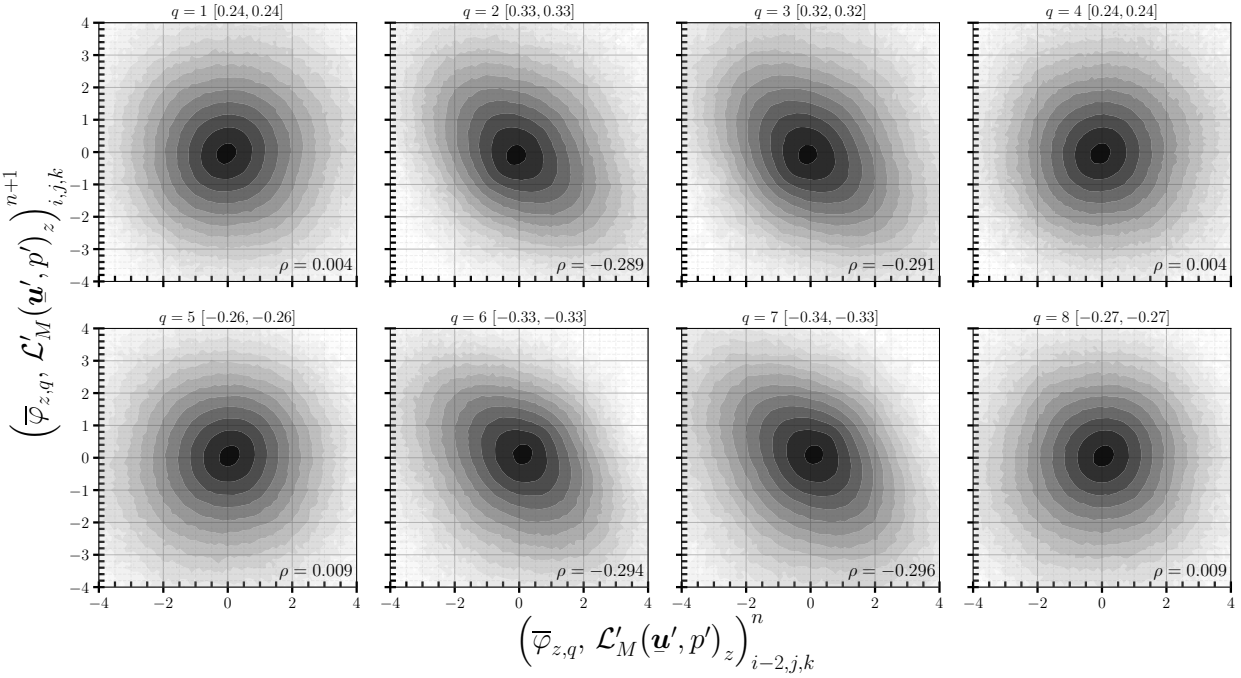


FIGURE A.73 Histograms of the t^n $(i-2, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

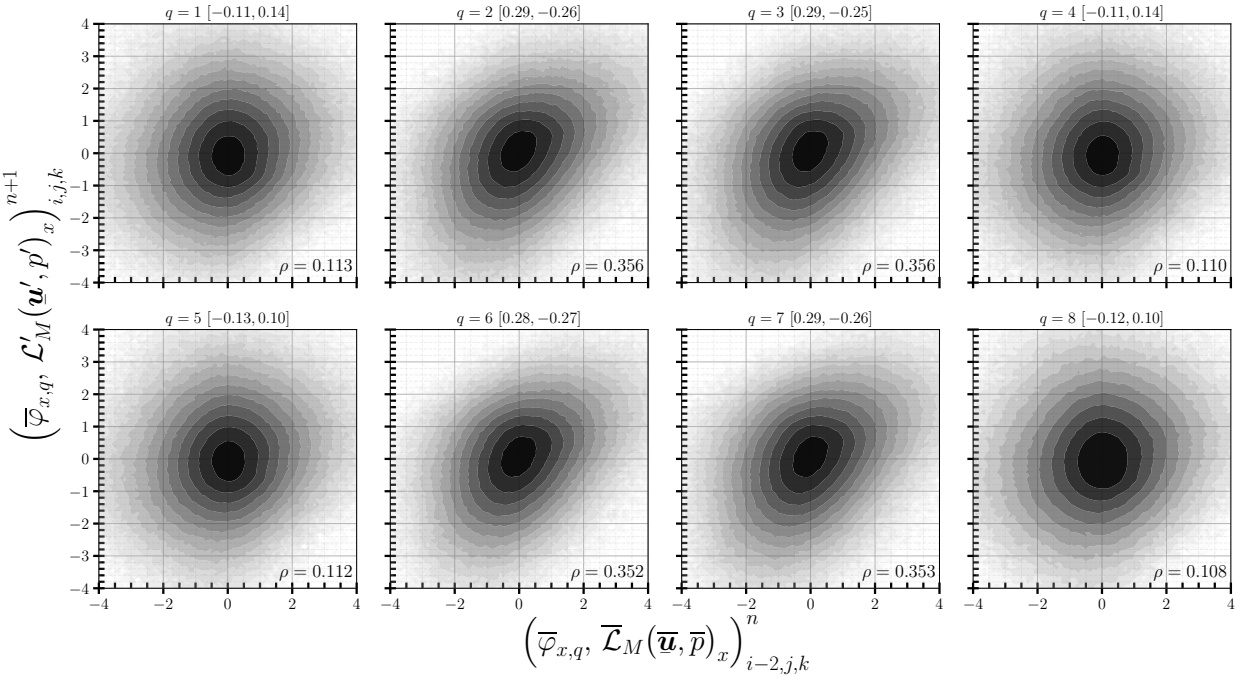


FIGURE A.74 Histograms of the t^n $(i-2, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

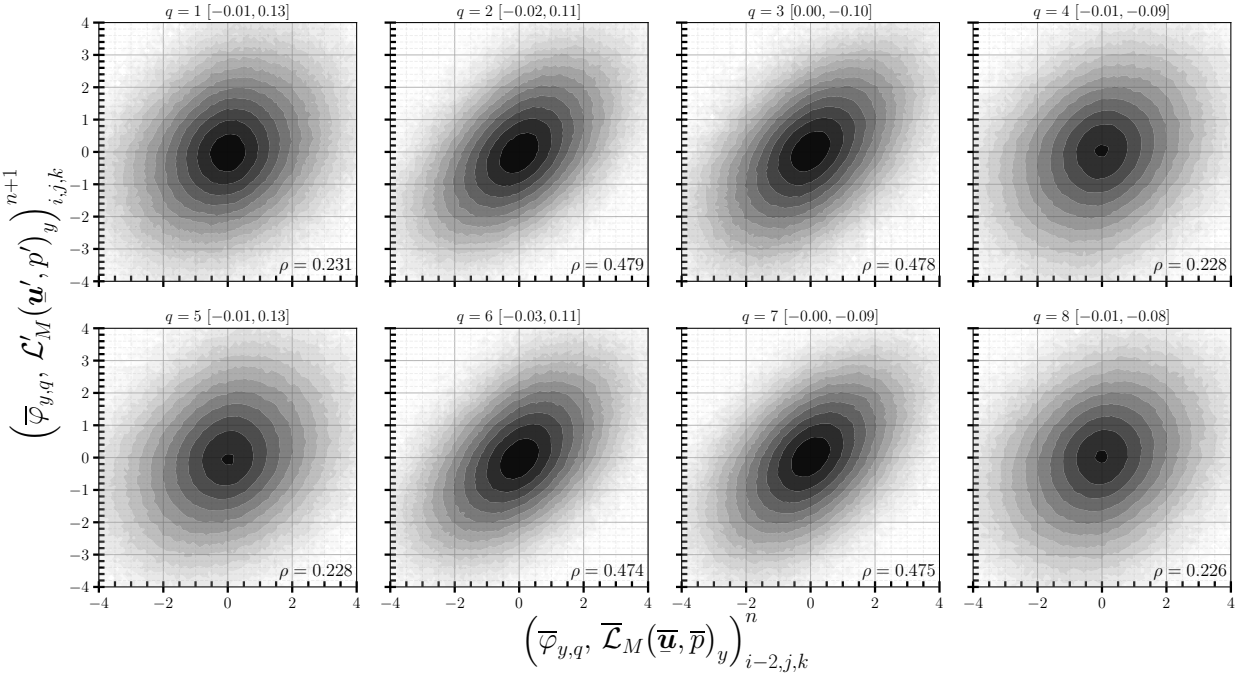


FIGURE A.75 Histograms of the t^n $(i-2, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

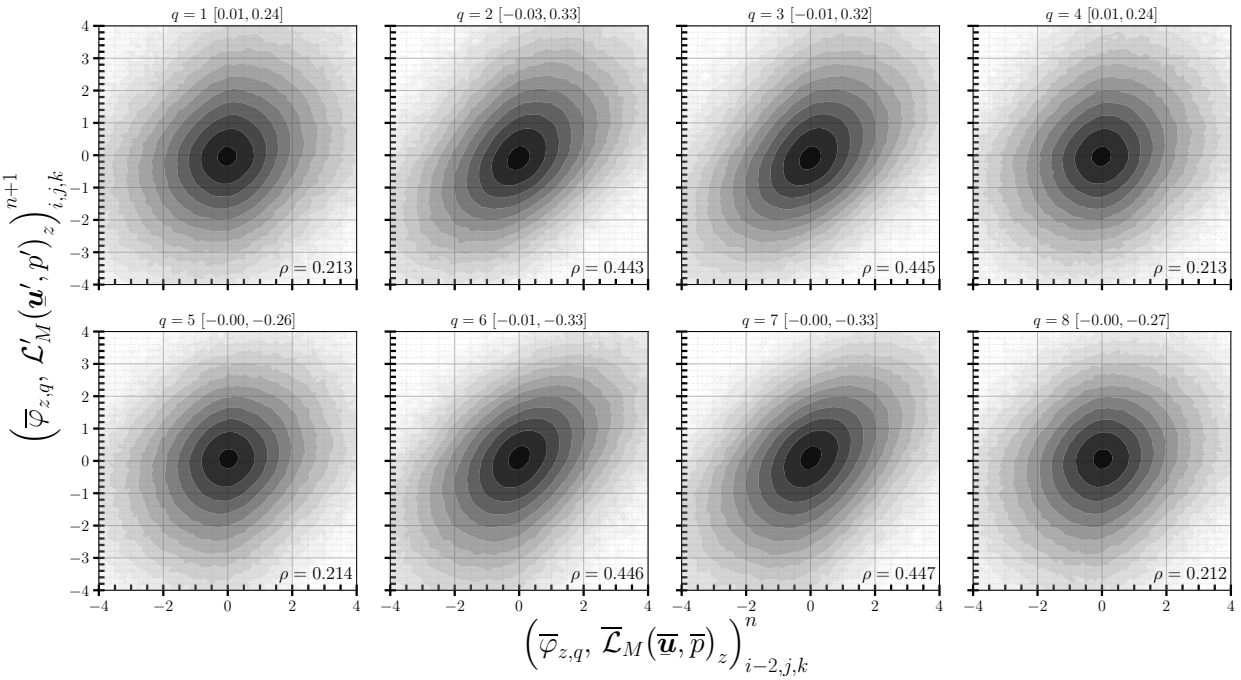


FIGURE A.76 Histograms of the t^n $(i-2, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

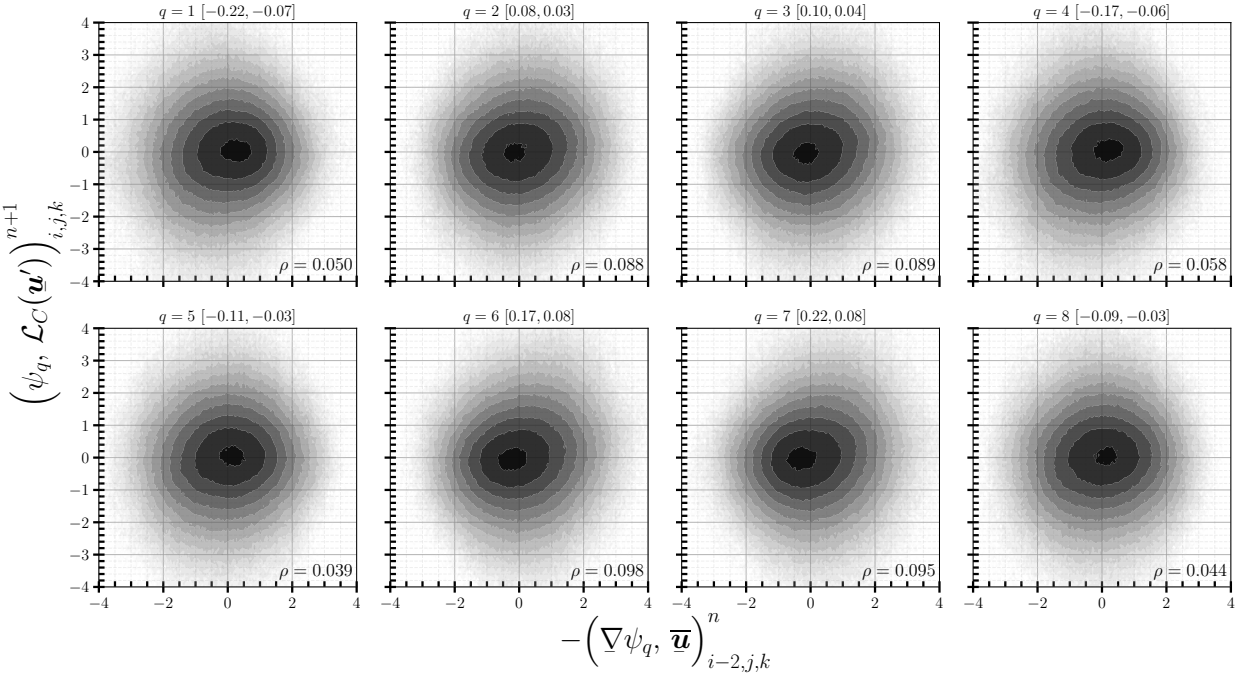


FIGURE A.77 Histograms of the t^n $(i-2, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

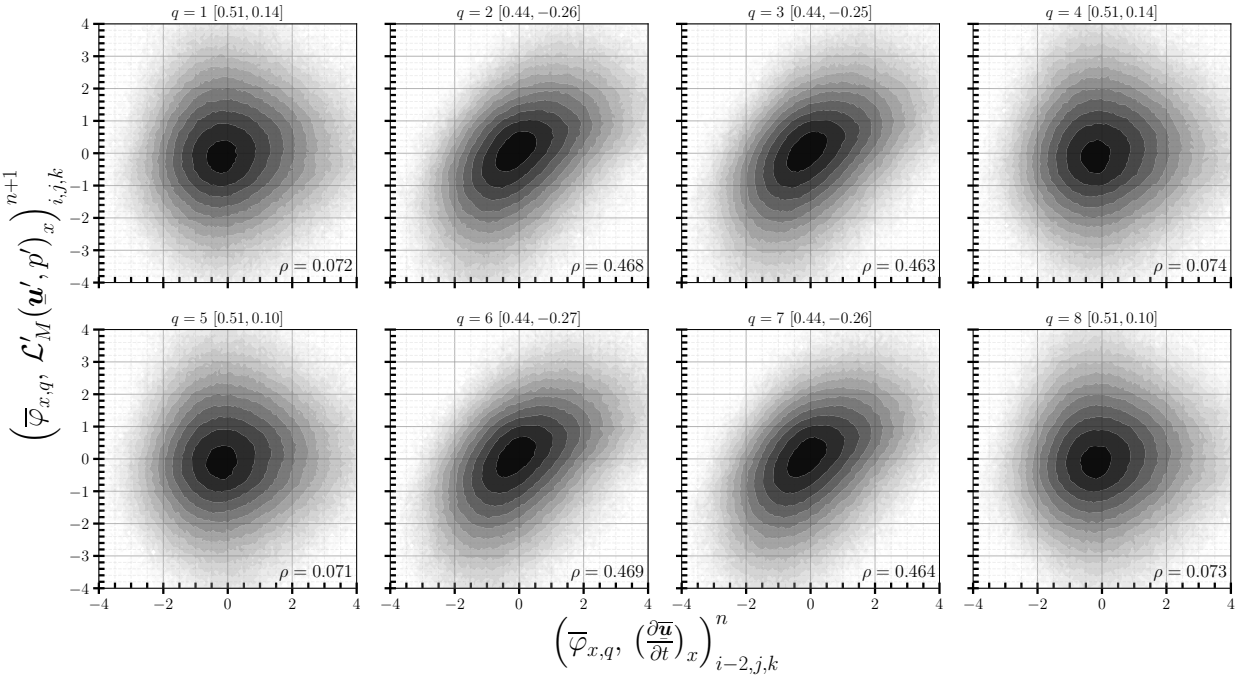


FIGURE A.78 Histograms of the t^n $(i-2, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

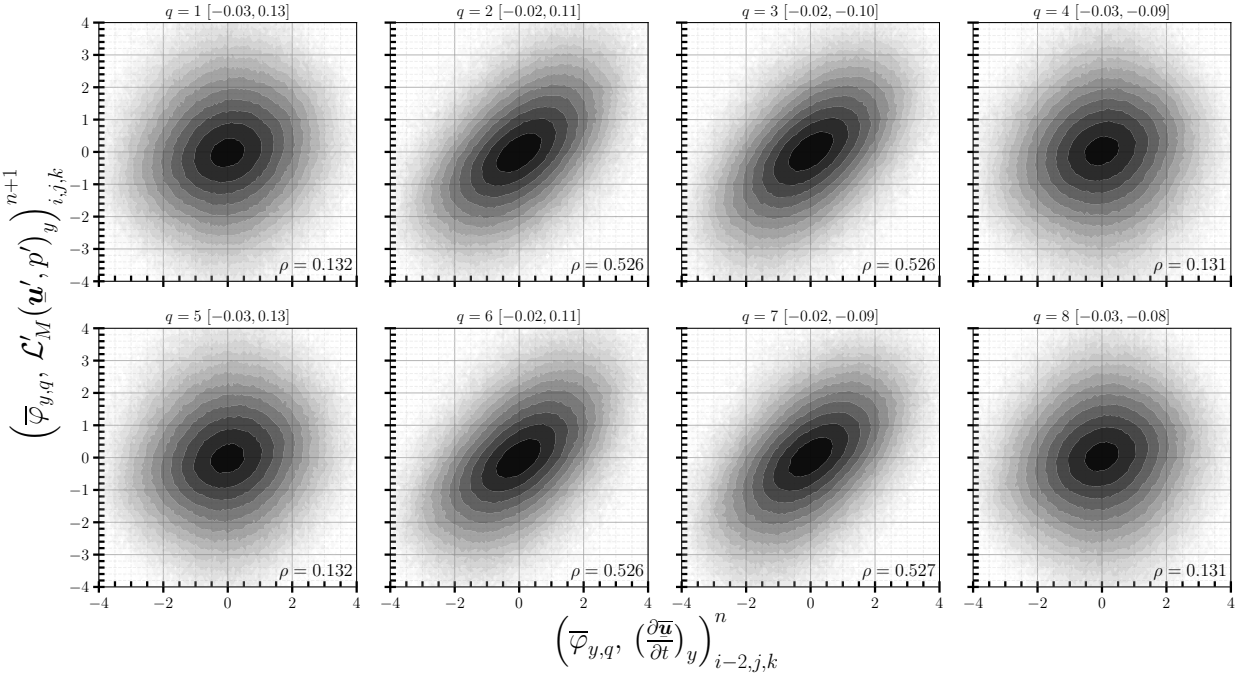


FIGURE A.79 Histograms of the t^n $(i-2, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

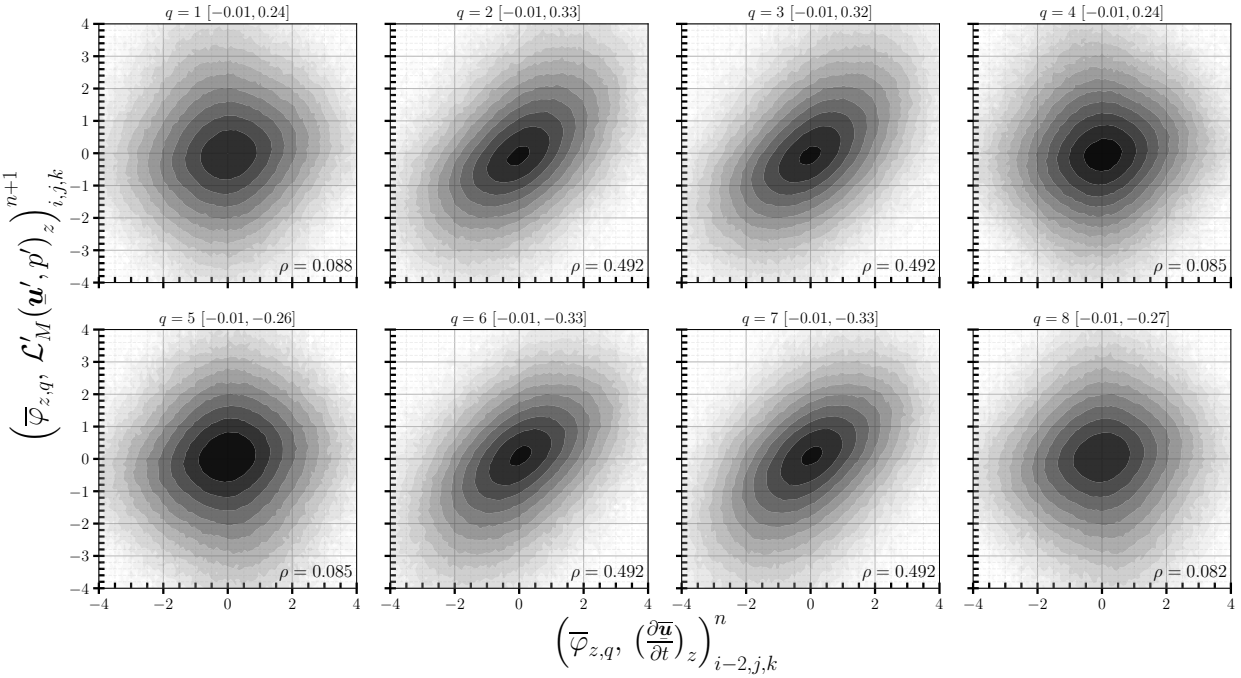


FIGURE A.80 Histograms of the t^n $(i-2, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

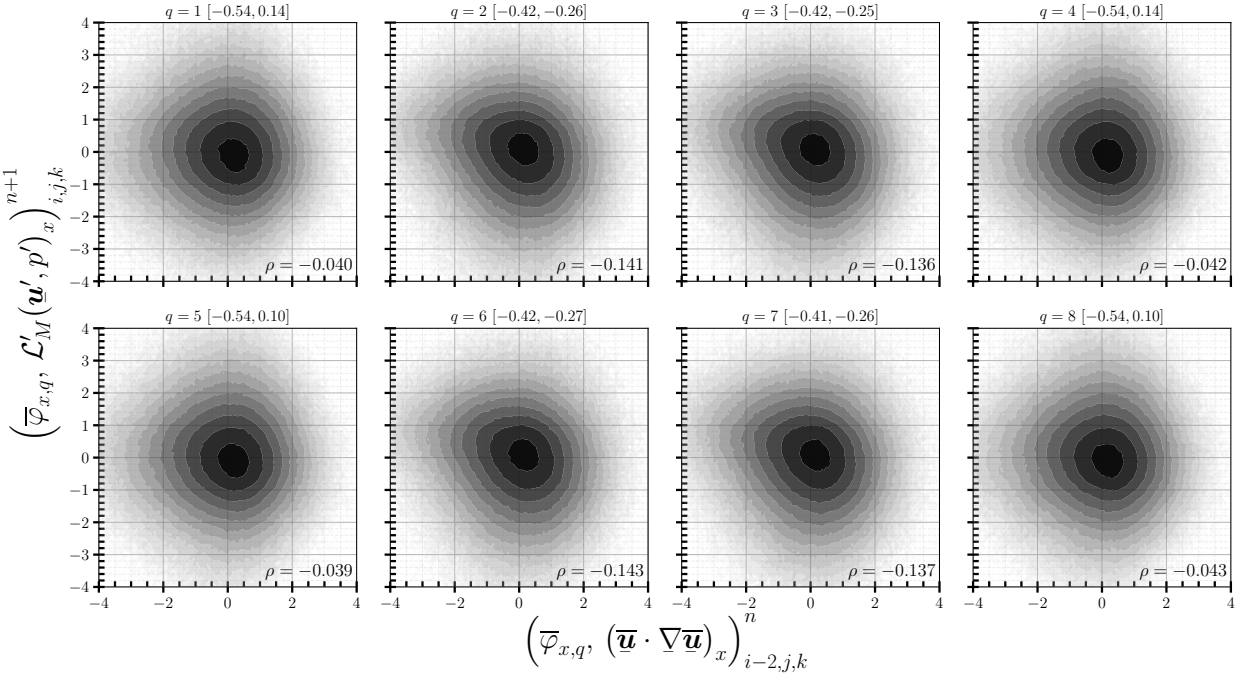


FIGURE A.81 Histograms of the t^n $(i-2, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

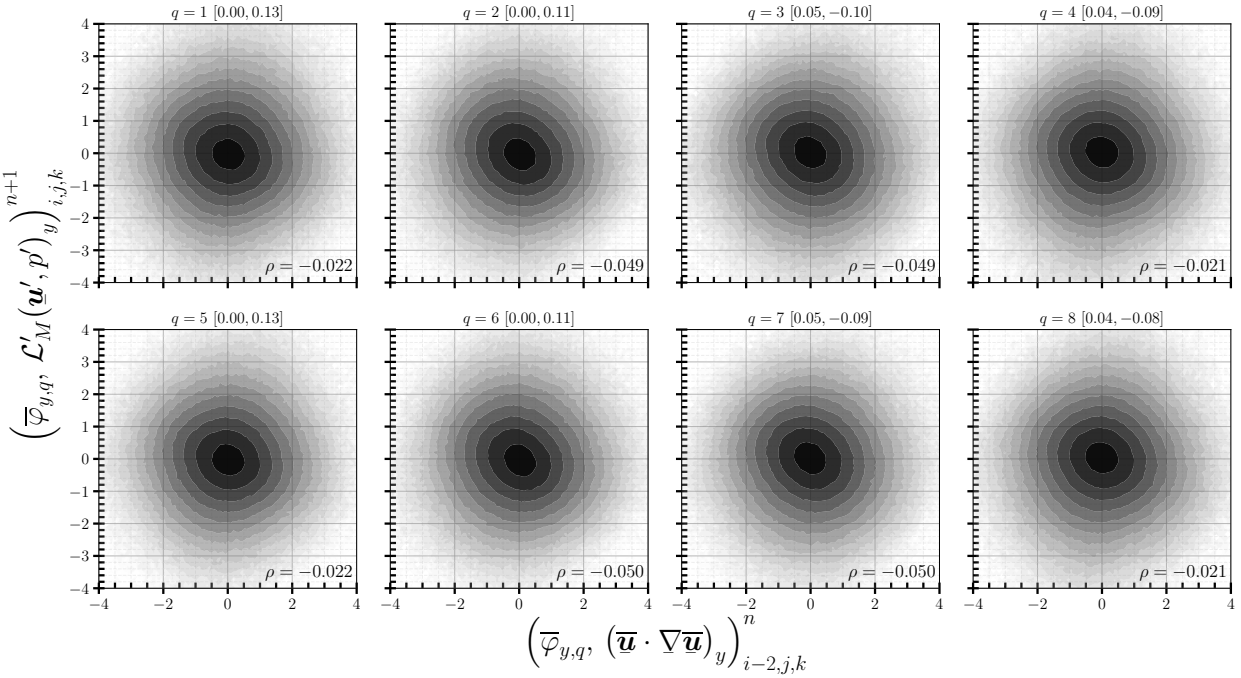


FIGURE A.82 Histograms of the t^n $(i-2, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

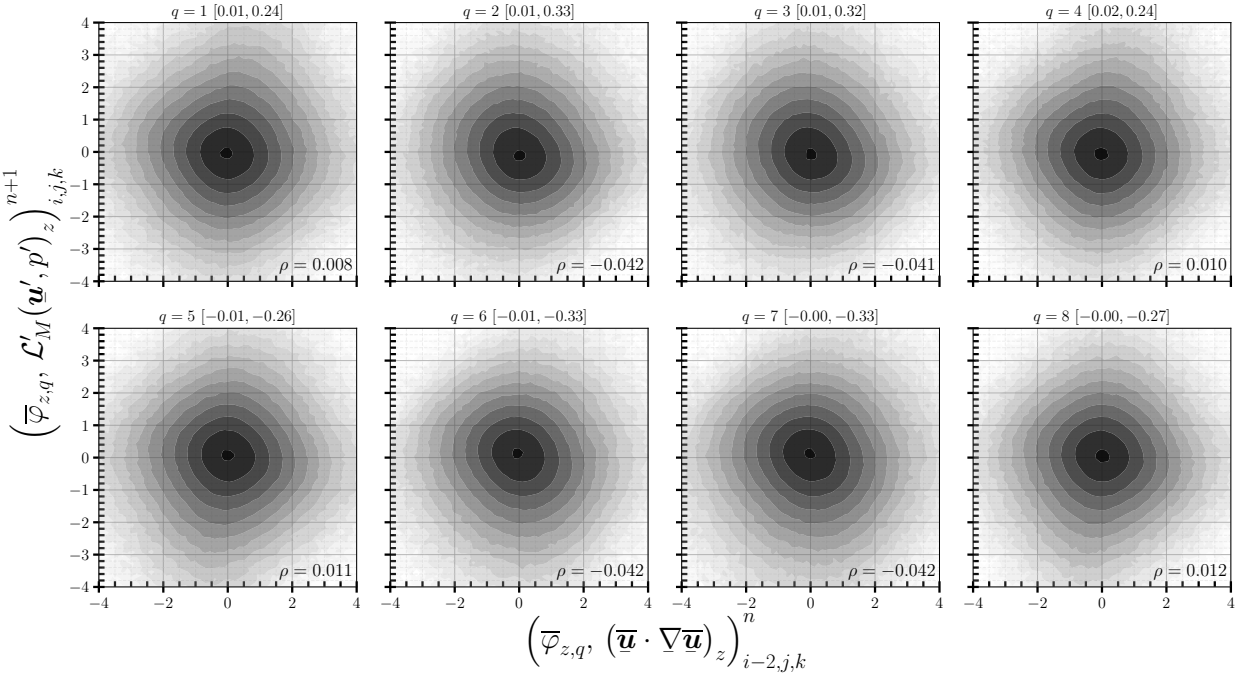


FIGURE A.83 Histograms of the t^n $(i-2, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

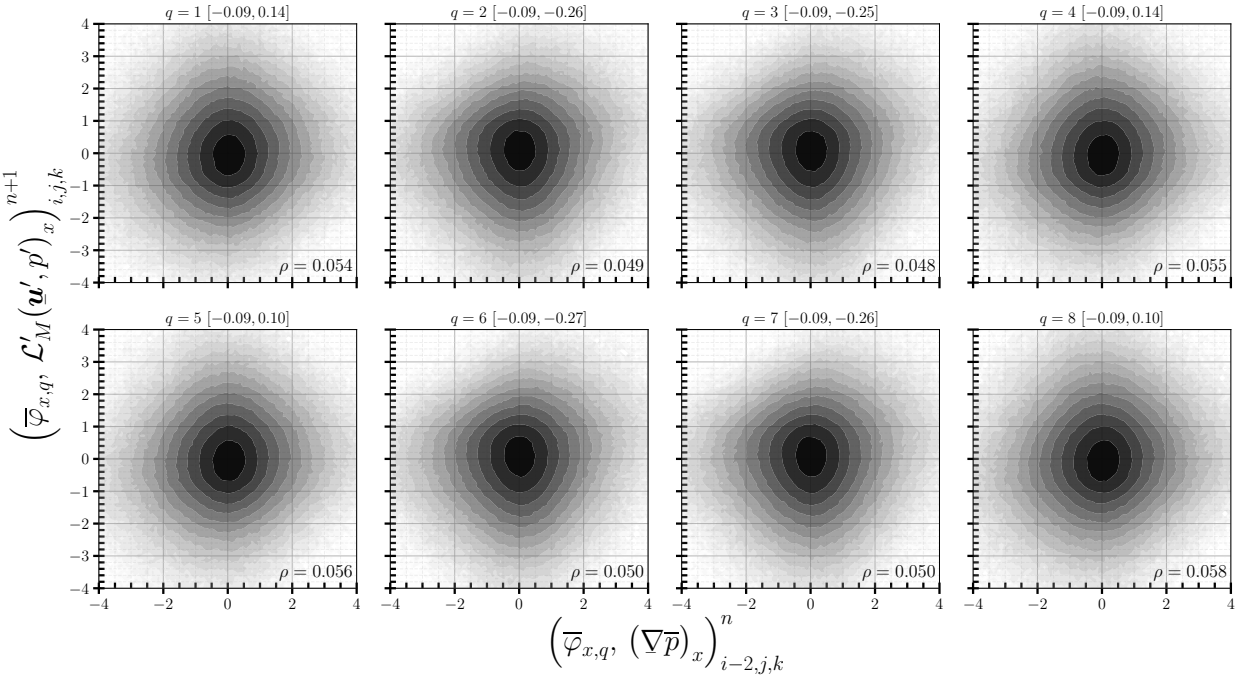


FIGURE A.84 Histograms of the t^n $(i-2, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

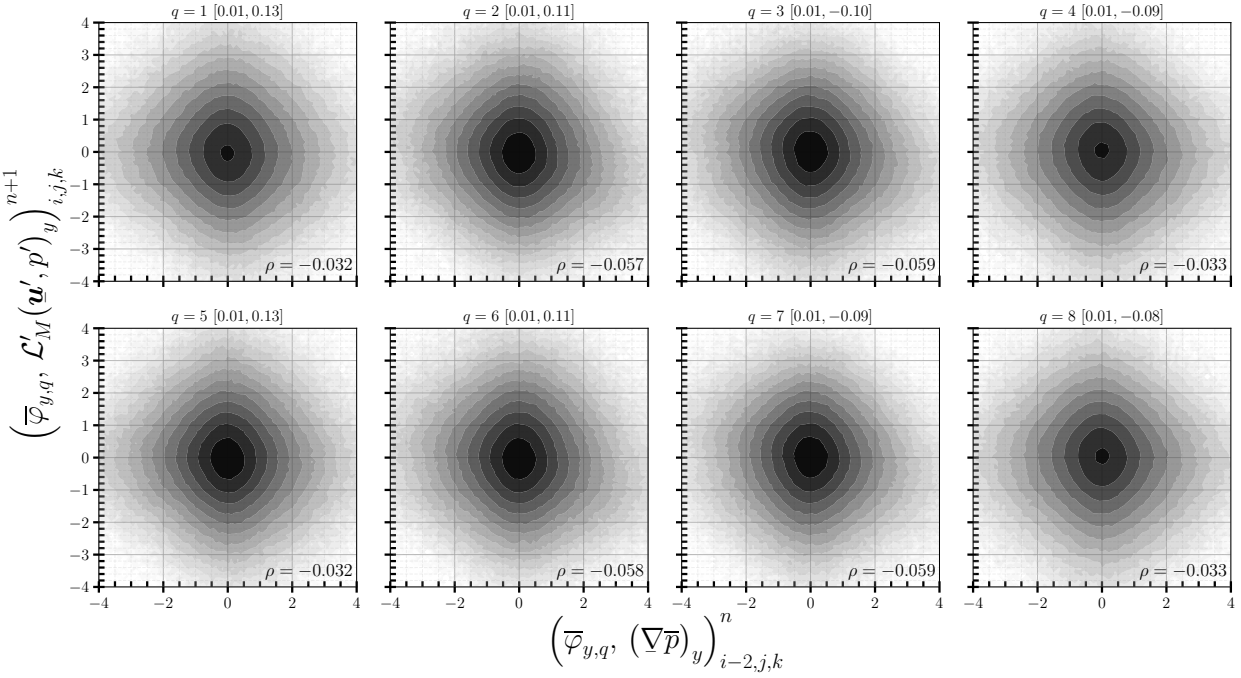


FIGURE A.85 Histograms of the t^n $(i-2, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

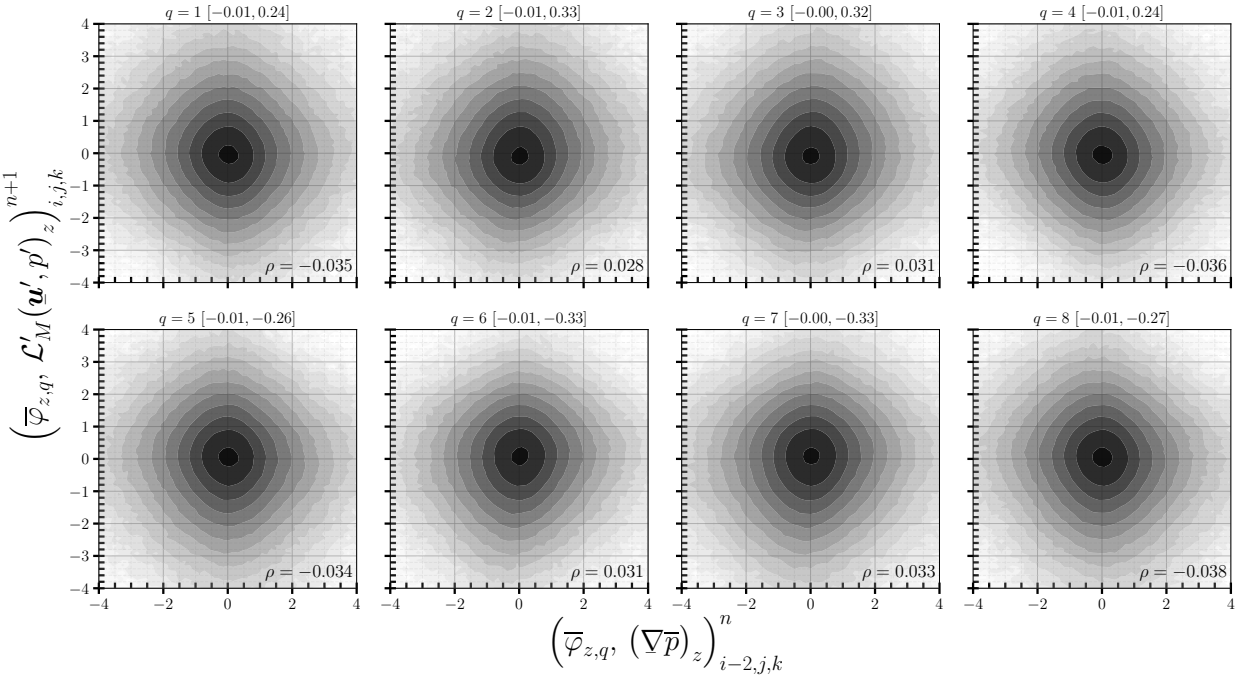


FIGURE A.86 Histograms of the t^n $(i-2, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

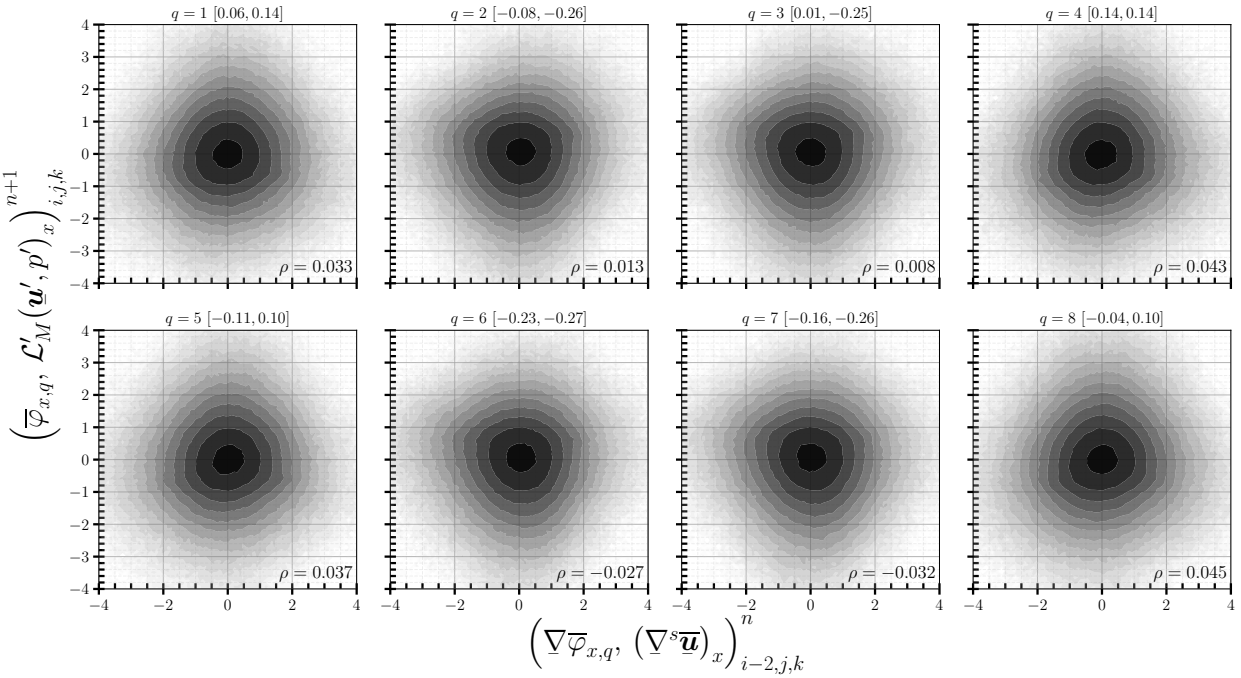


FIGURE A.87 Histograms of the t^n $(i-2, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

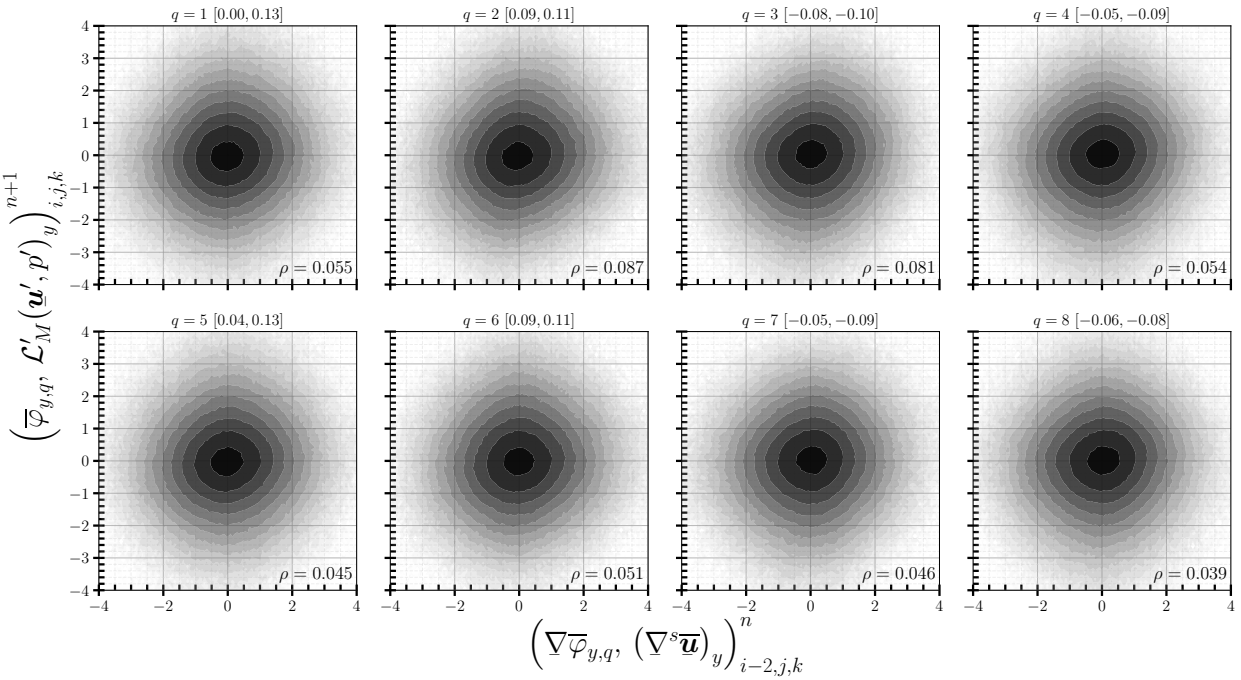


FIGURE A.88 Histograms of the t^n $(i-2, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

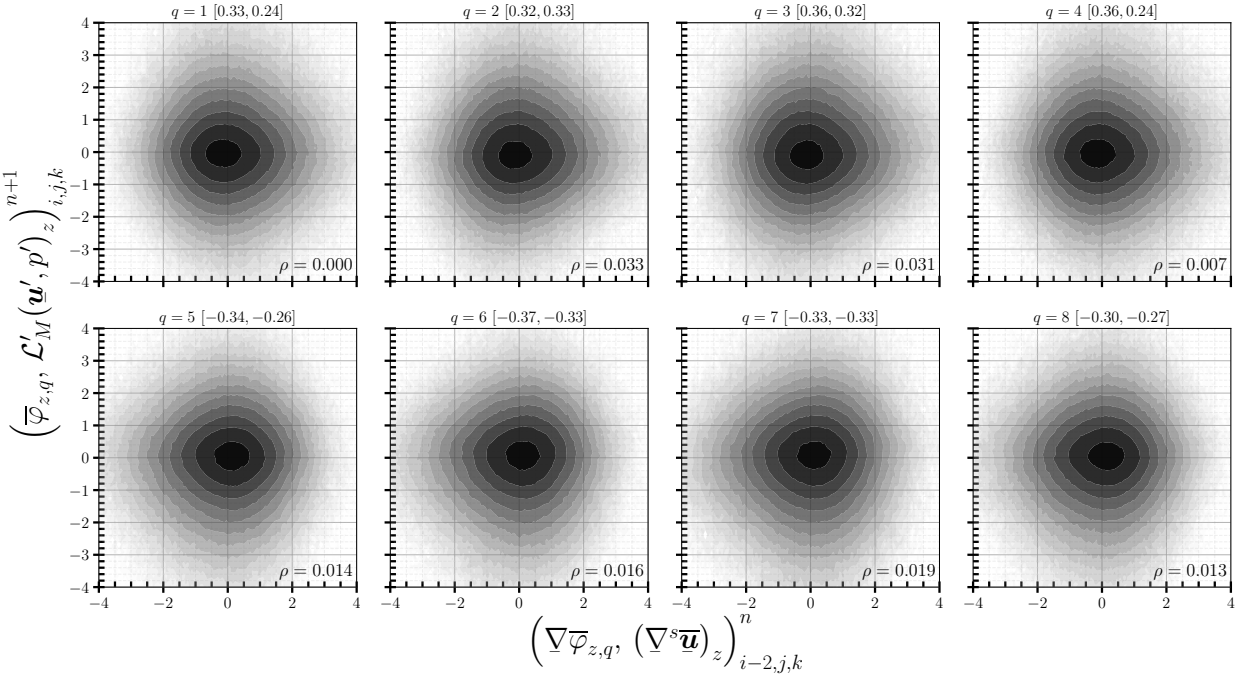


FIGURE A.89 Histograms of the t^n $(i-2, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.4 $(i-3, j, k)$ Histograms

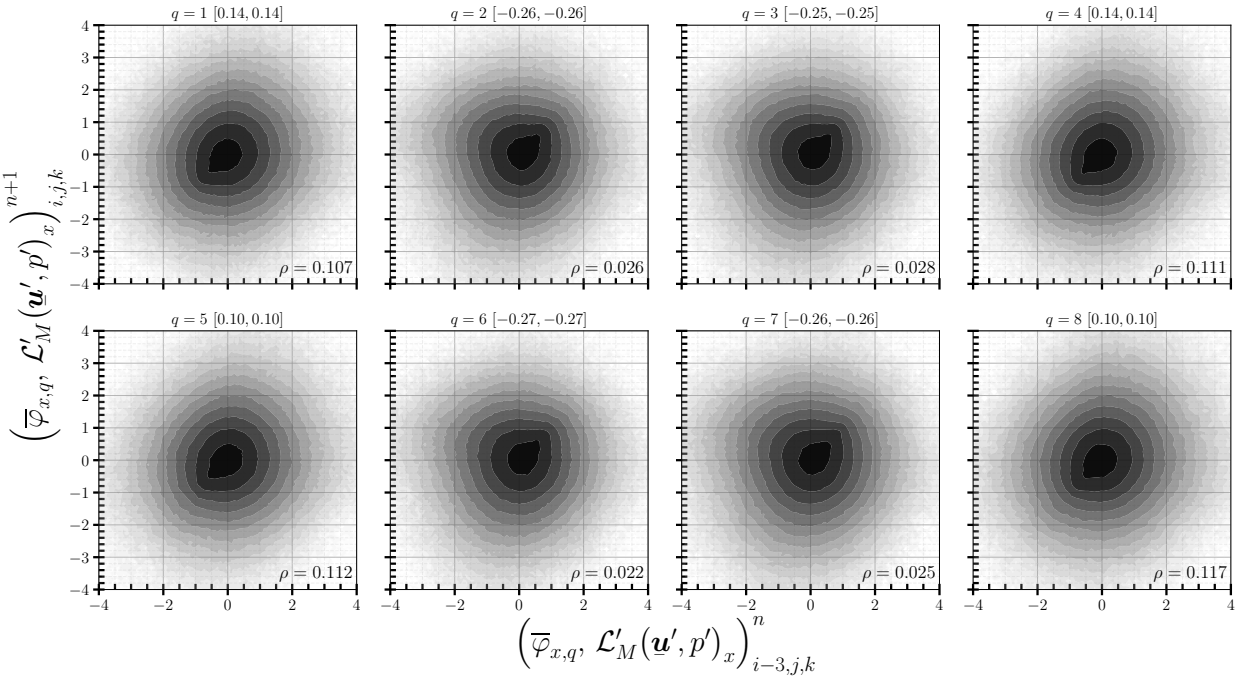


FIGURE A.90 Histograms of the t^n $(i-3, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

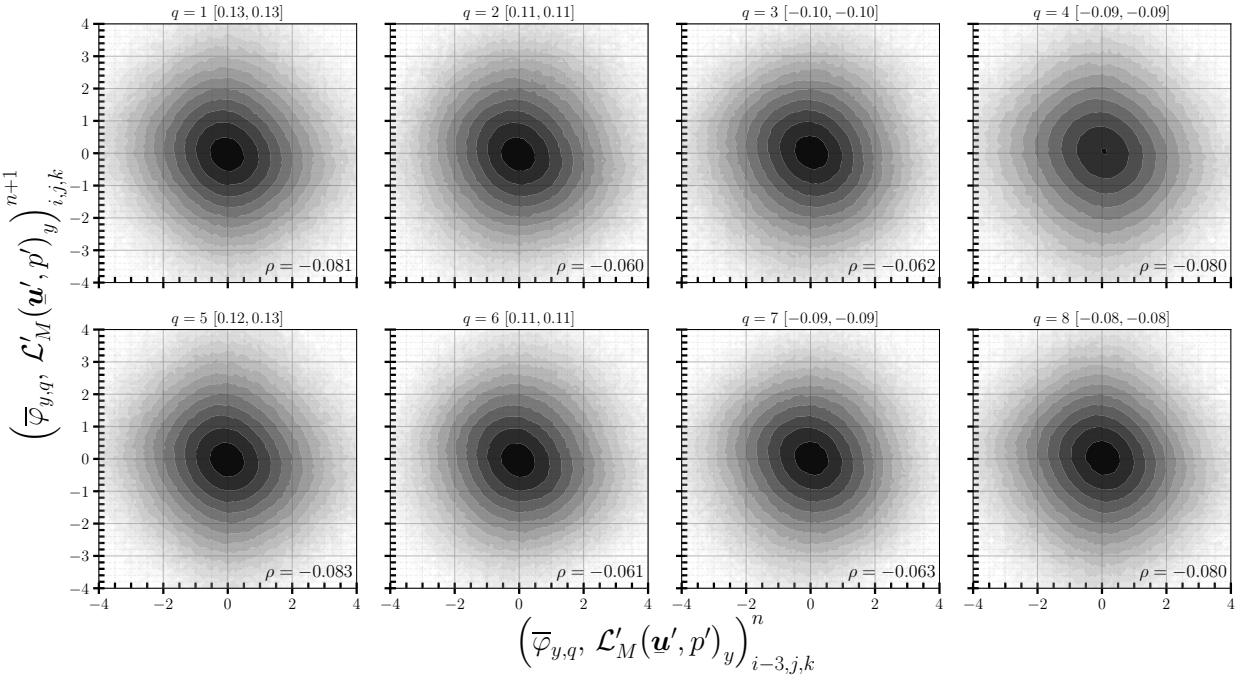


FIGURE A.91 Histograms of the t^n $(i-3, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

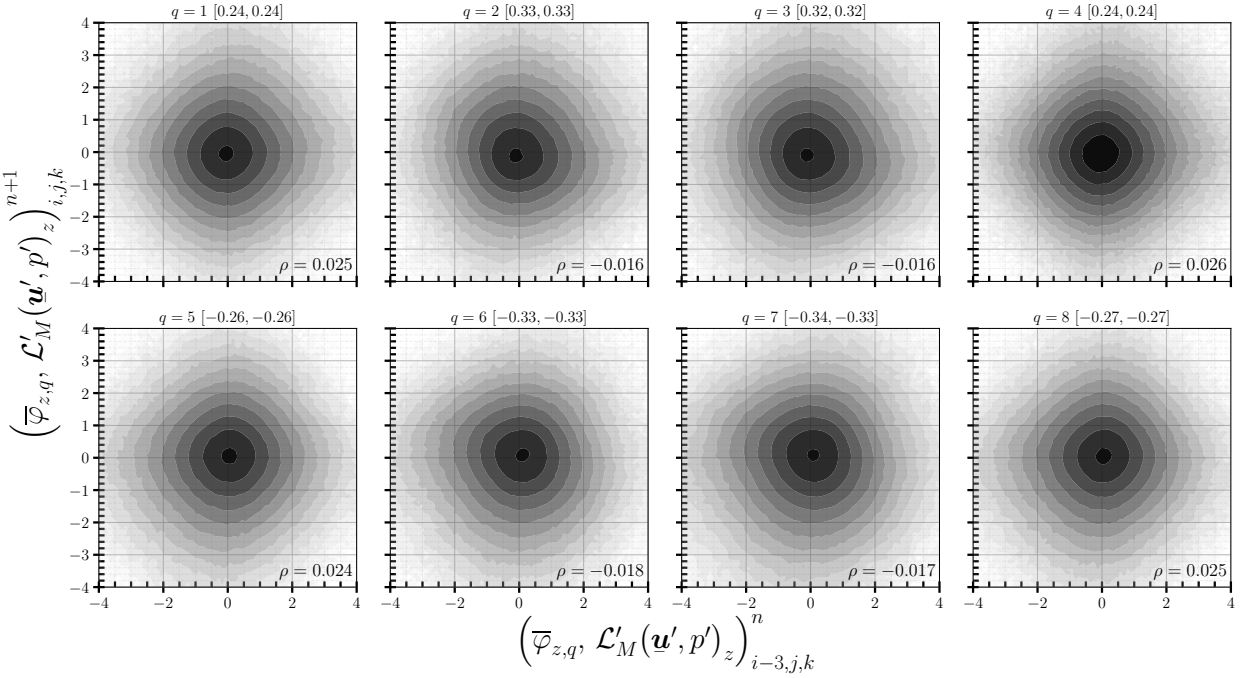


FIGURE A.92 Histograms of the t^n $(i-3, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

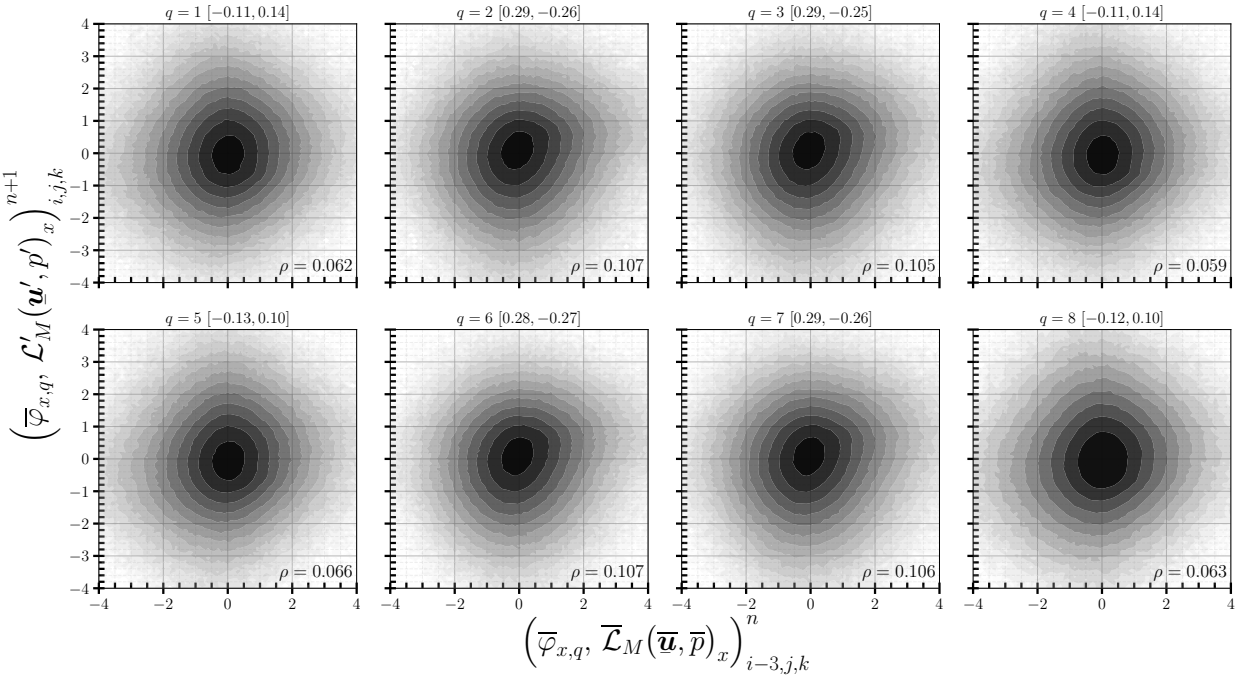


FIGURE A.93 Histograms of the t^n $(i-3, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

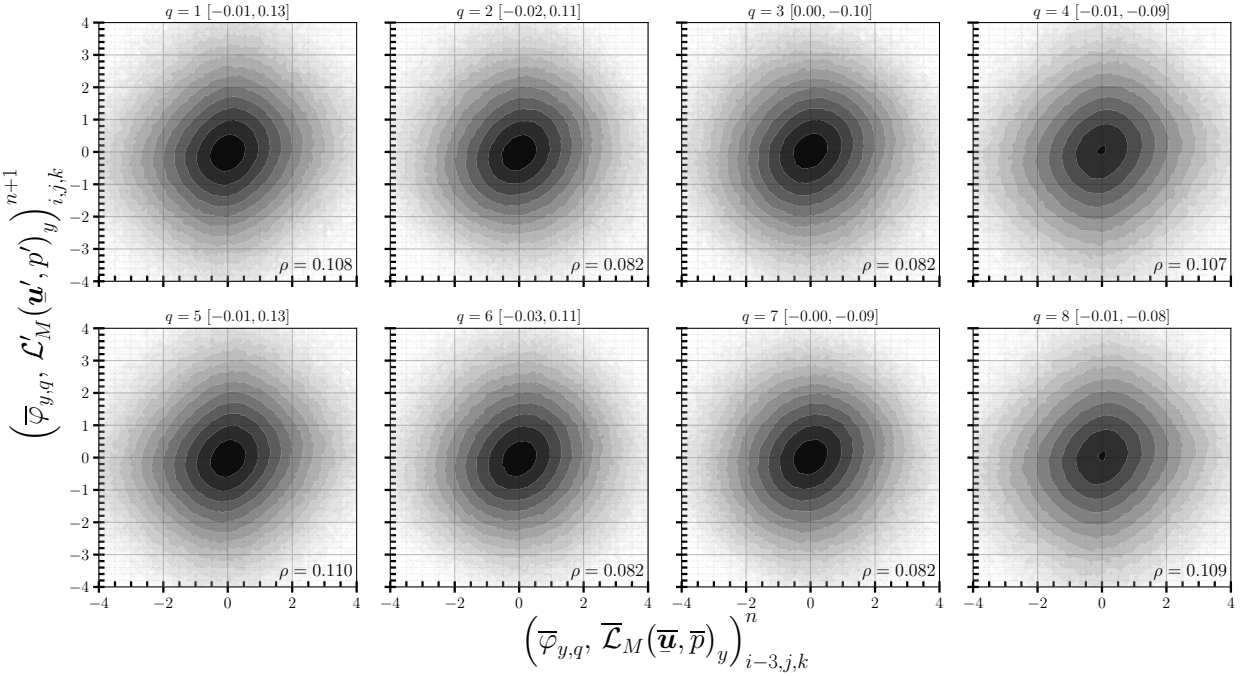


FIGURE A.94 Histograms of the t^n $(i-3, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

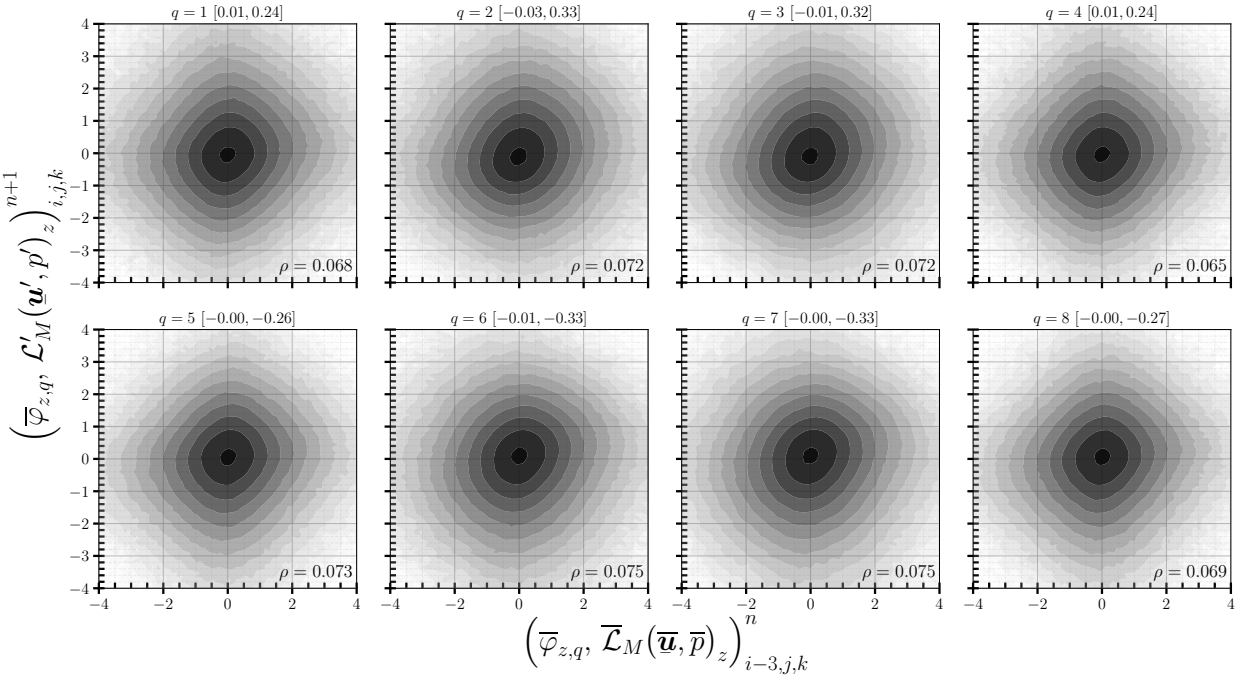


FIGURE A.95 Histograms of the t^n $(i-3, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

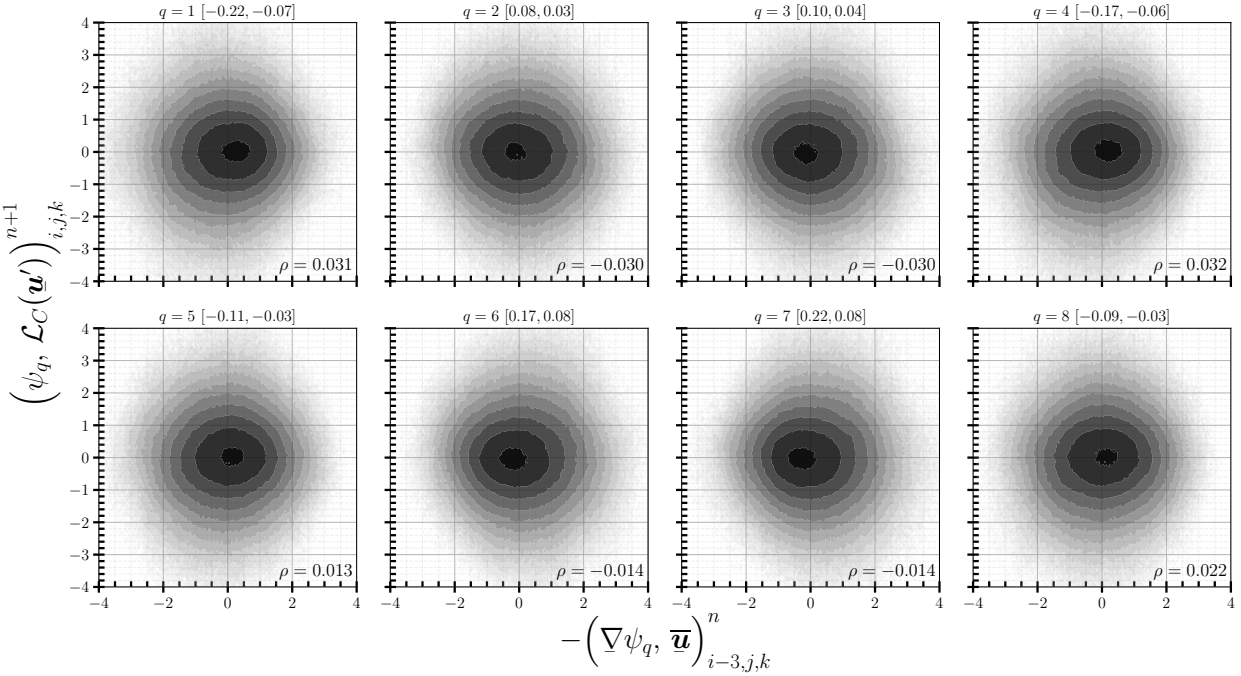


FIGURE A.96 Histograms of the t^n $(i-3, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

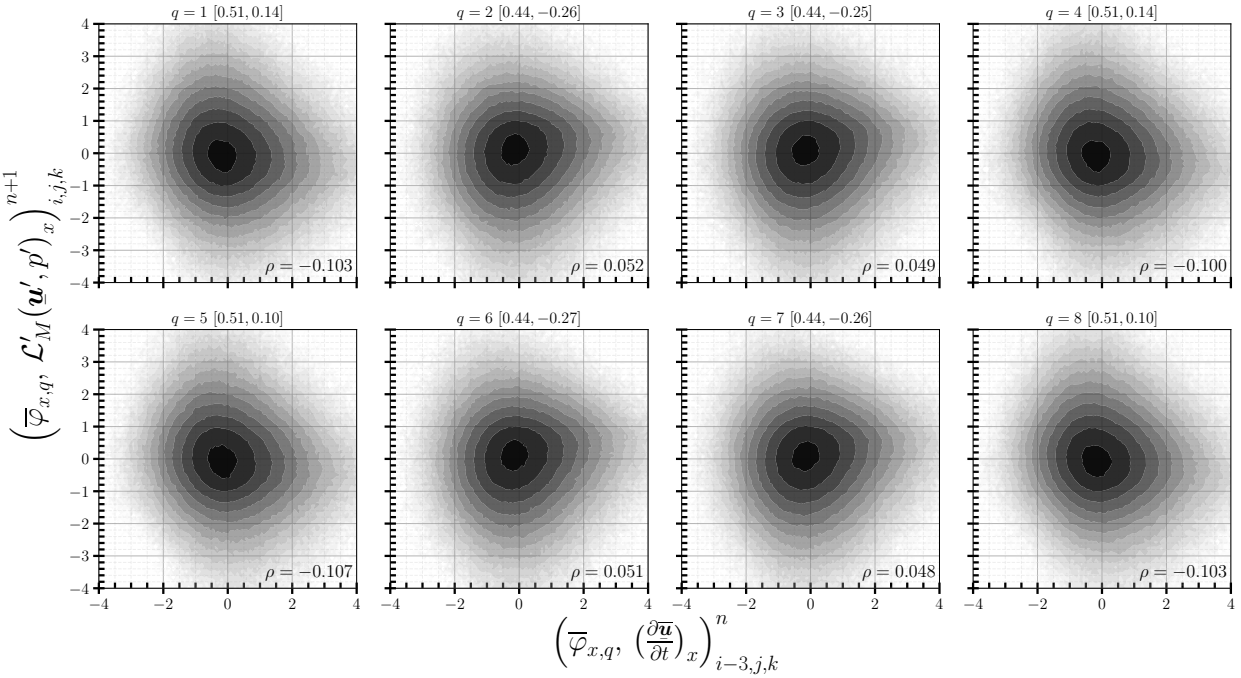


FIGURE A.97 Histograms of the t^n $(i-3, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

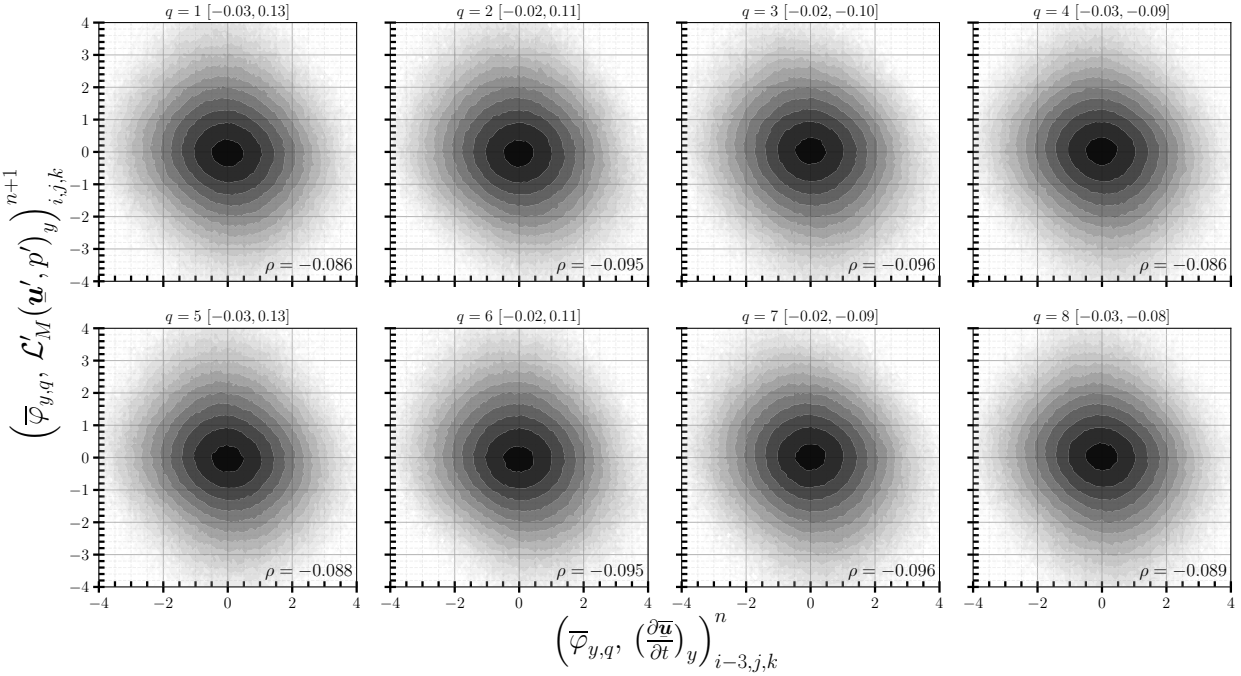


FIGURE A.98 Histograms of the t^n $(i-3, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

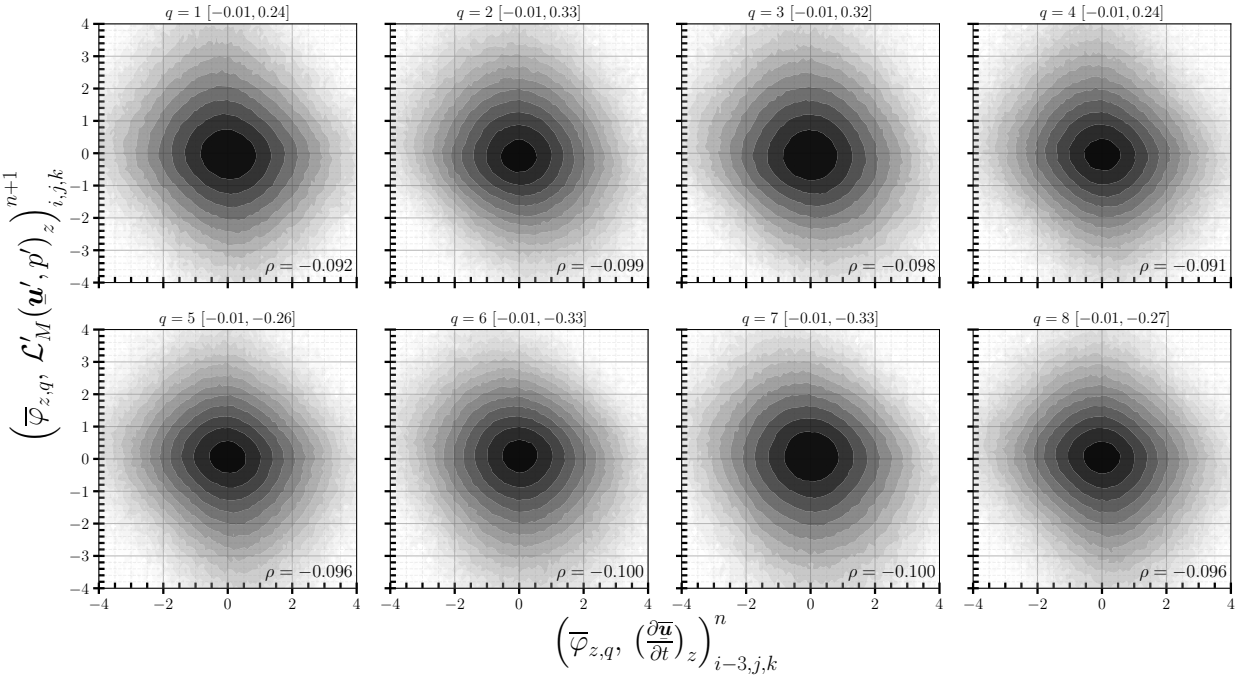


FIGURE A.99 Histograms of the t^n $(i-3, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

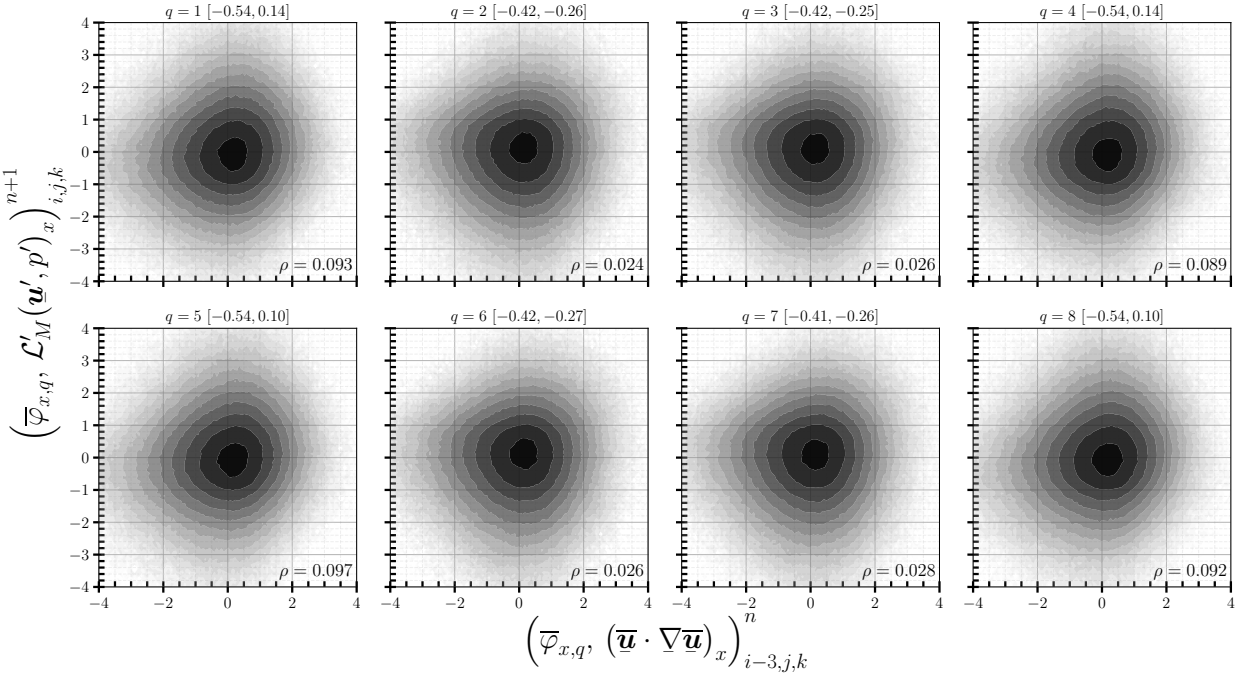


FIGURE A.100 Histograms of the t^n $(i-3, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

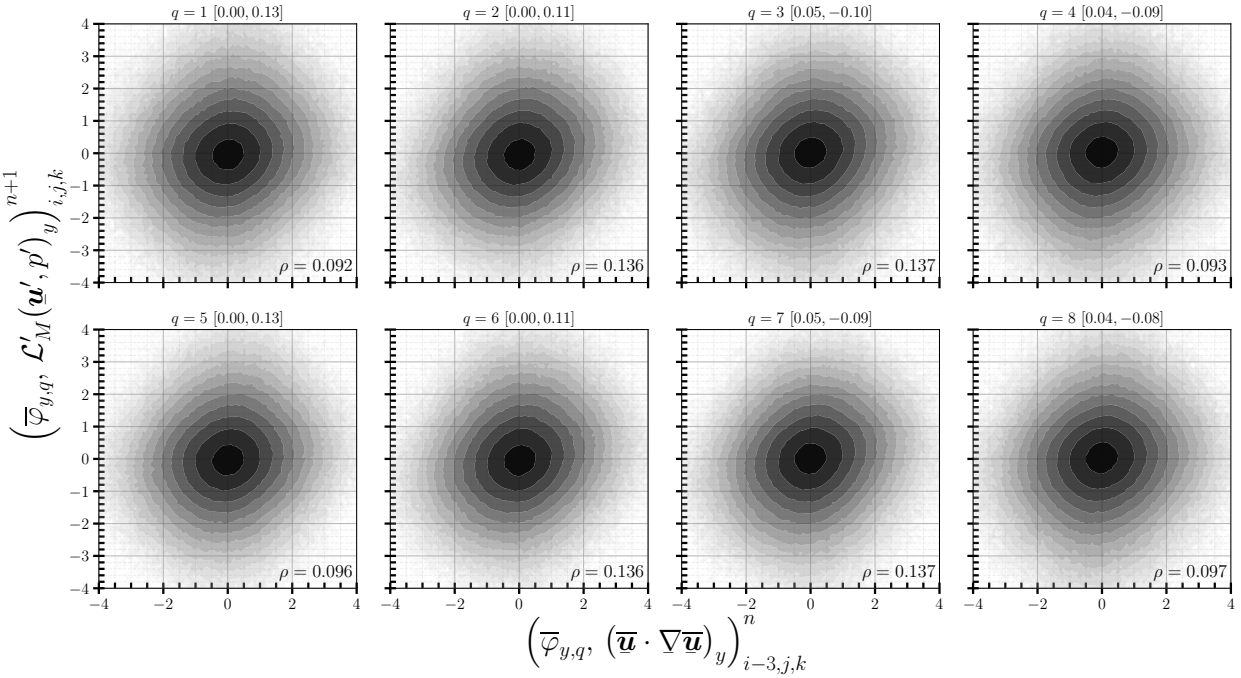


FIGURE A.101 Histograms of the t^n $(i-3, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

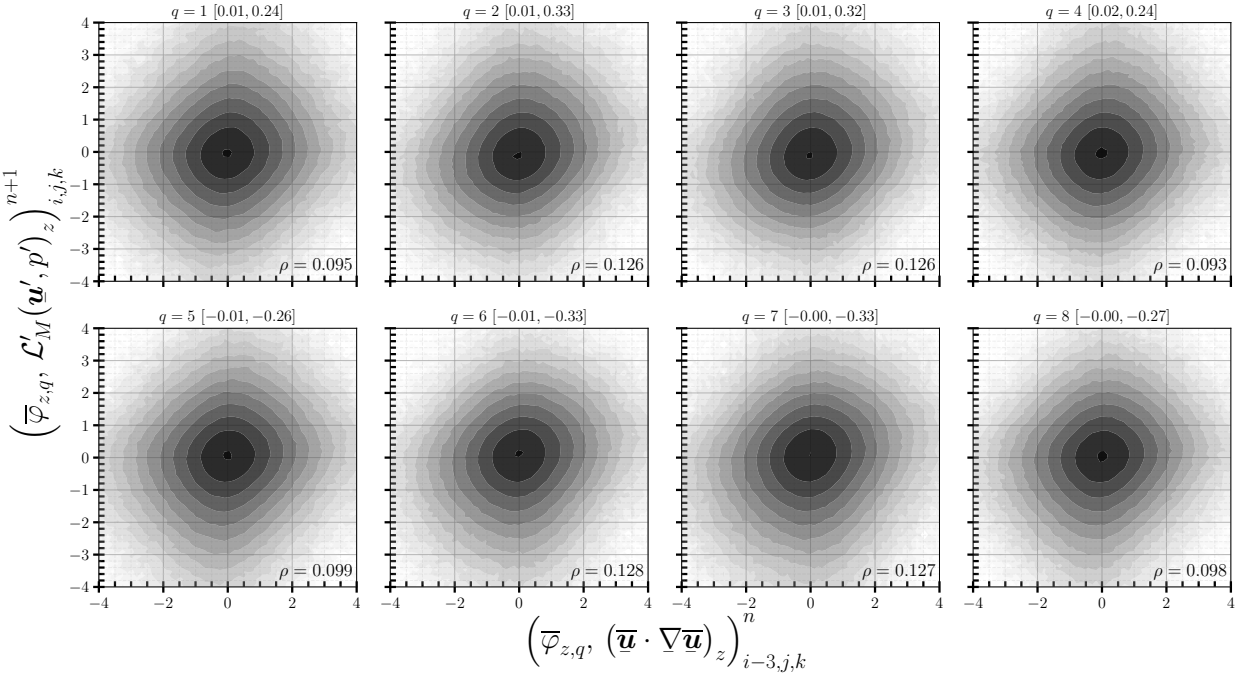


FIGURE A.102 Histograms of the t^n $(i-3, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

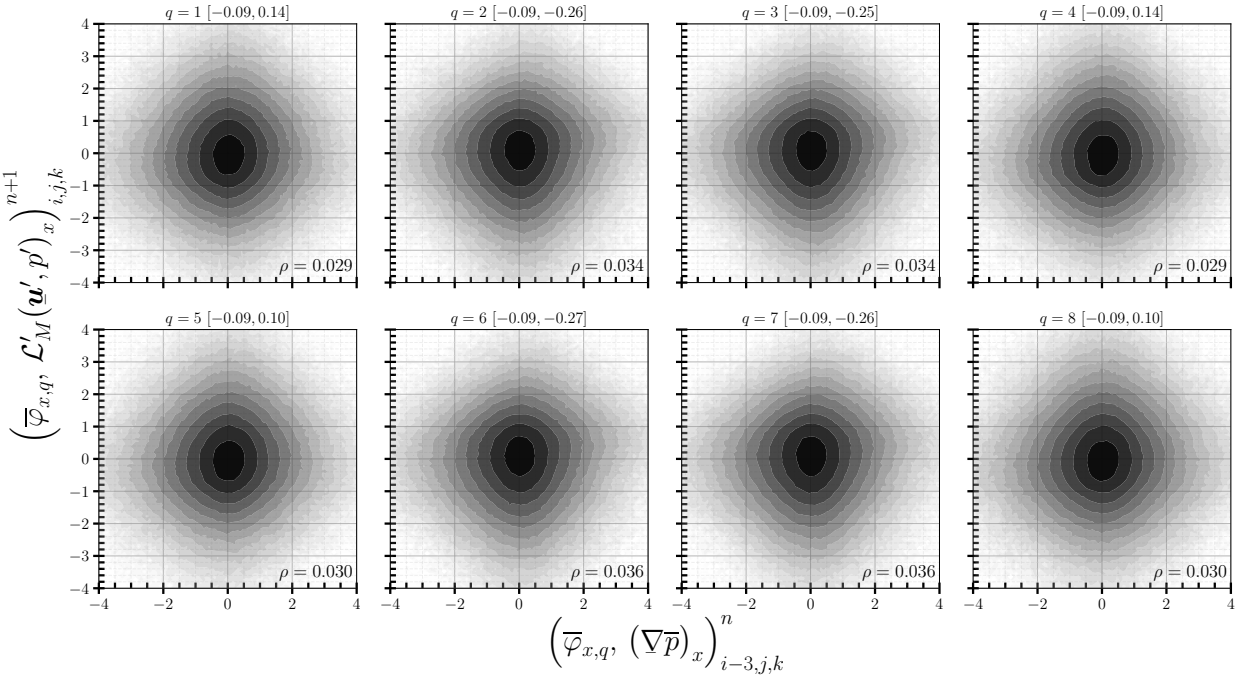


FIGURE A.103 Histograms of the t^n $(i-3, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

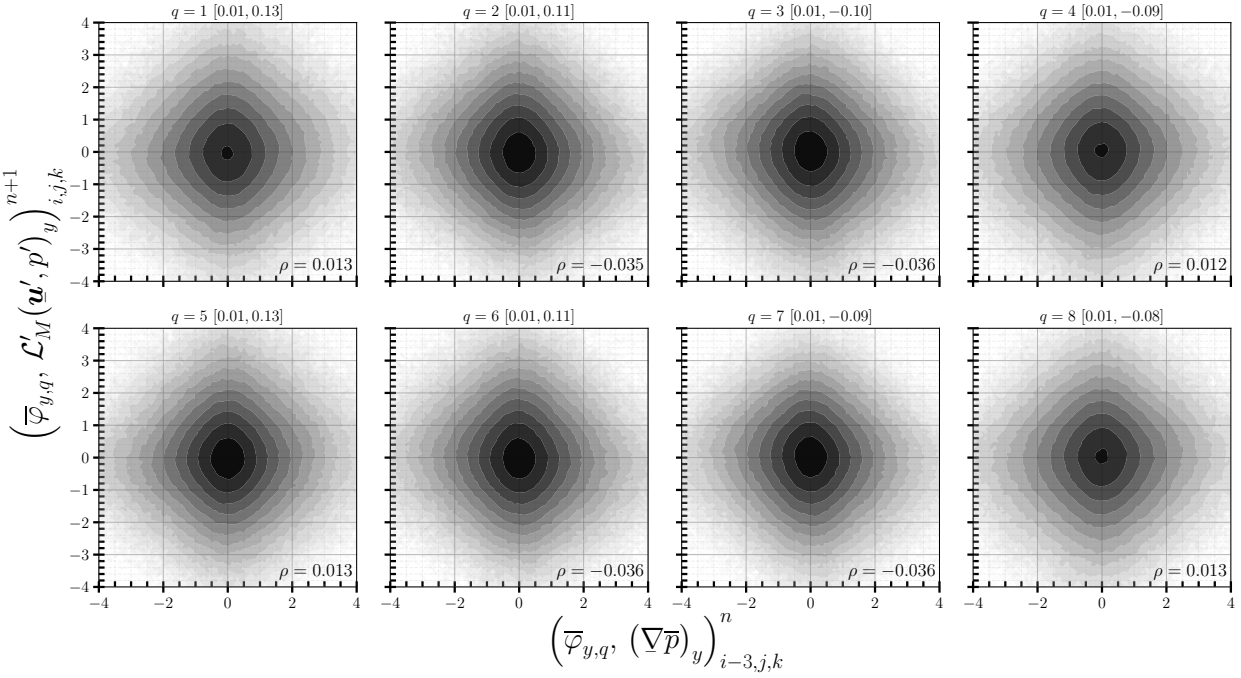


FIGURE A.104 Histograms of the t^n $(i-3, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

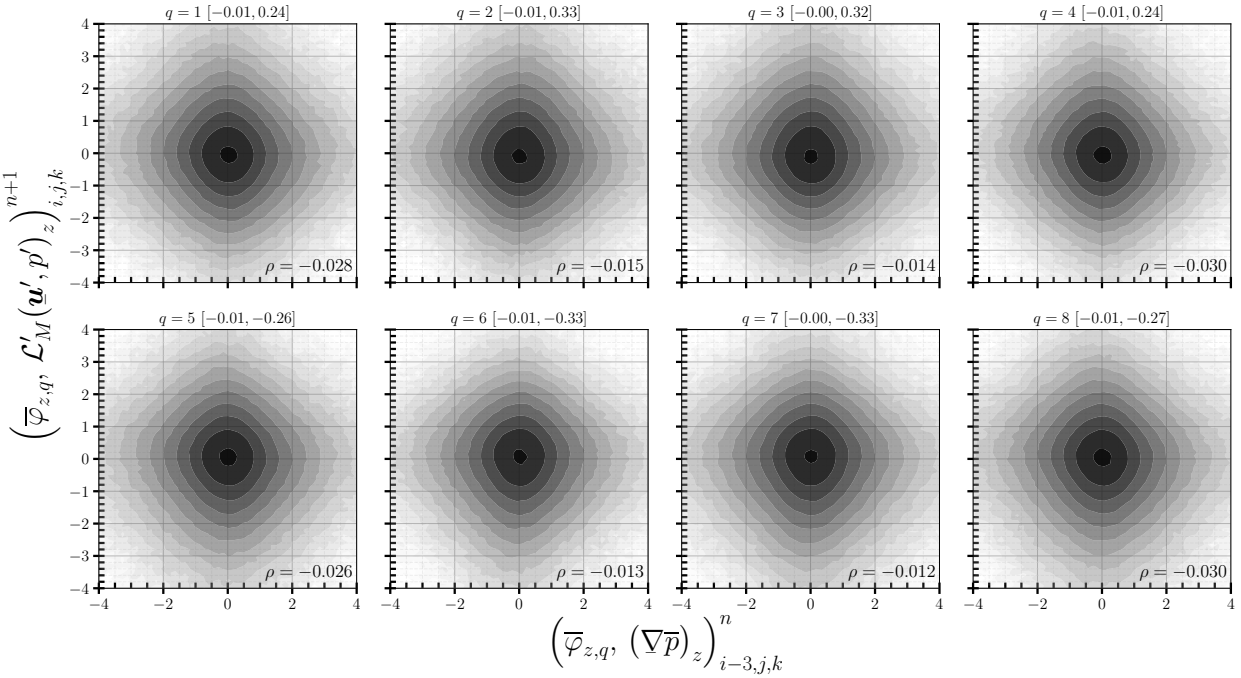


FIGURE A.105 Histograms of the t^n $(i-3, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

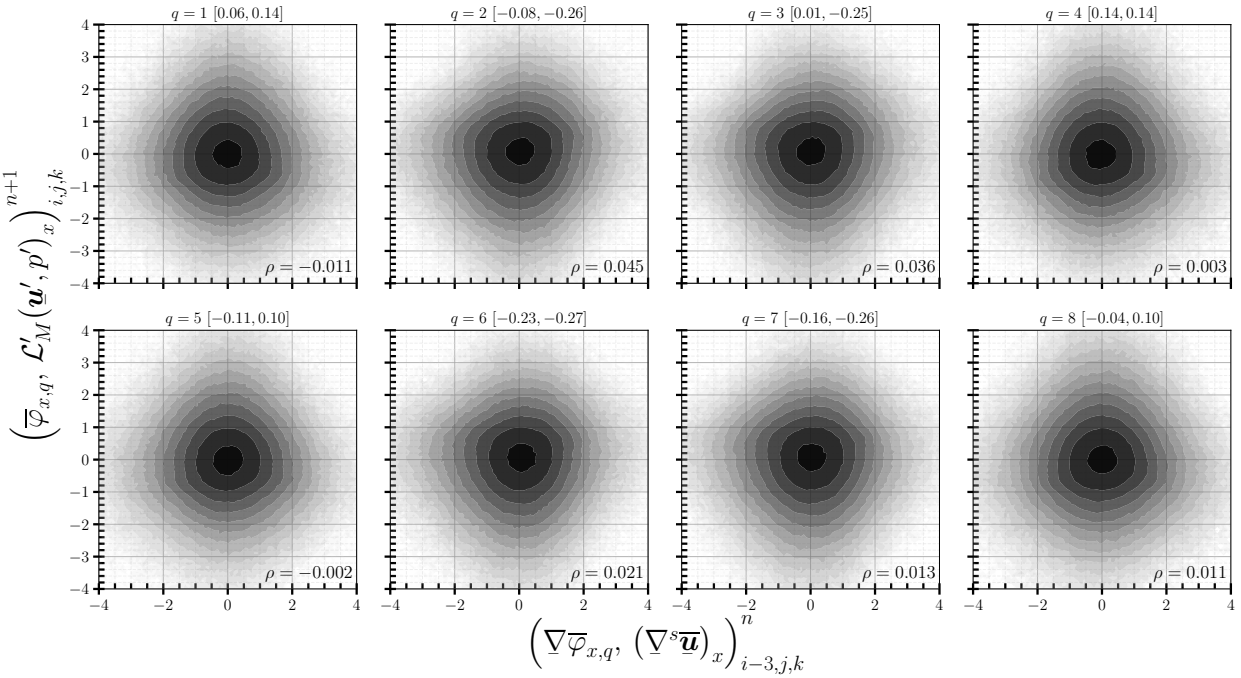


FIGURE A.106 Histograms of the t^n $(i-3, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

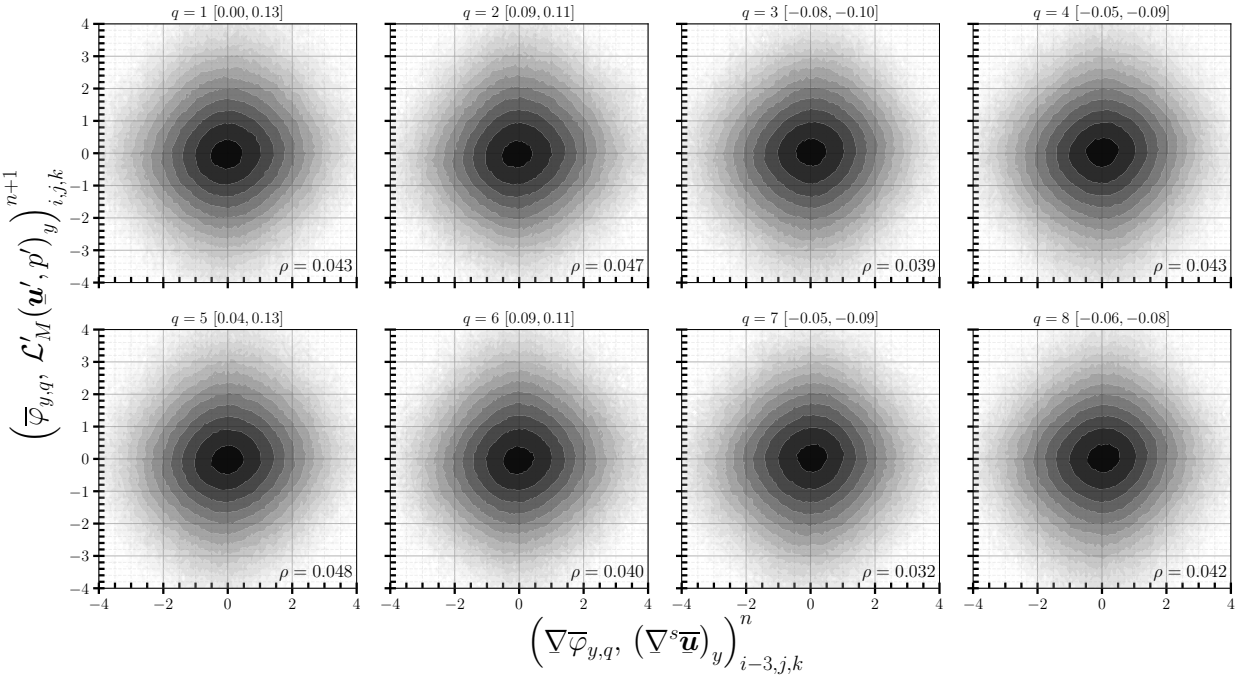


FIGURE A.107 Histograms of the t^n $(i-3, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

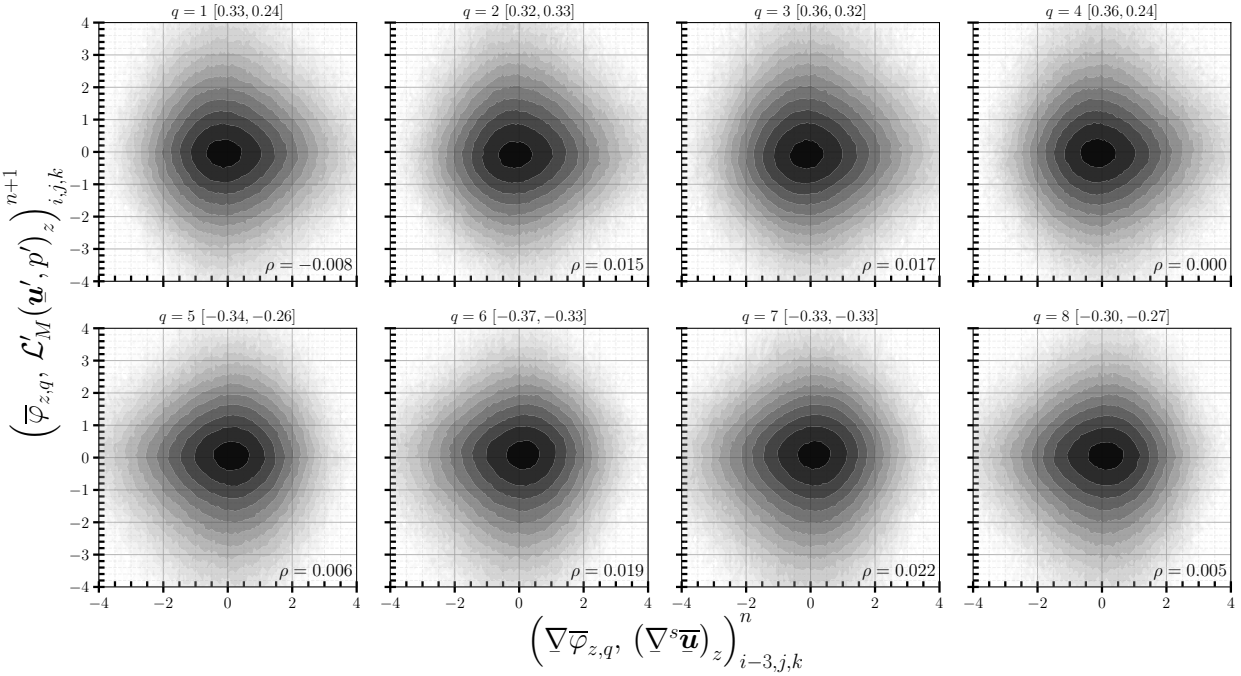


FIGURE A.108 Histograms of the t^n $(i-3, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.5 $(i-4, j, k)$ Histograms

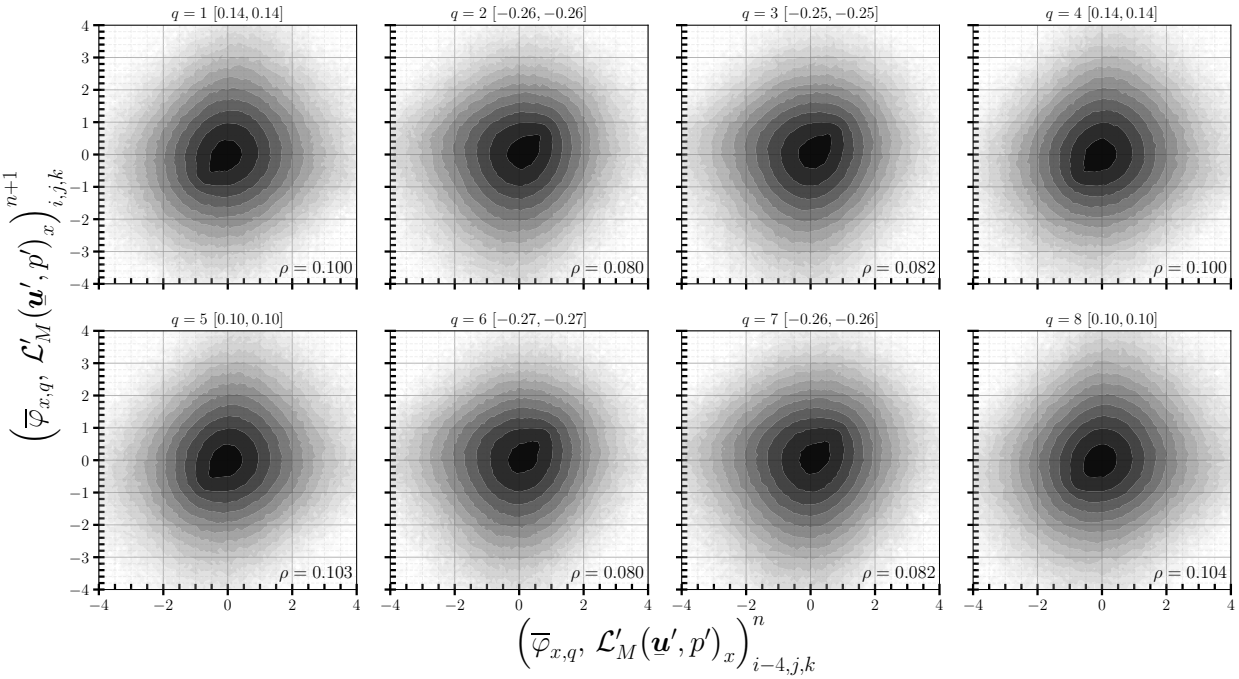


FIGURE A.109 Histograms of the t^n $(i-4, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

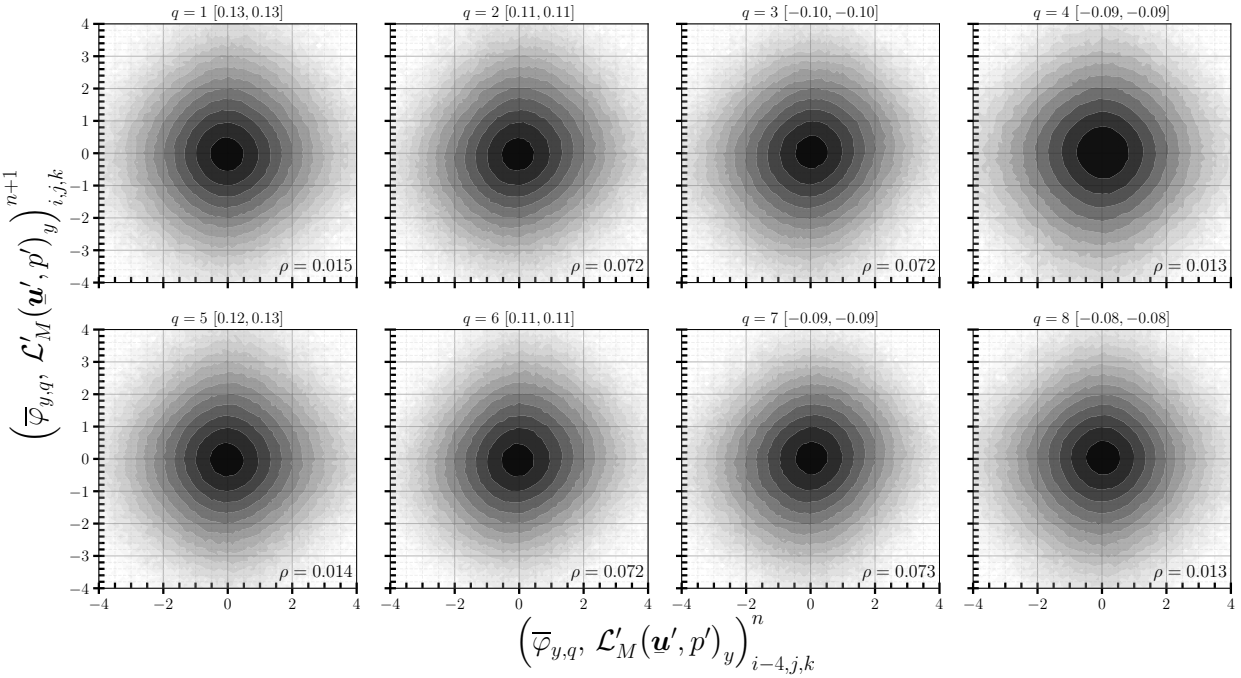


FIGURE A.110 Histograms of the t^n $(i-4, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

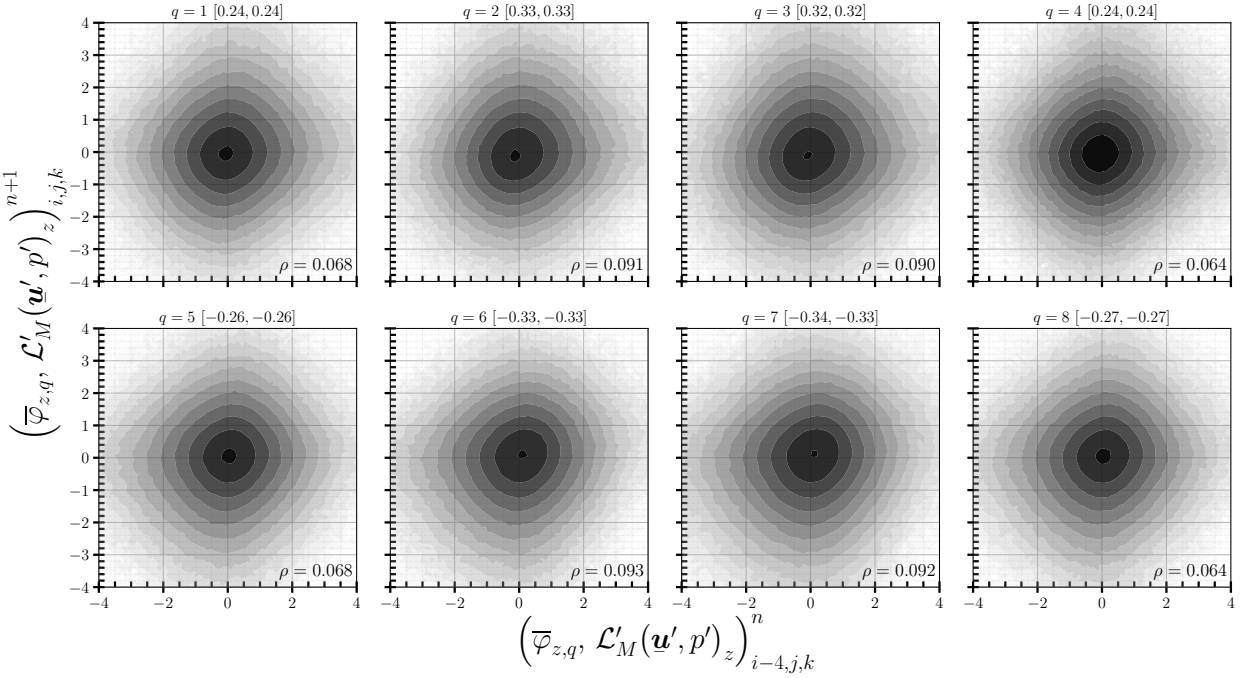


FIGURE A.111 Histograms of the t^n $(i-4, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

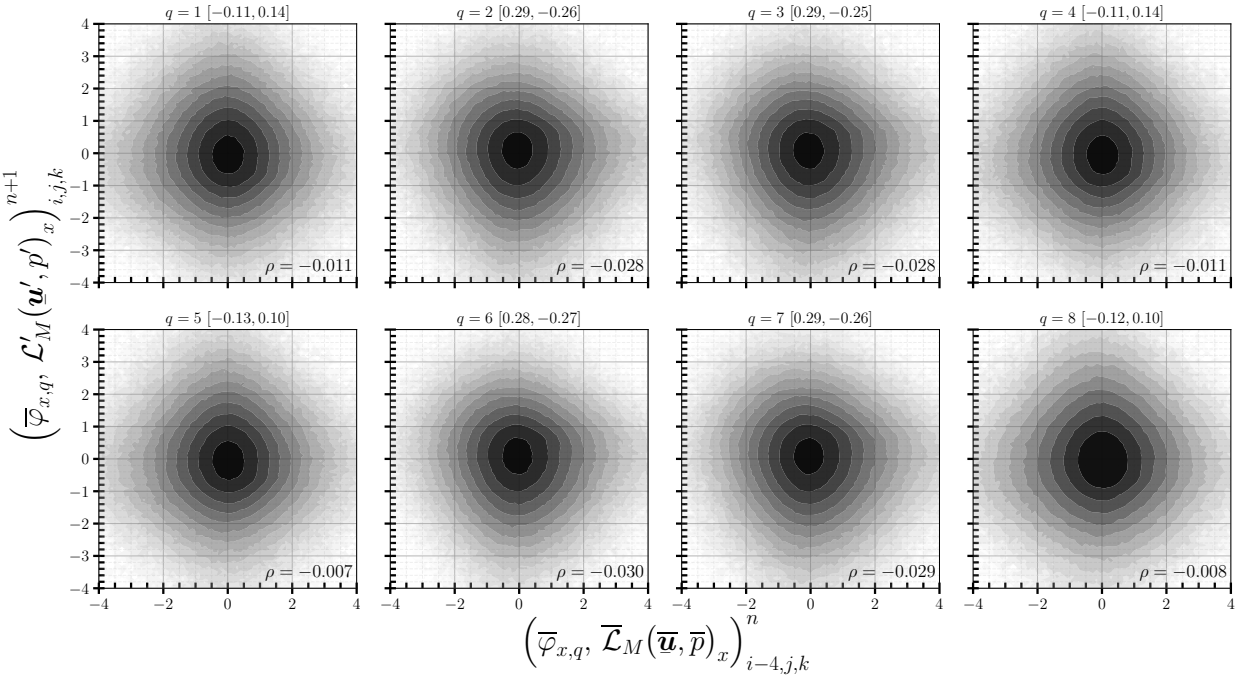


FIGURE A.112 Histograms of the t^n $(i-4, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

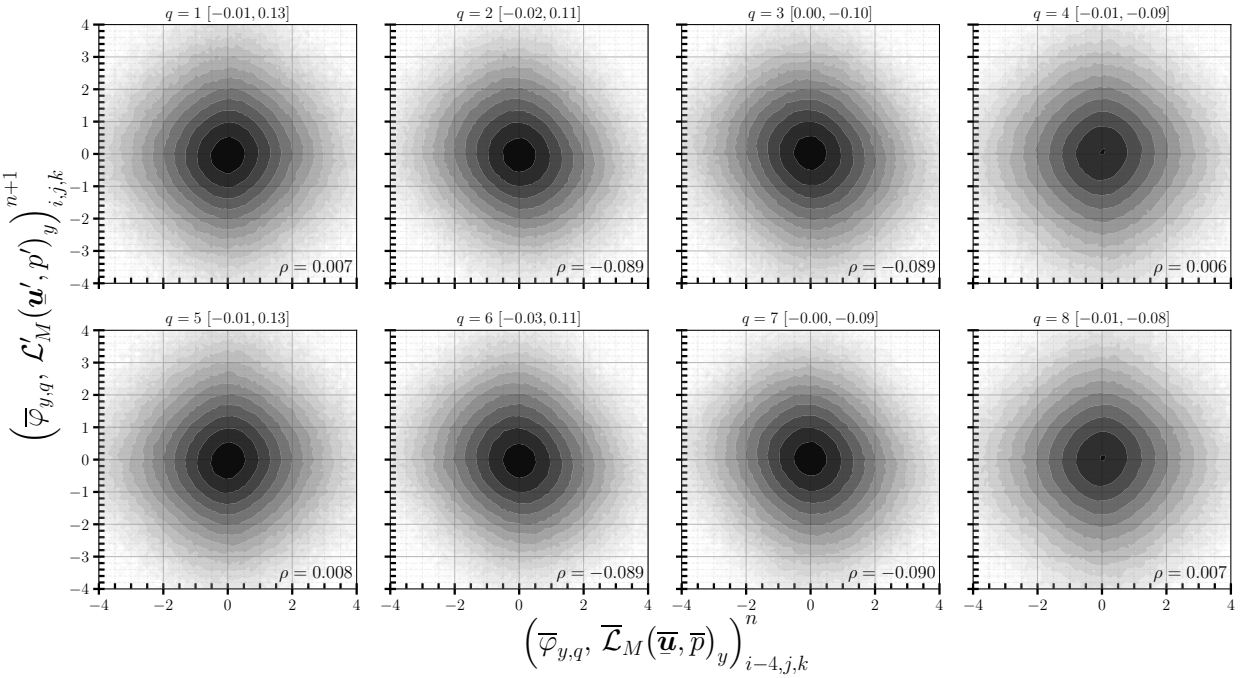


FIGURE A.113 Histograms of the t^n $(i-4, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

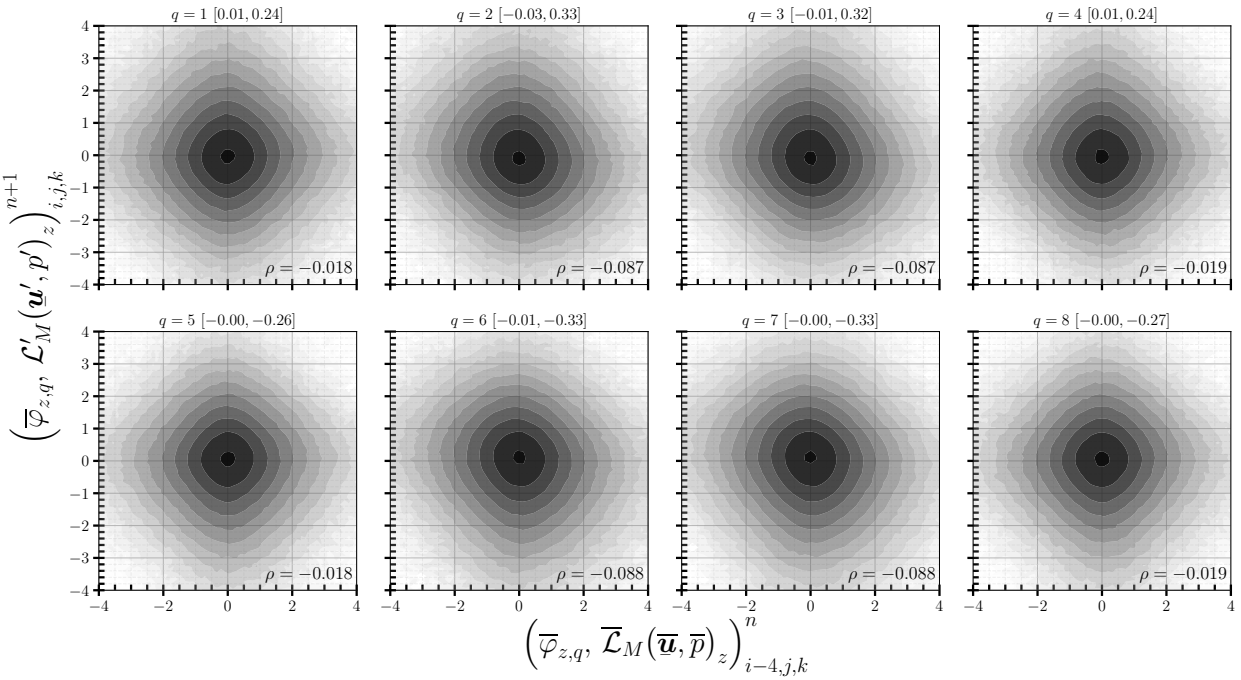


FIGURE A.114 Histograms of the t^n $(i-4, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

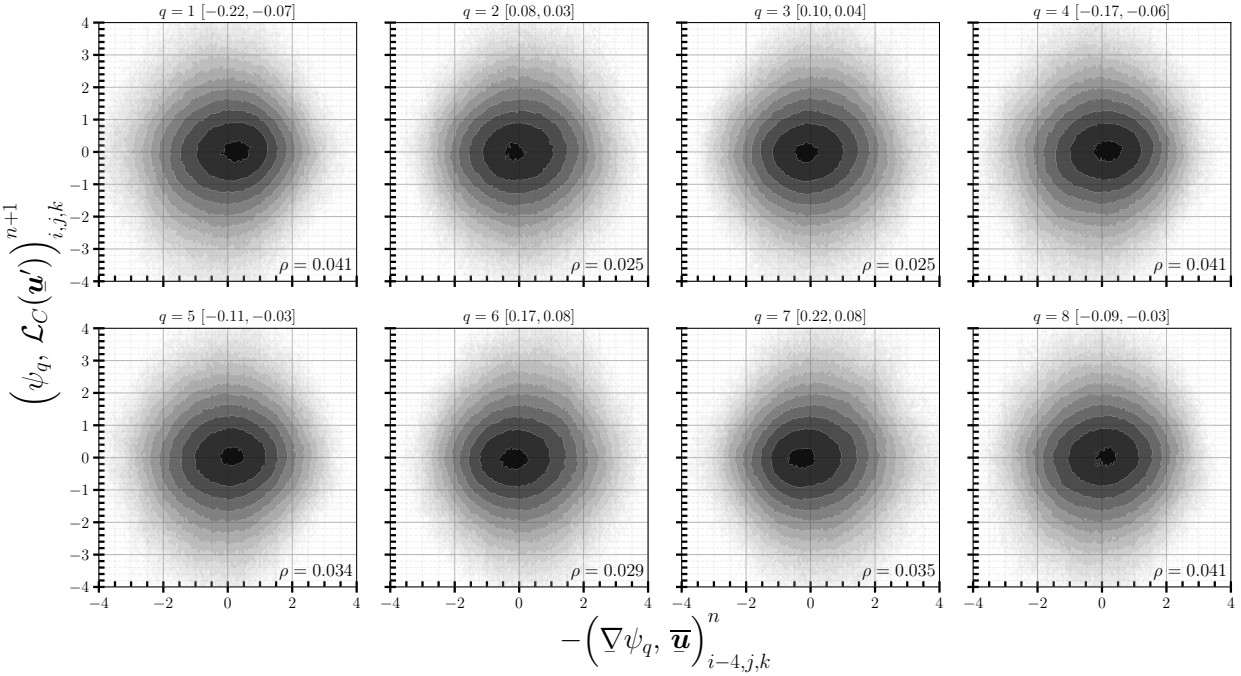


FIGURE A.115 Histograms of the t^n $(i-4, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

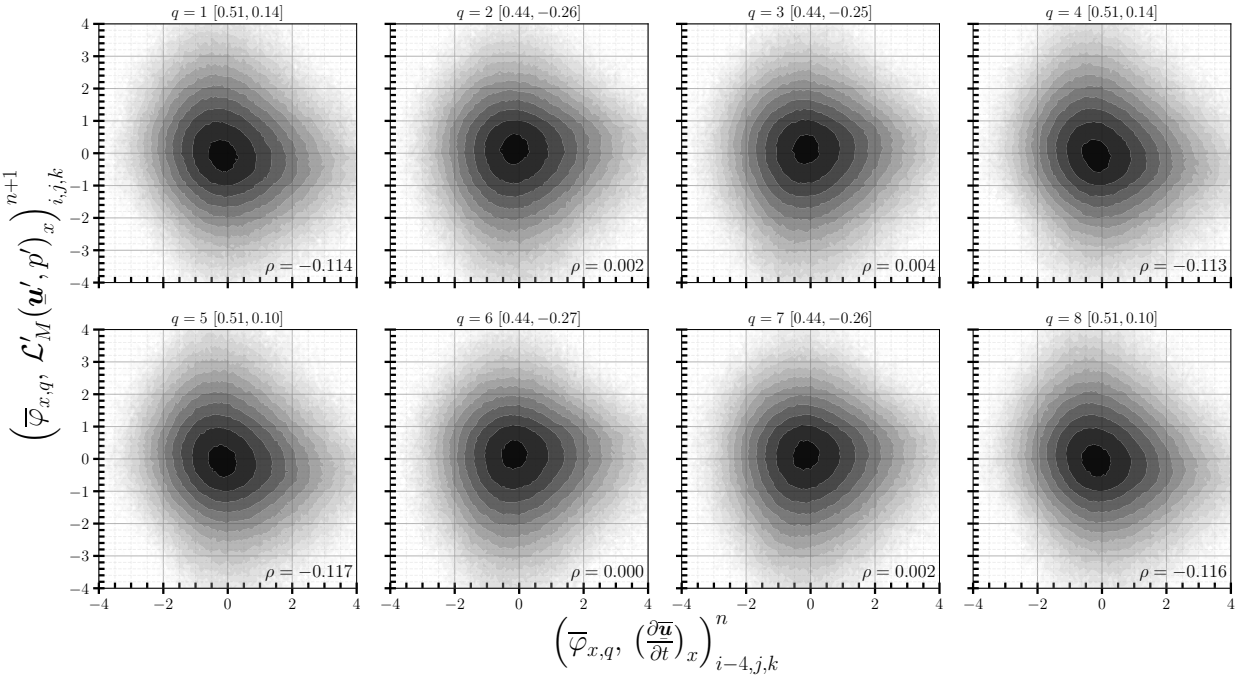


FIGURE A.116 Histograms of the t^n $(i-4, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

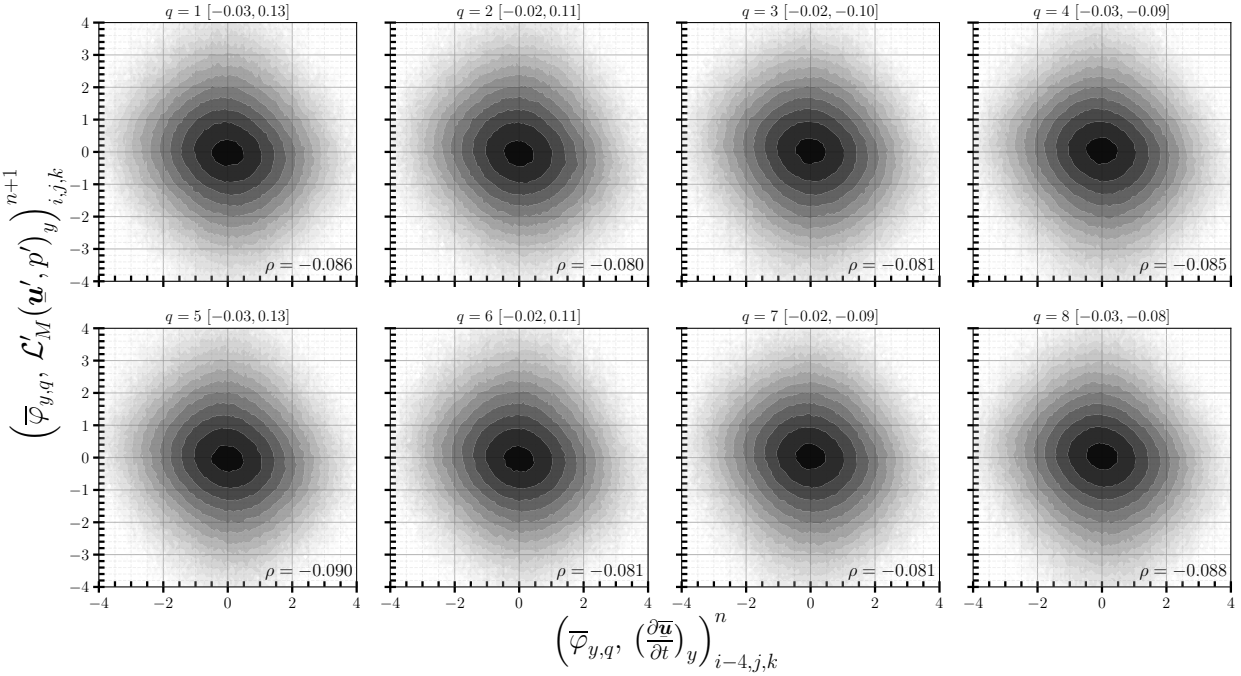


FIGURE A.117 Histograms of the t^n $(i-4, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

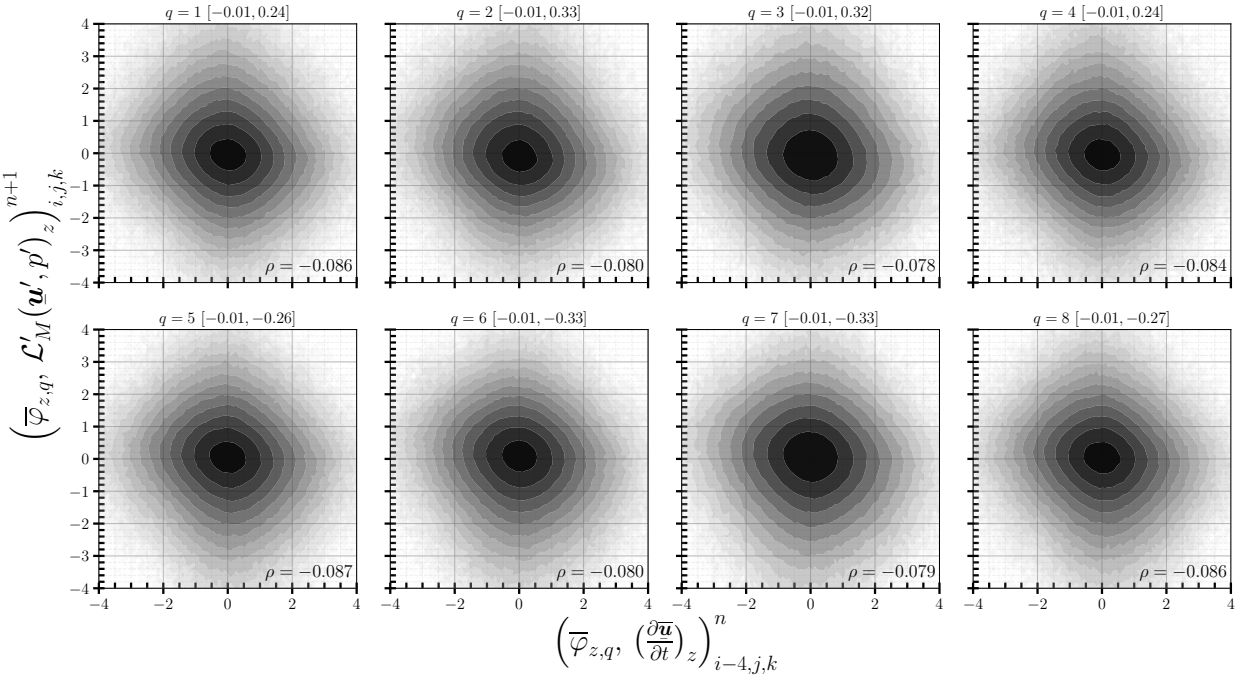


FIGURE A.118 Histograms of the t^n $(i-4, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

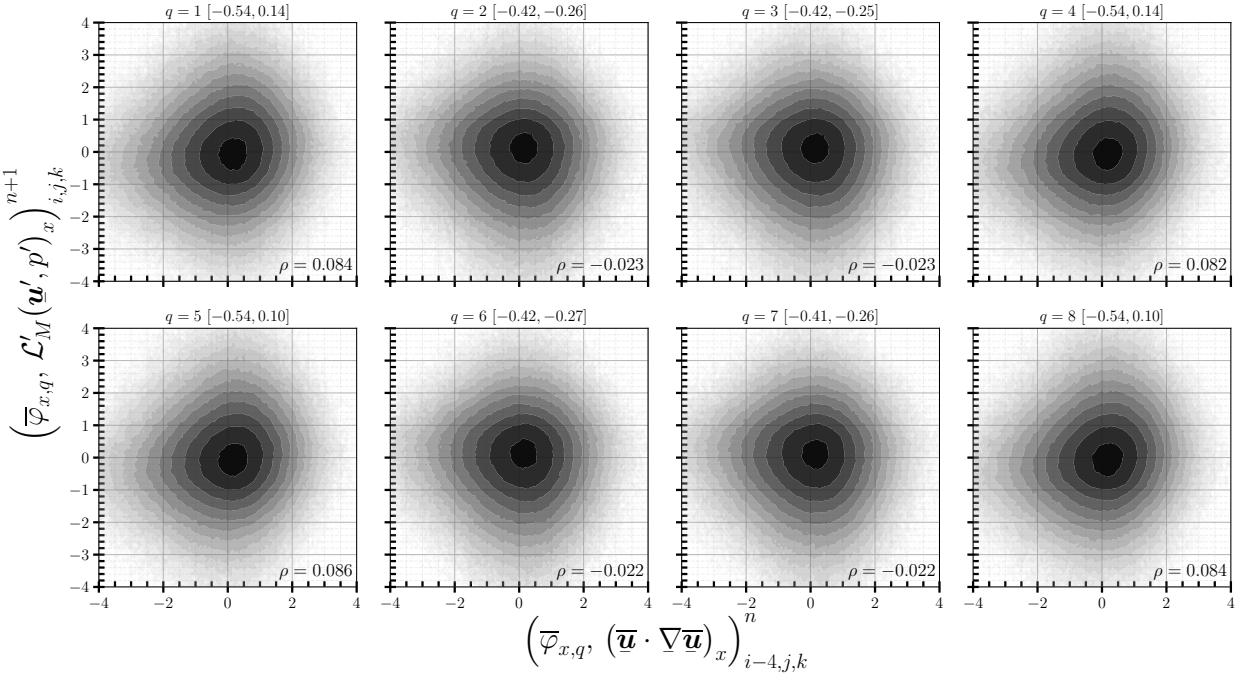


FIGURE A.119 Histograms of the t^n $(i-4, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

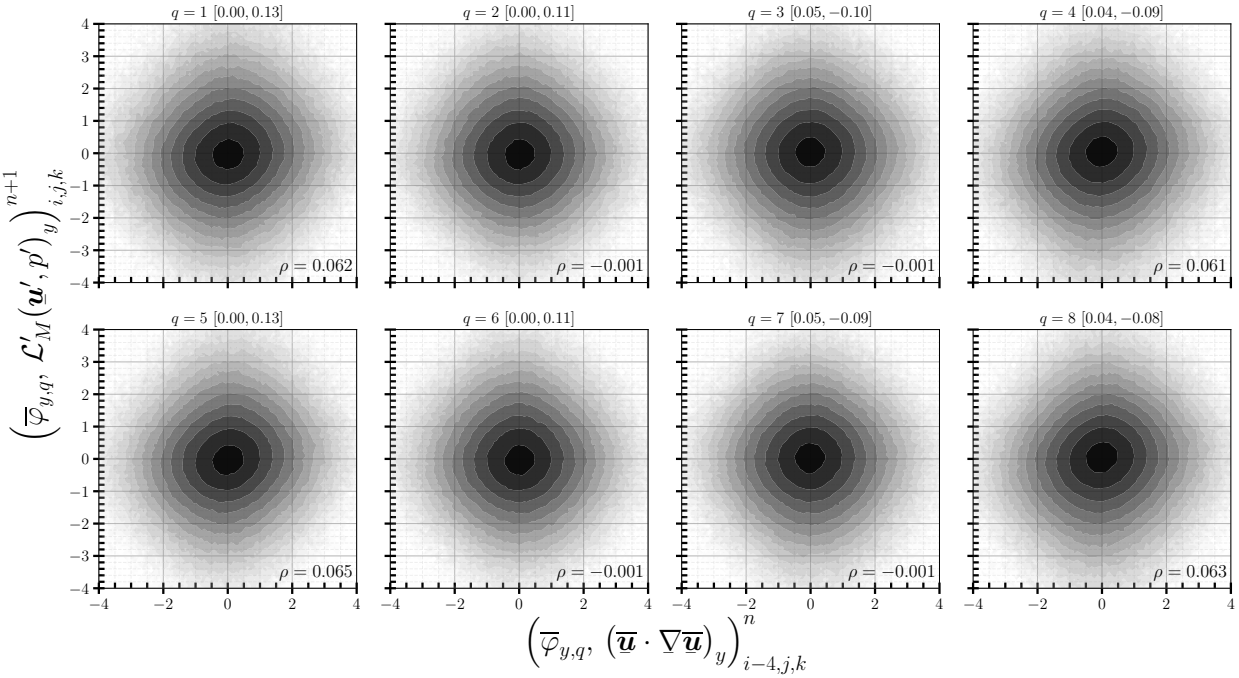


FIGURE A.120 Histograms of the t^n $(i-4, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

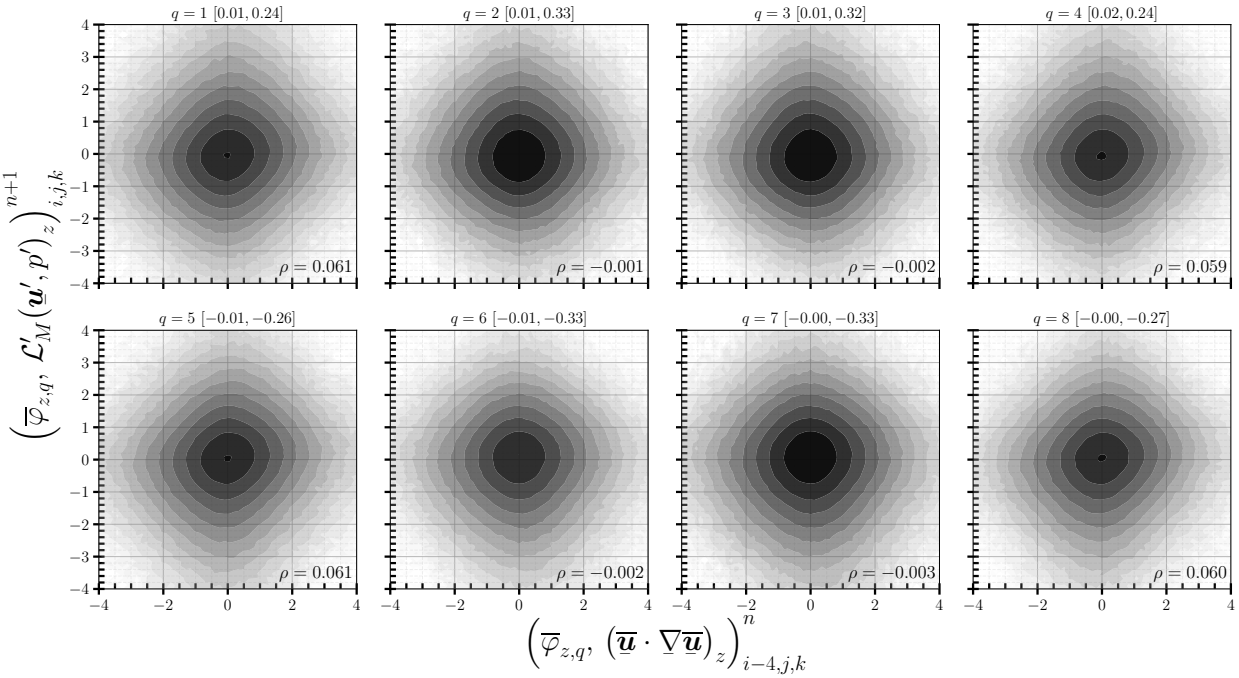


FIGURE A.121 Histograms of the t^n $(i-4, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

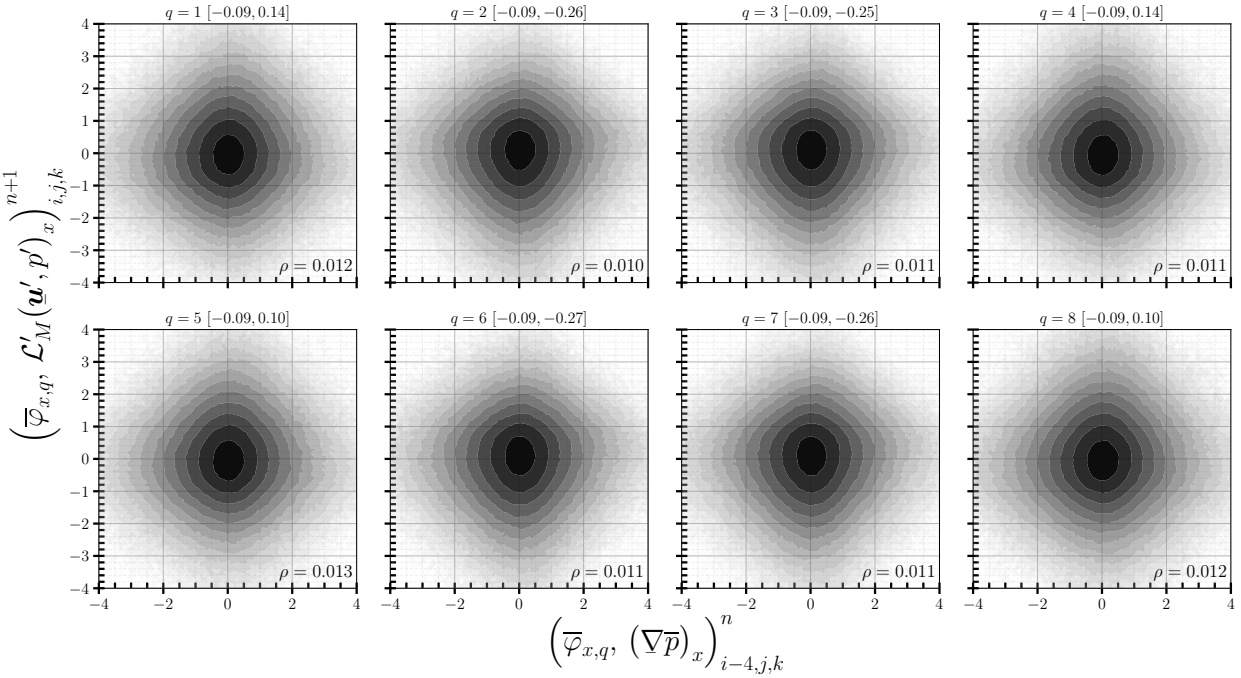


FIGURE A.122 Histograms of the t^n $(i-4, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

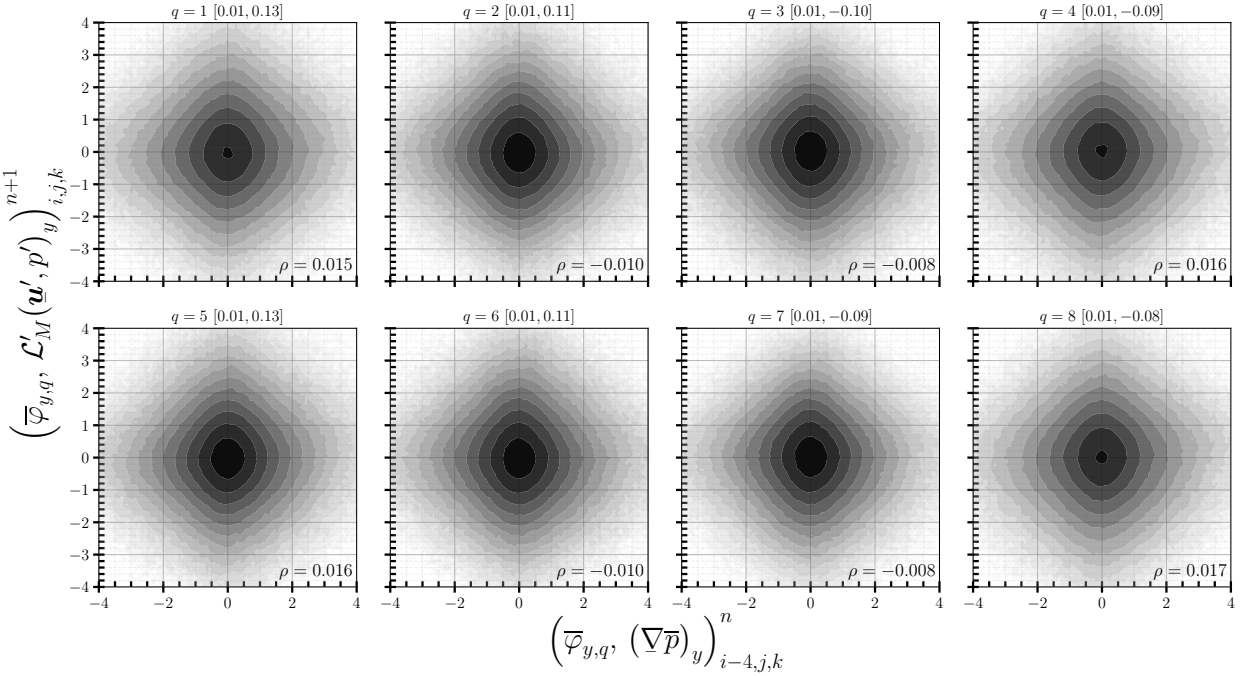


FIGURE A.123 Histograms of the t^n $(i-4, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

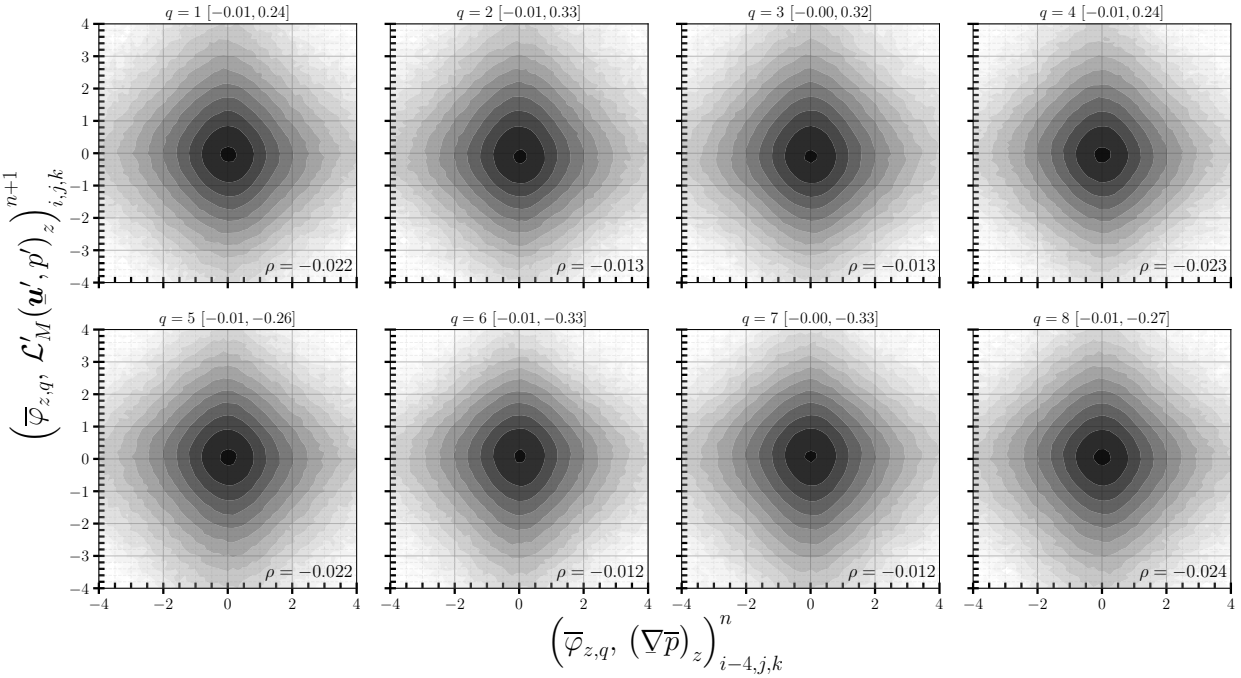


FIGURE A.124 Histograms of the t^n $(i-4, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

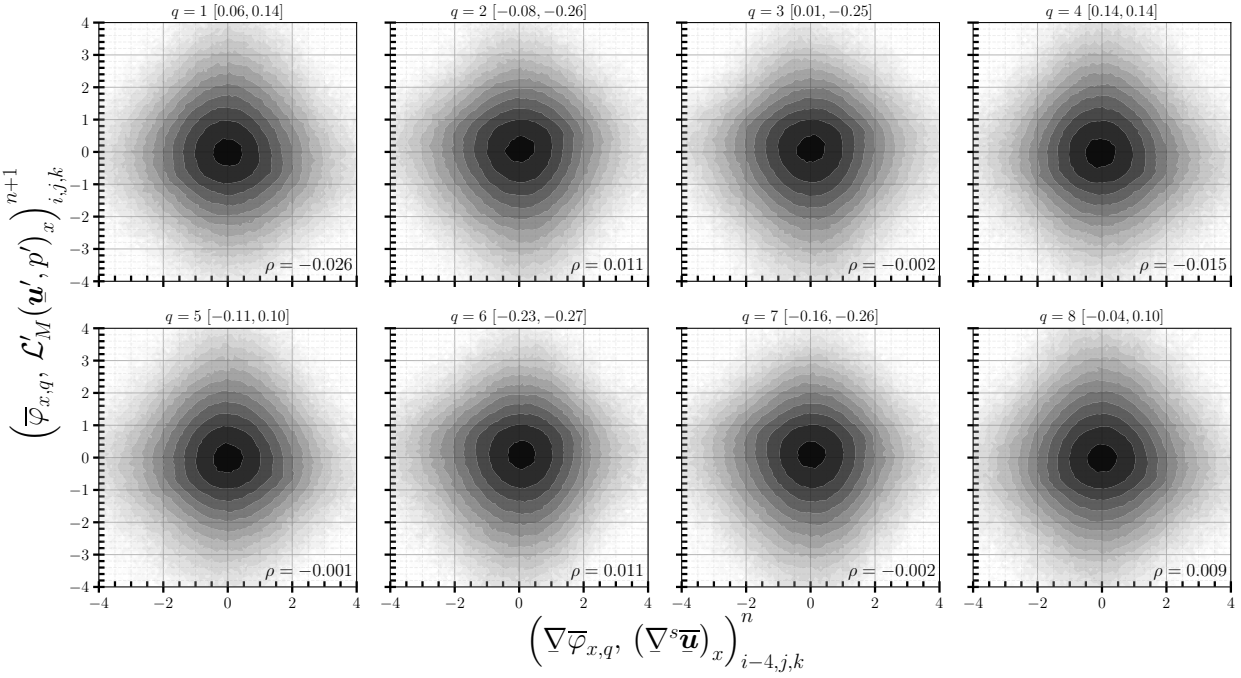


FIGURE A.125 Histograms of the t^n $(i-4, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

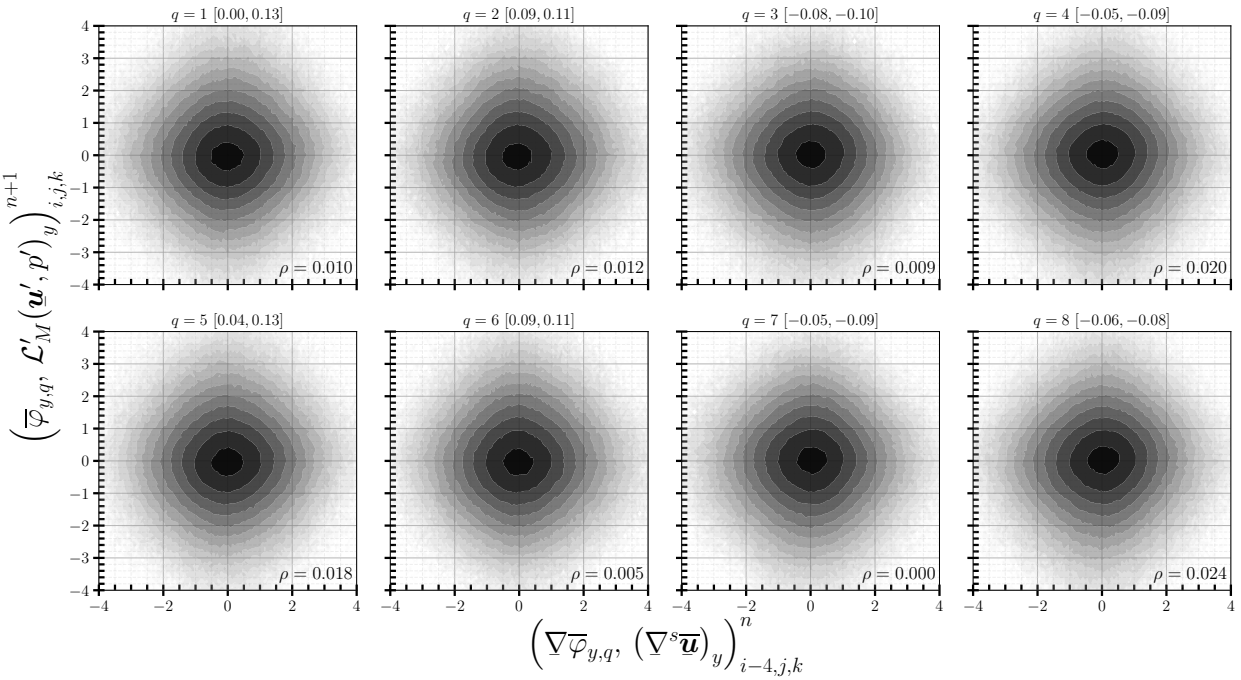


FIGURE A.126 Histograms of the t^n $(i-4, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

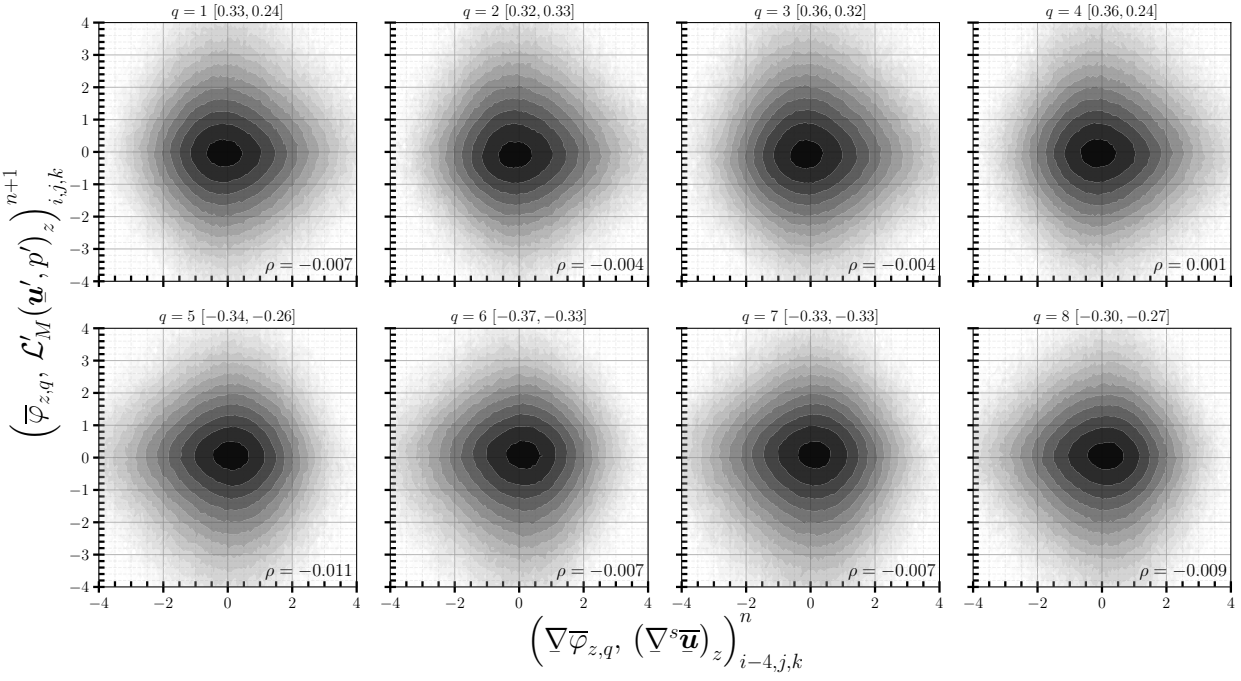


FIGURE A.127 Histograms of the t^n $(i-4, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.6 $(i+1, j, k)$ Histograms

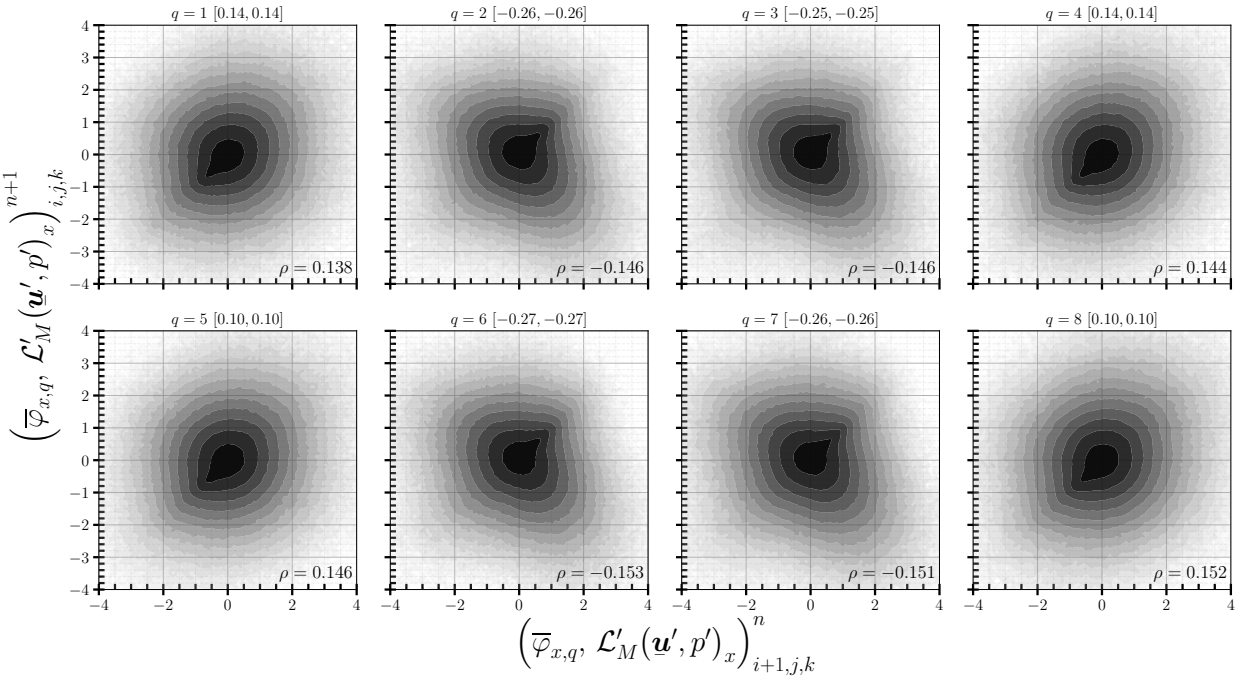


FIGURE A.128 Histograms of the t^n $(i+1, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

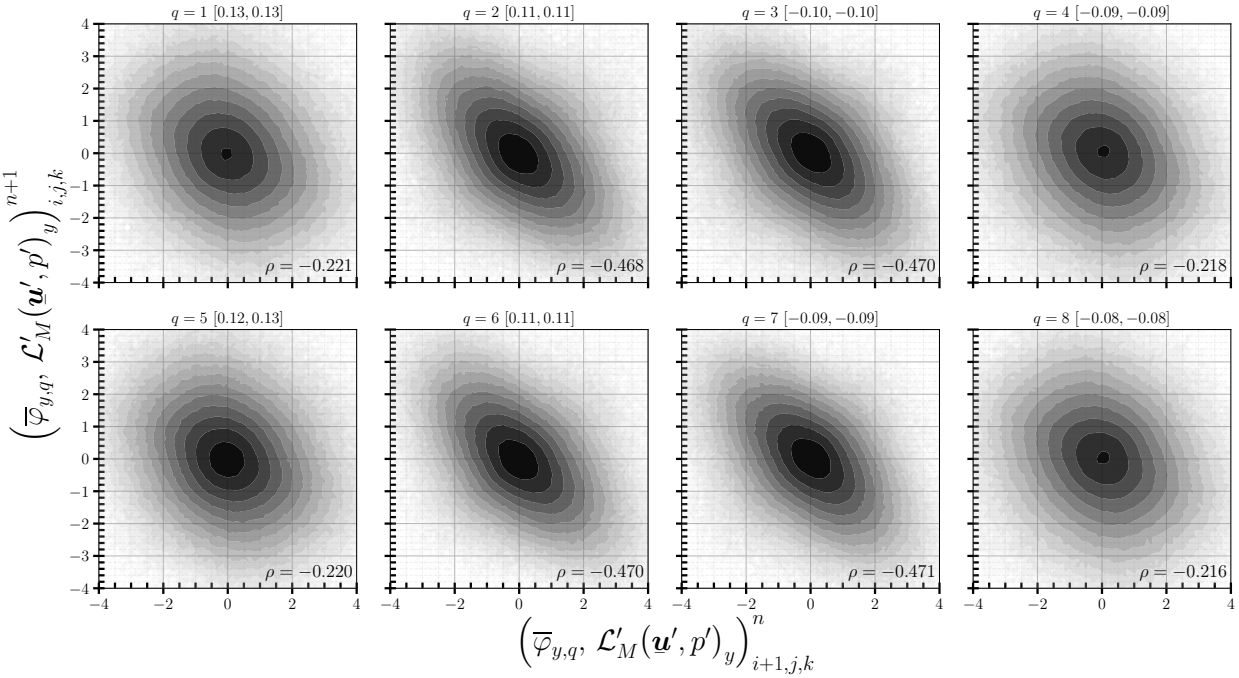


FIGURE A.129 Histograms of the t^n $(i+1, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

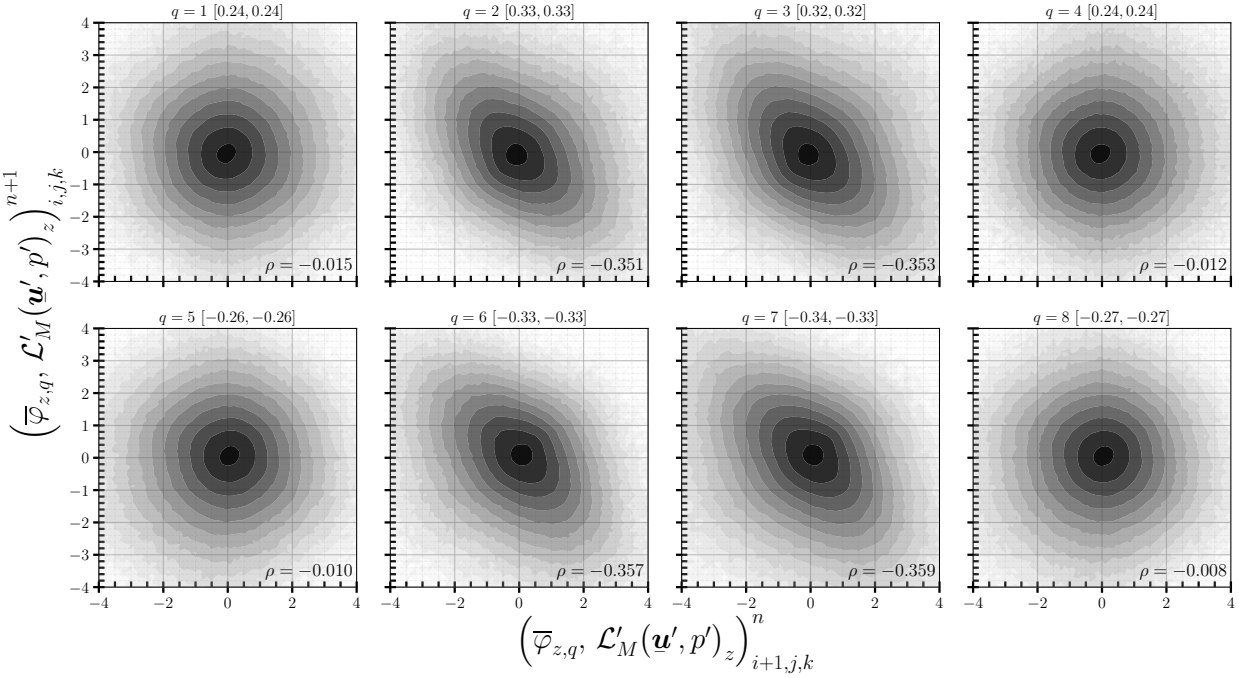


FIGURE A.130 Histograms of the t^n $(i+1, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

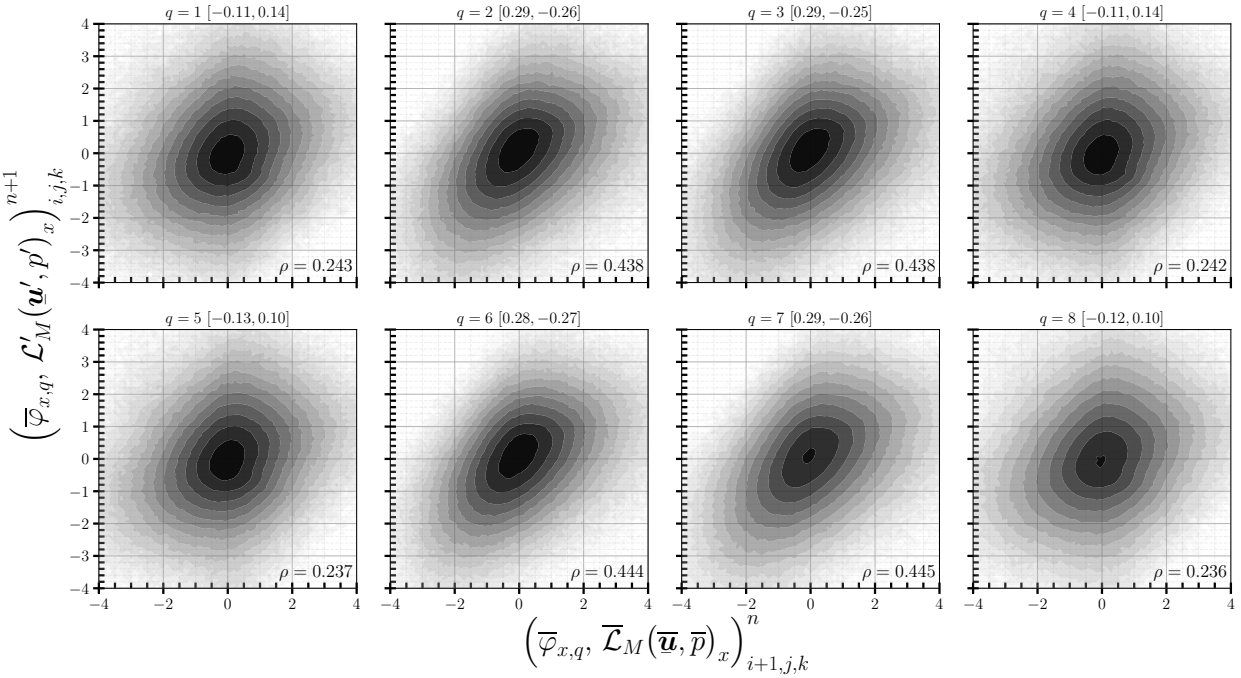


FIGURE A.131 Histograms of the t^n $(i+1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

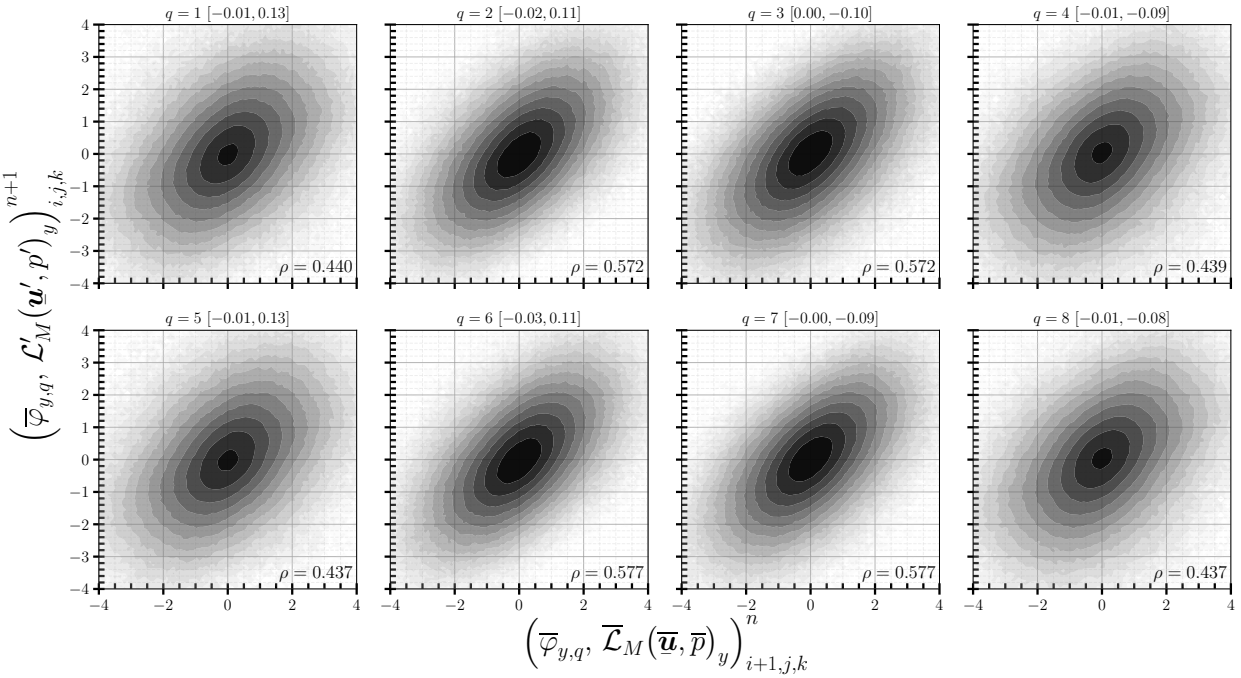


FIGURE A.132 Histograms of the t^n $(i+1, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

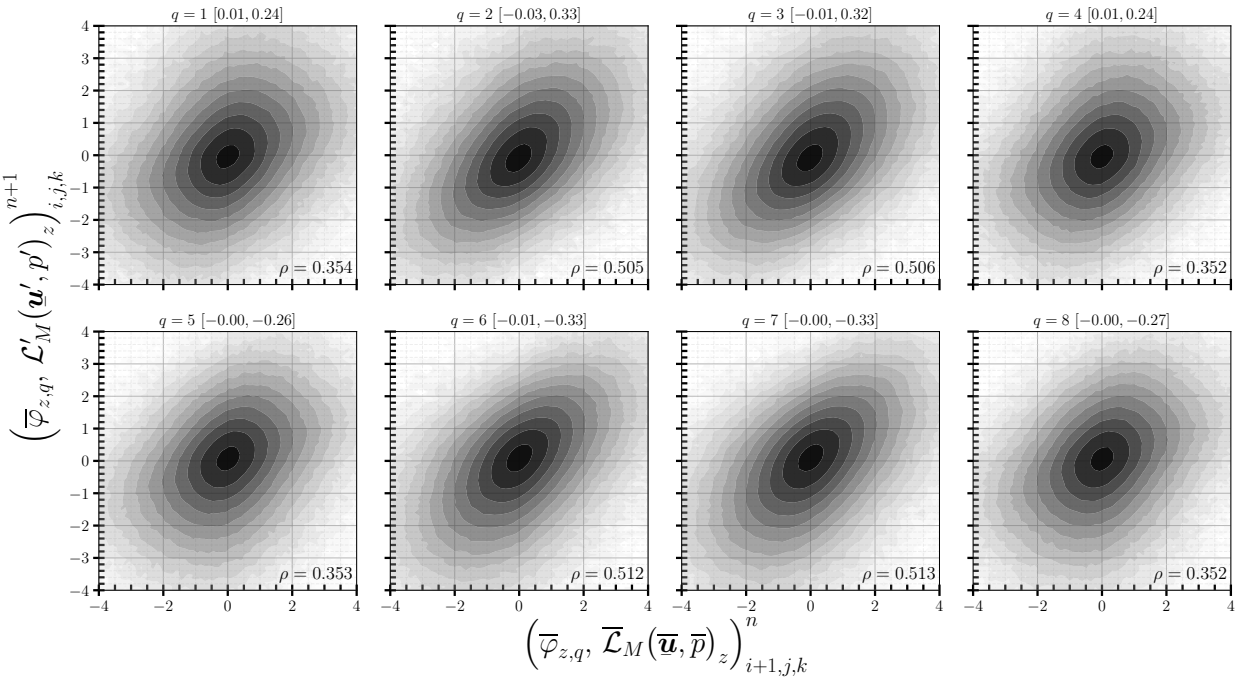


FIGURE A.133 Histograms of the t^n $(i+1, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

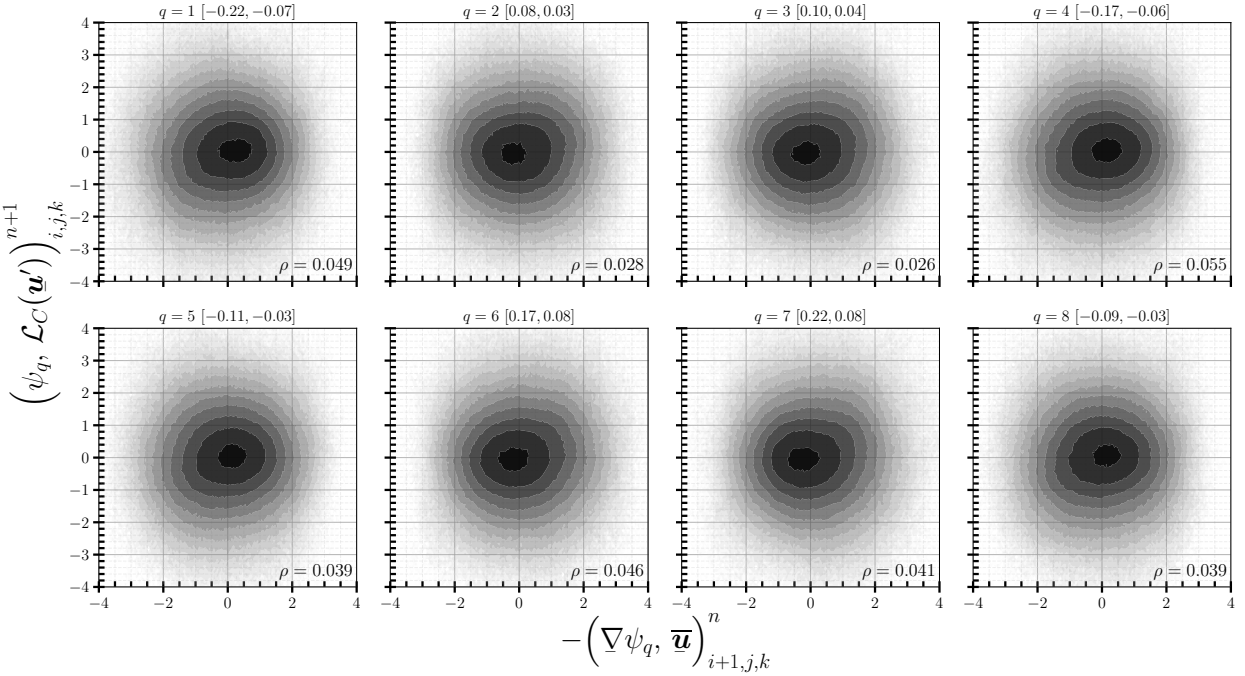


FIGURE A.134 Histograms of the t^n $(i+1, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

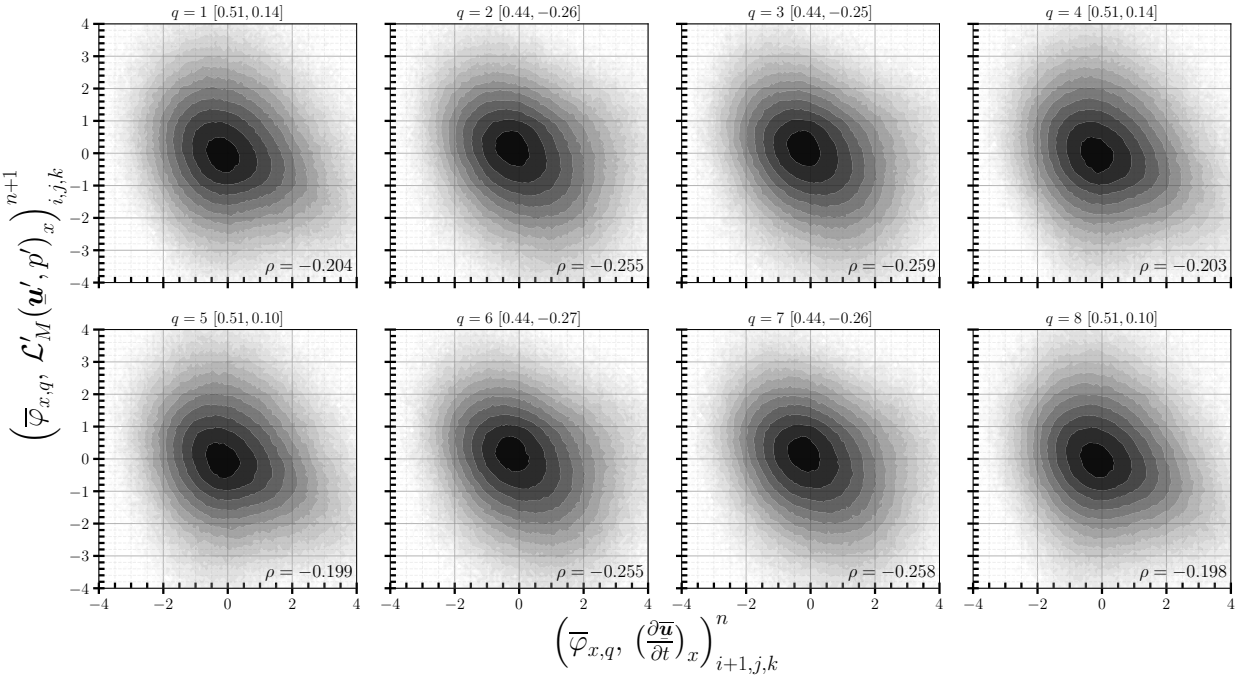


FIGURE A.135 Histograms of the t^n $(i+1, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

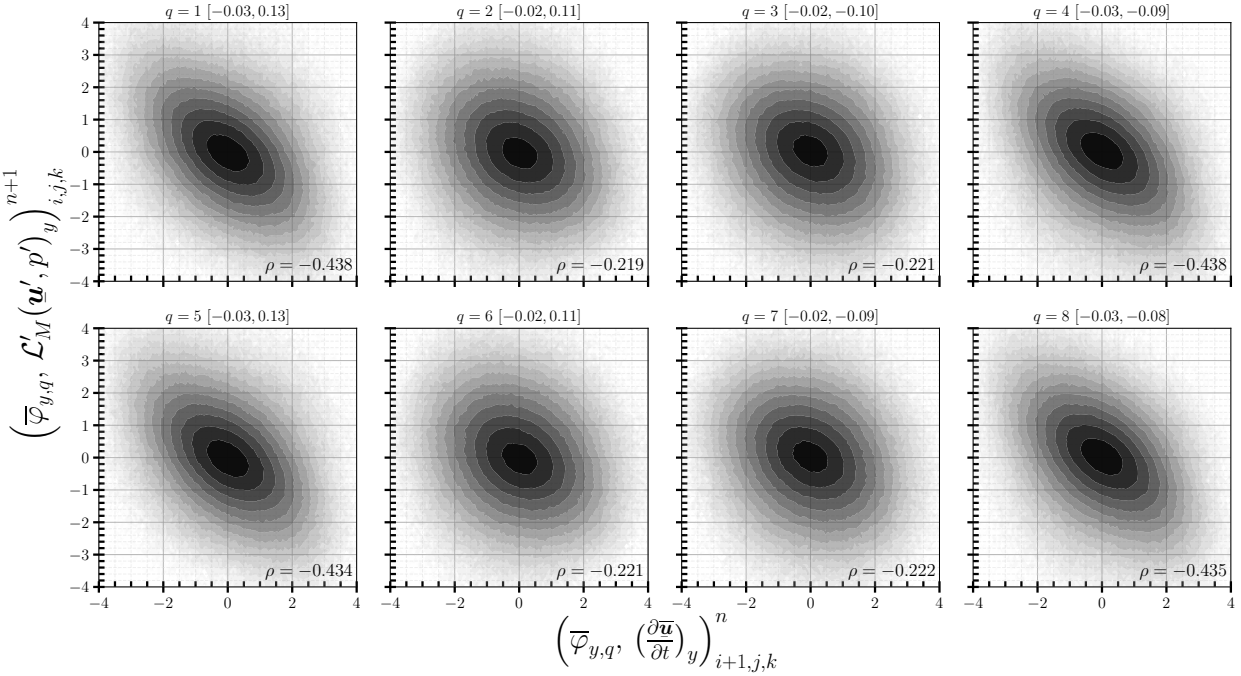


FIGURE A.136 Histograms of the t^n $(i+1, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

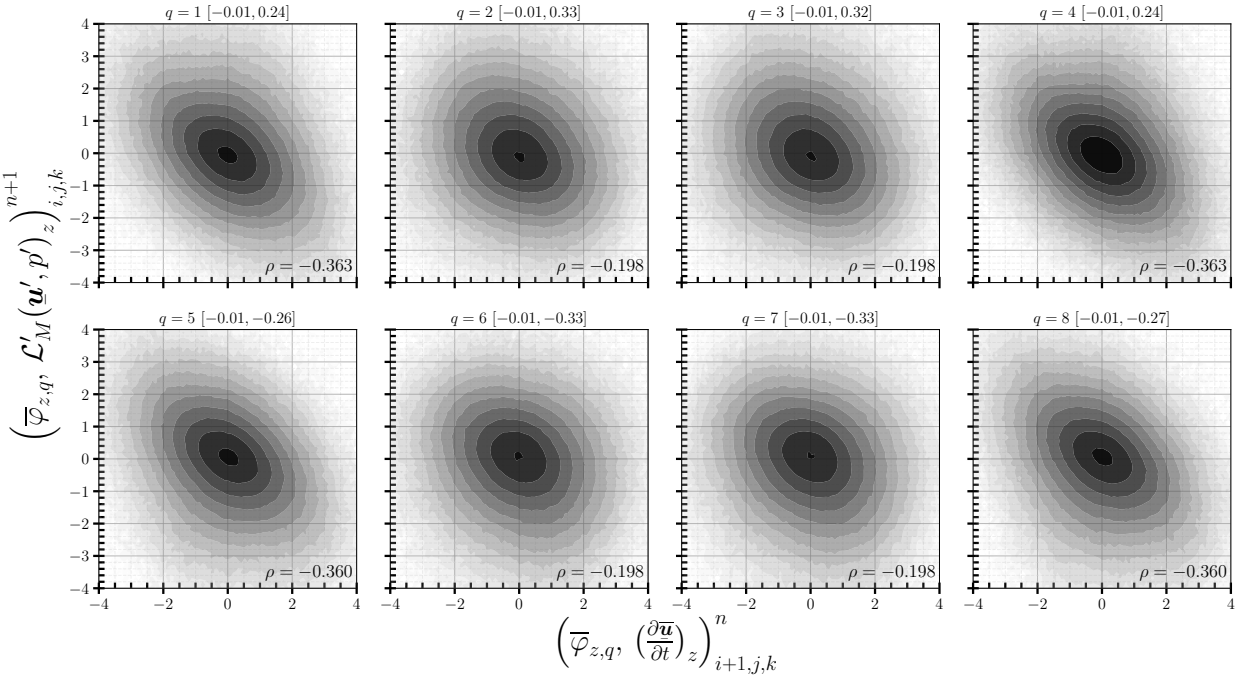


FIGURE A.137 Histograms of the t^n $(i+1, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

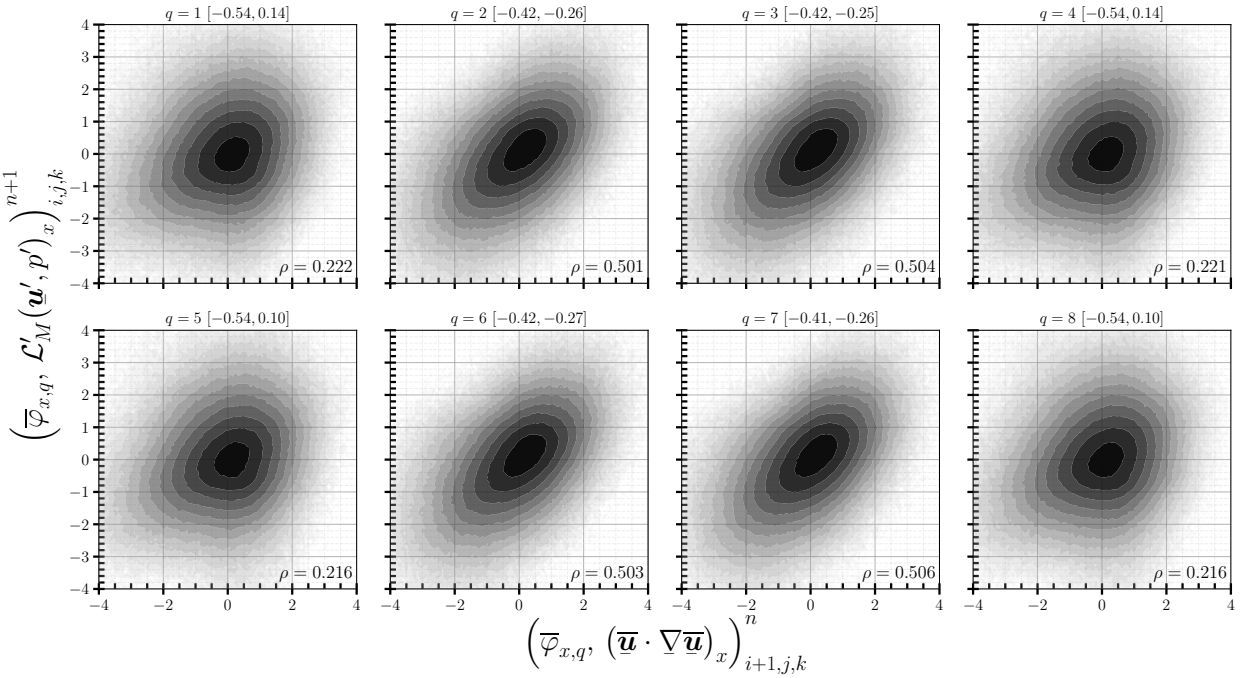


FIGURE A.138 Histograms of the t^n $(i+1, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

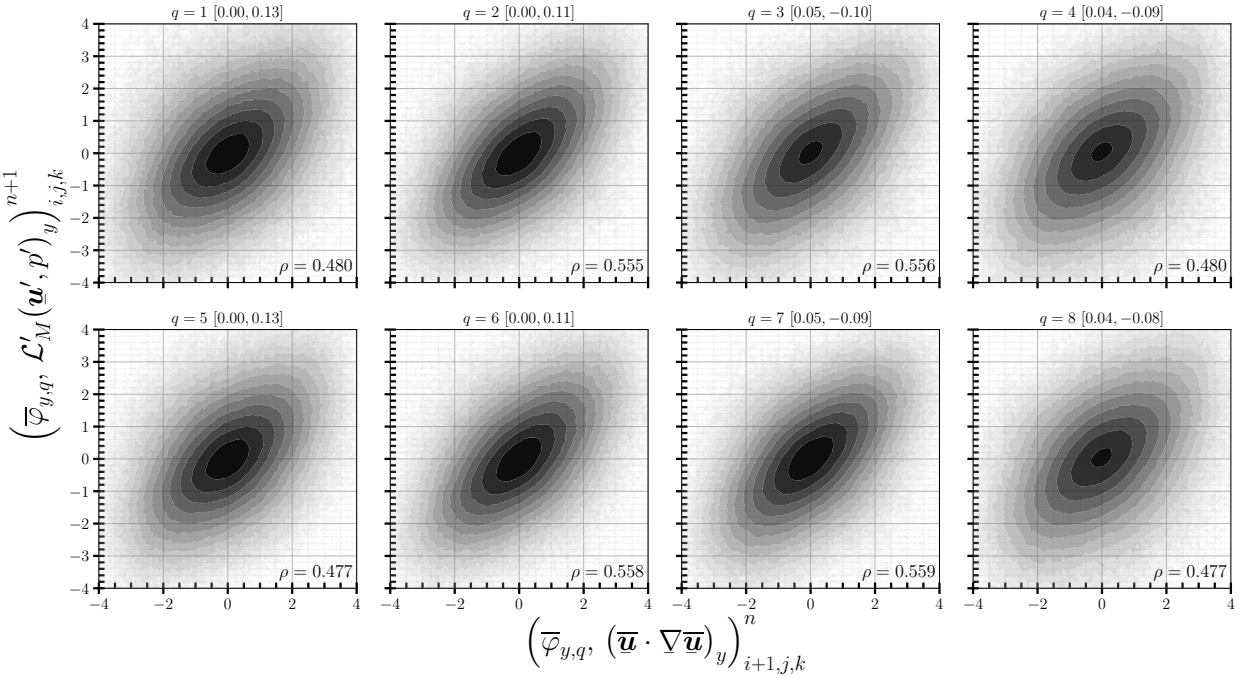


FIGURE A.139 Histograms of the t^n $(i+1, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

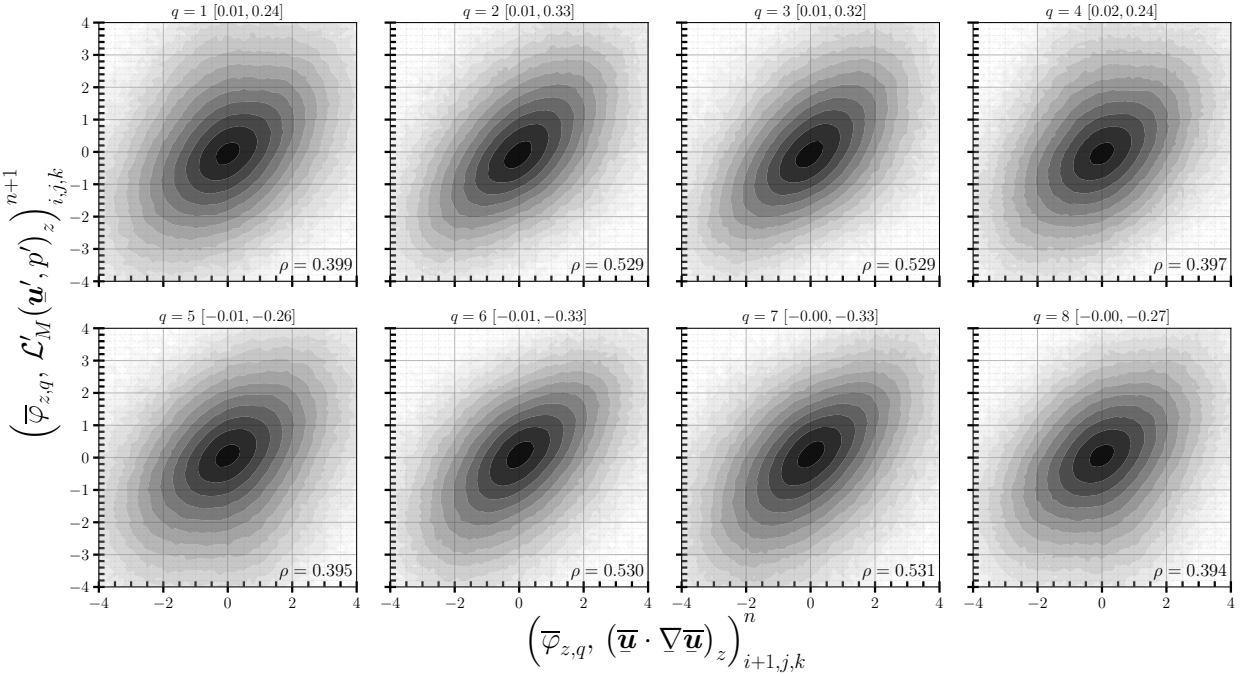


FIGURE A.140 Histograms of the t^n $(i+1, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

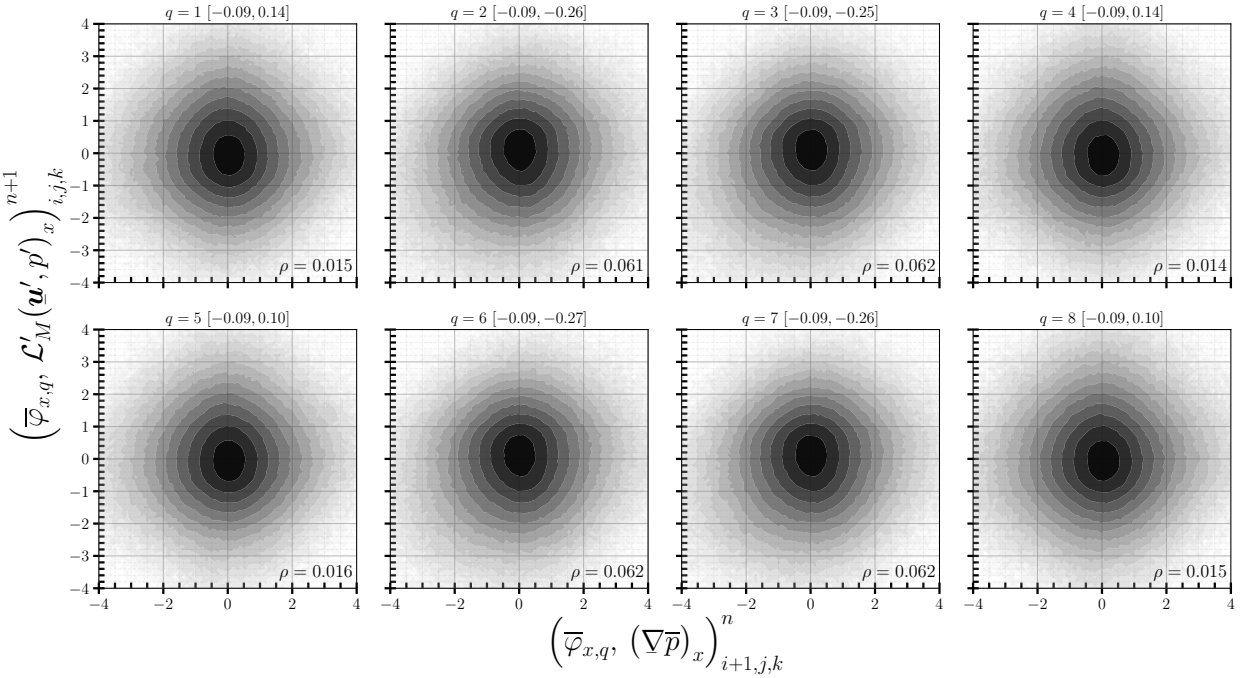


FIGURE A.141 Histograms of the t^n $(i+1, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

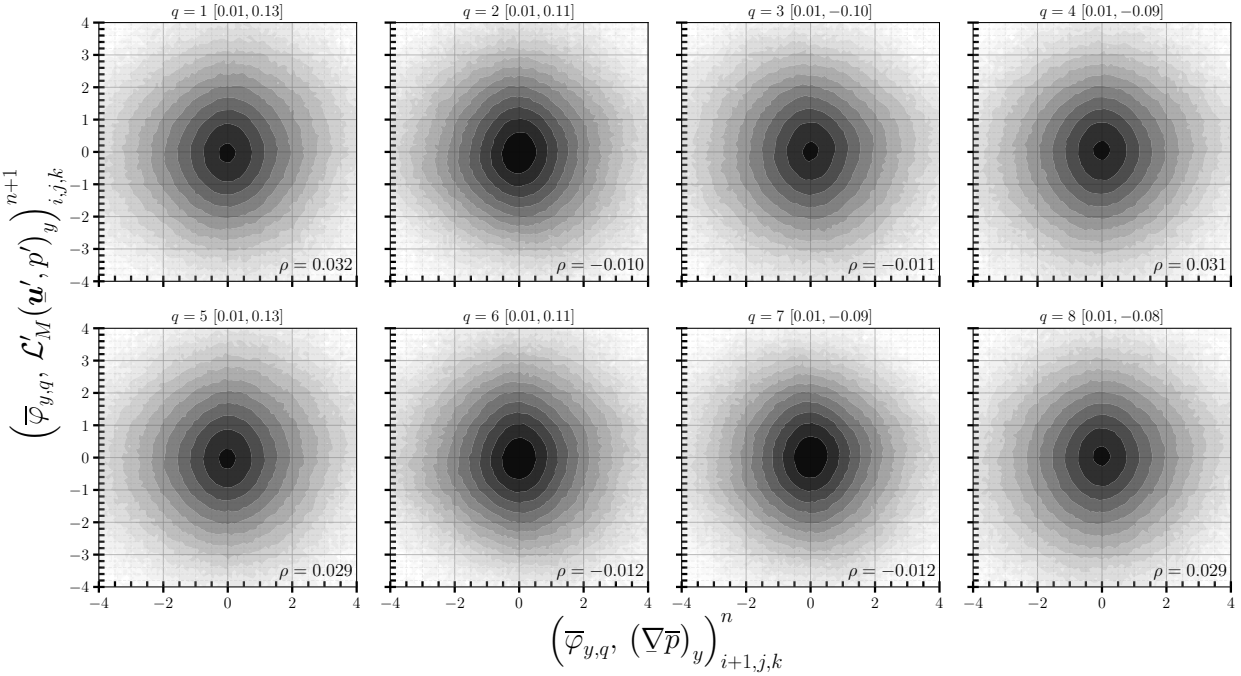


FIGURE A.142 Histograms of the t^n $(i+1, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

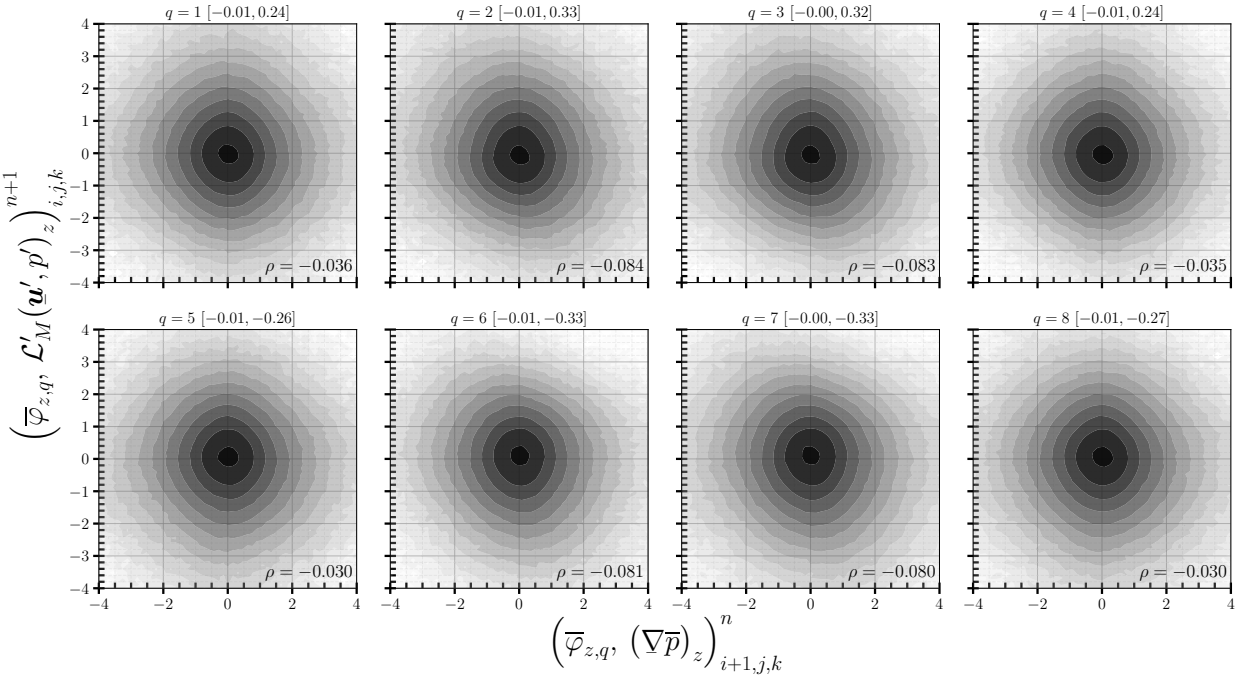


FIGURE A.143 Histograms of the t^n $(i+1, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

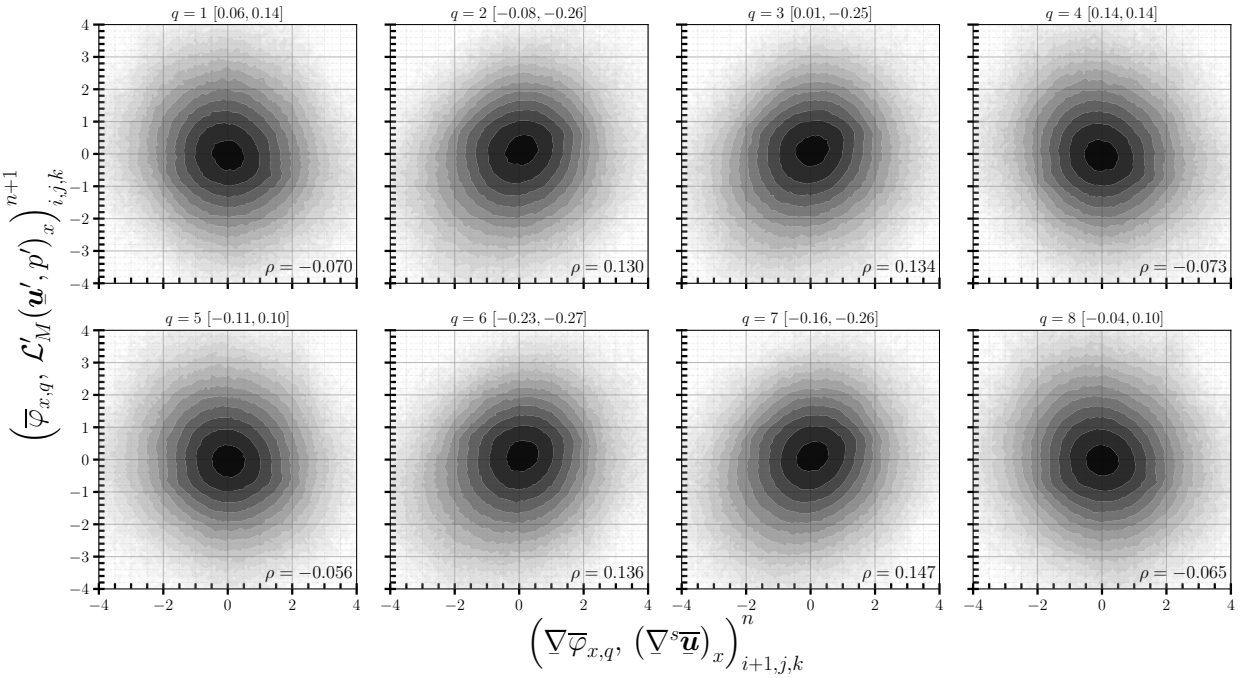


FIGURE A.144 Histograms of the t^n $(i+1, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

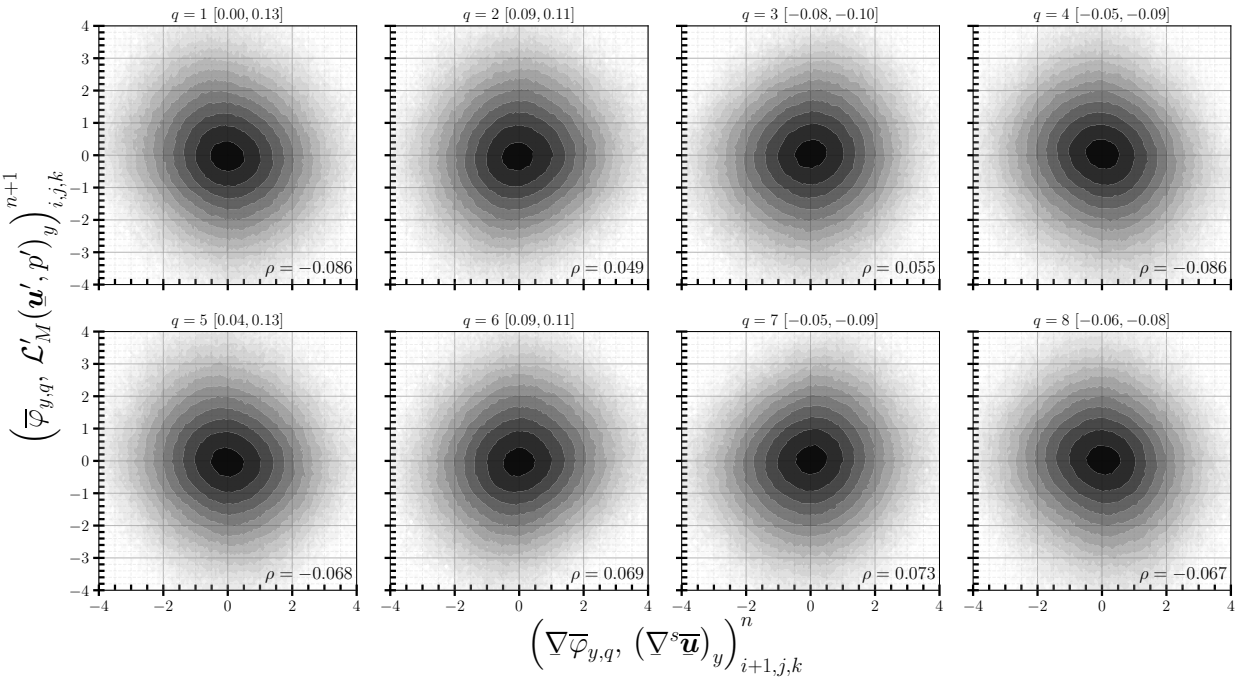


FIGURE A.145 Histograms of the t^n $(i+1, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

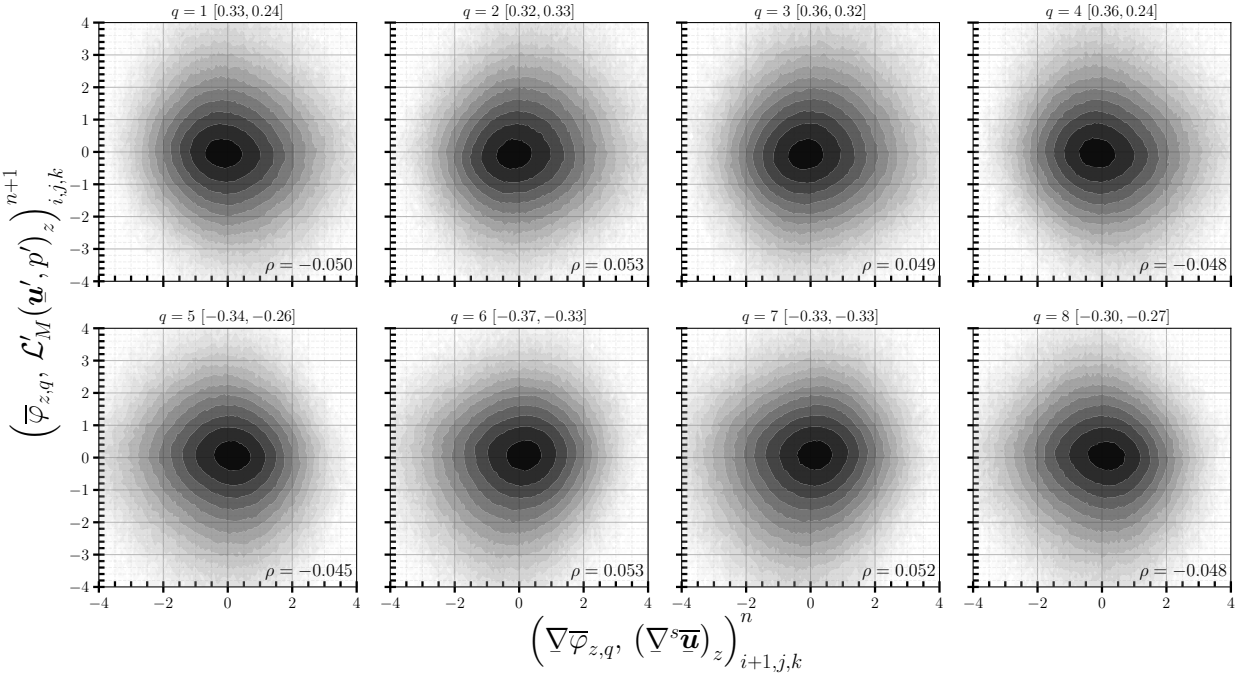


FIGURE A.146 Histograms of the t^n $(i+1, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.7 $(i, j-1, k)$ Histograms

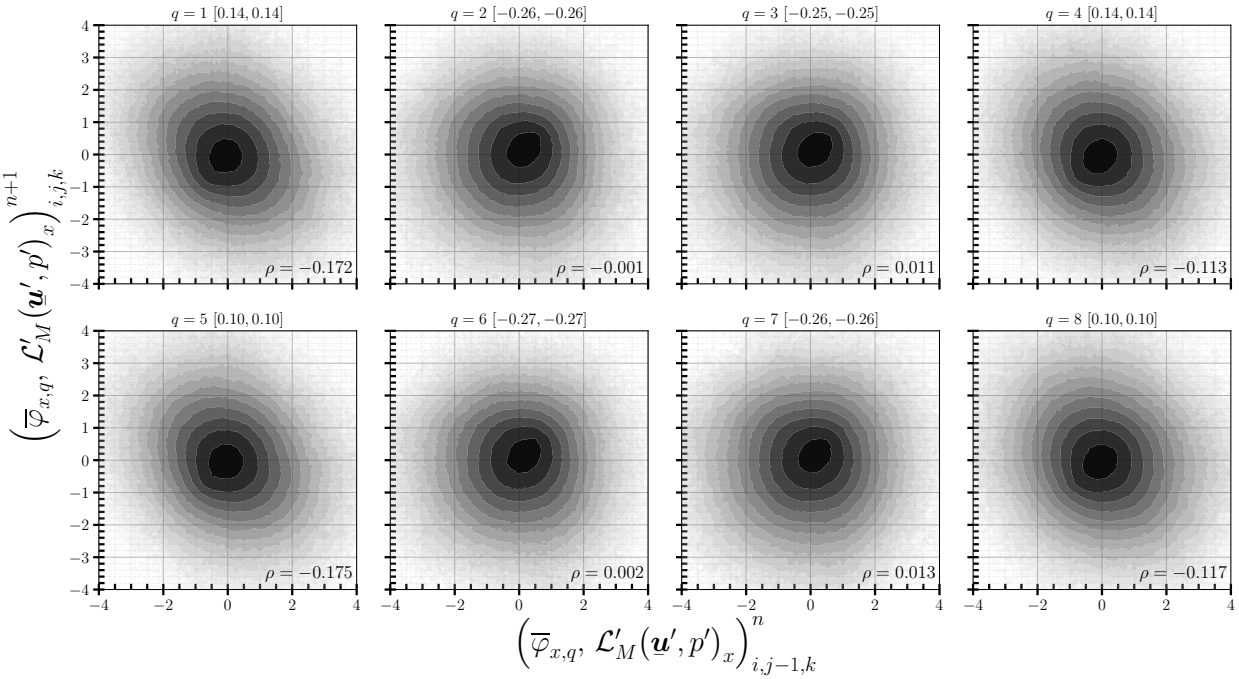


FIGURE A.147 Histograms of the t^n $(i, j-1, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

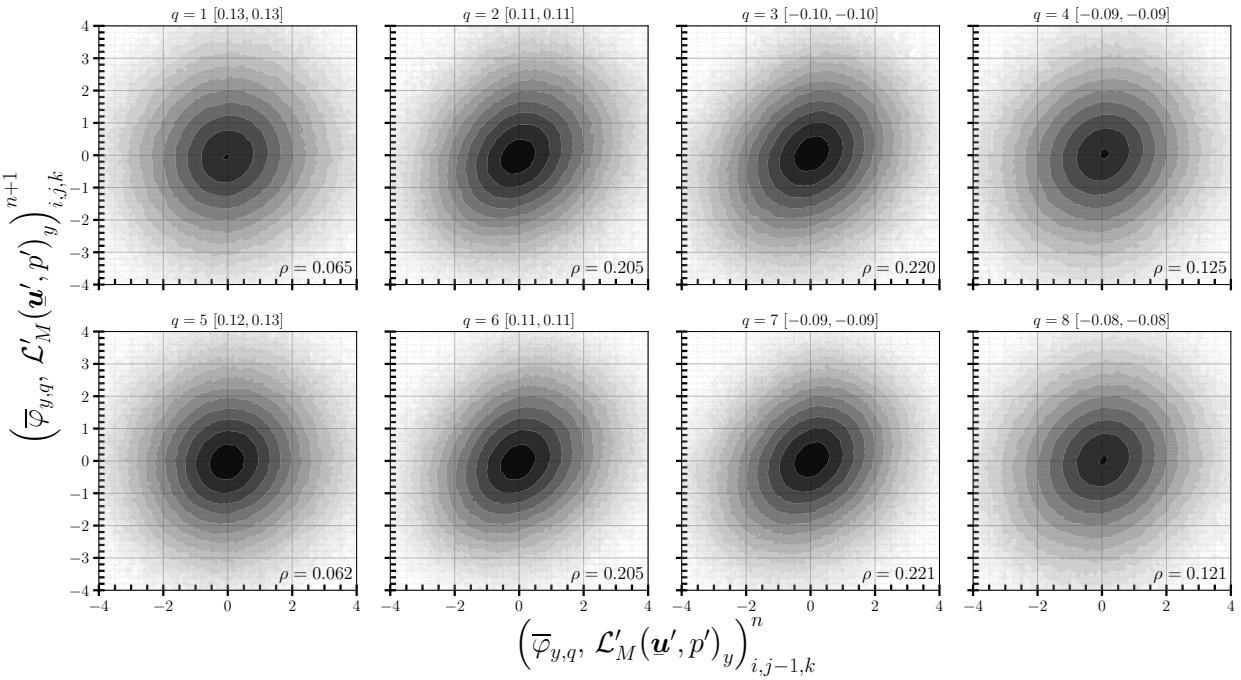


FIGURE A.148 Histograms of the t^n $(i, j-1, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

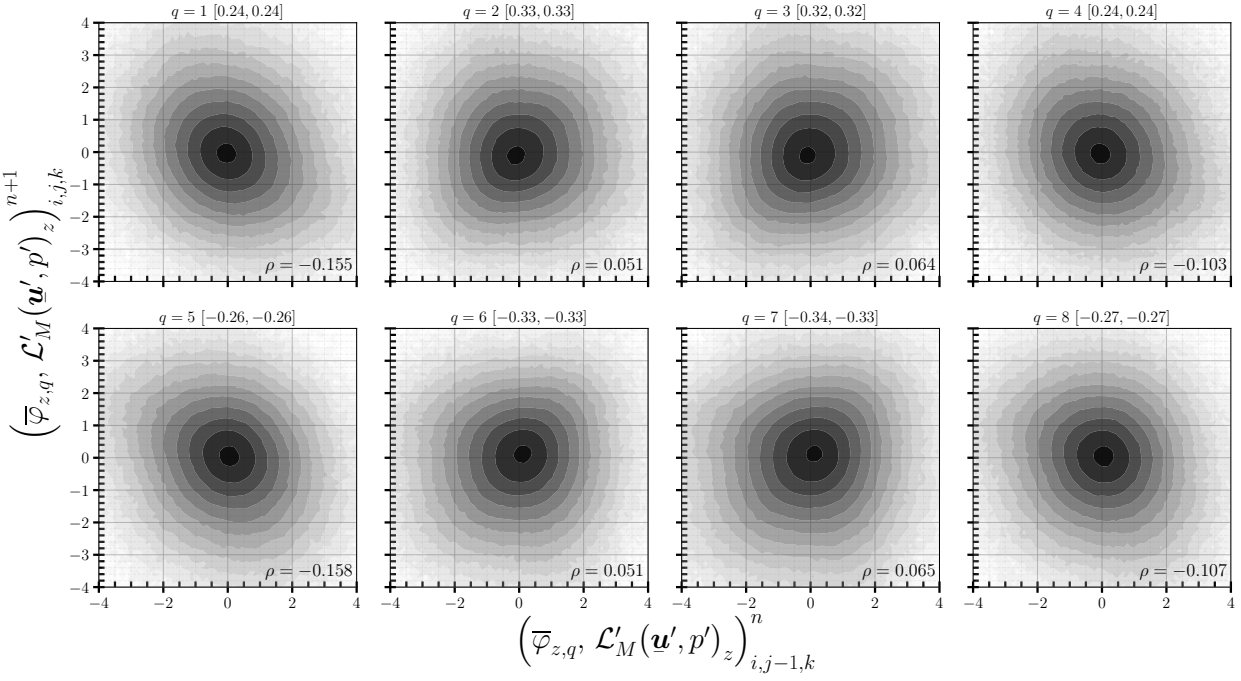


FIGURE A.149 Histograms of the t^n $(i, j-1, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

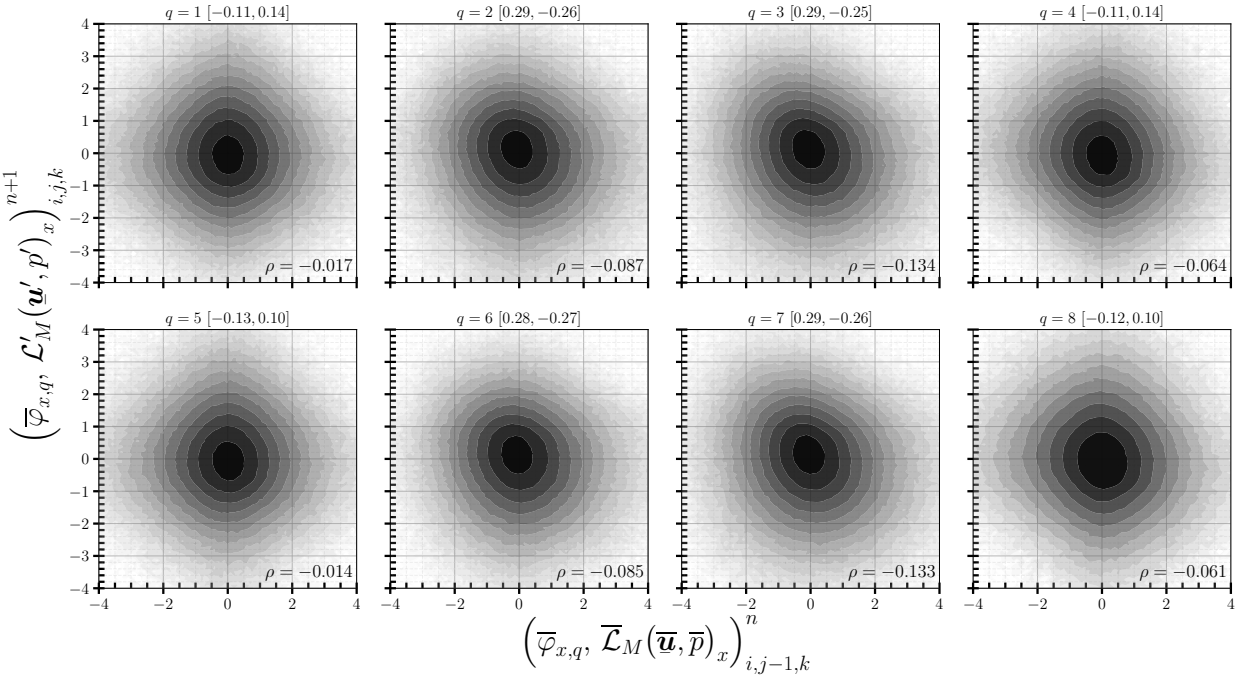


FIGURE A.150 Histograms of the t^n $(i, j-1, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

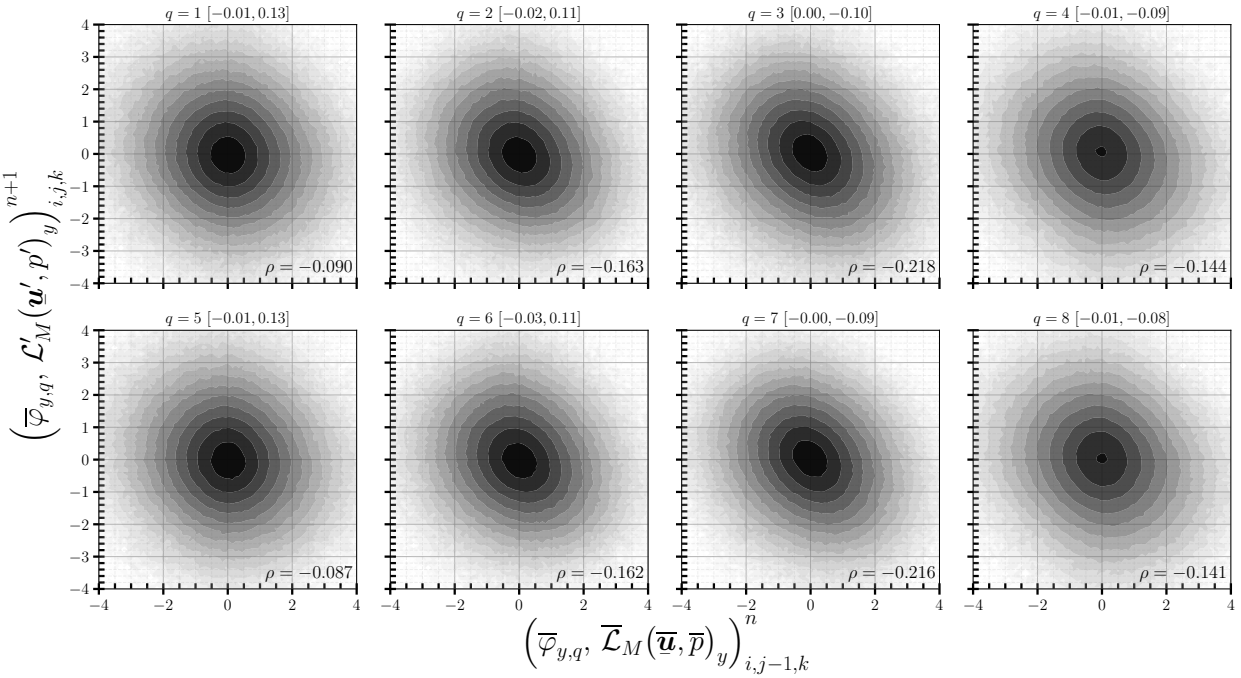


FIGURE A.151 Histograms of the t^n $(i, j-1, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

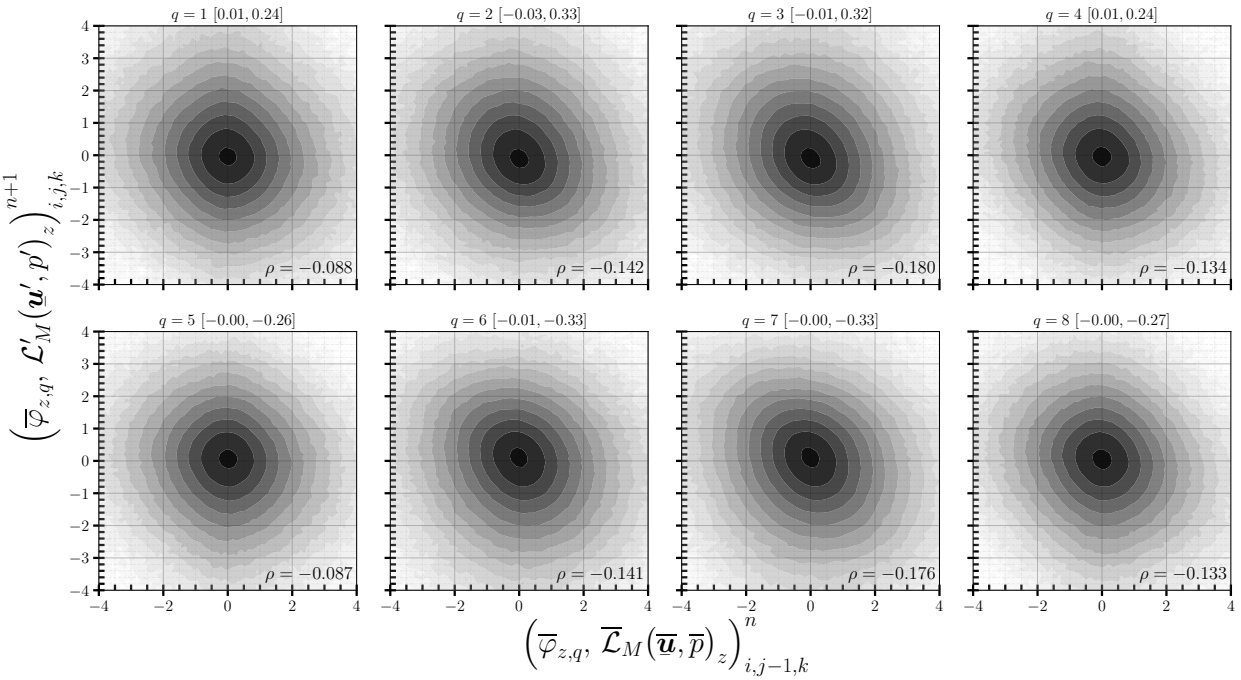


FIGURE A.152 Histograms of the t^n $(i, j-1, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

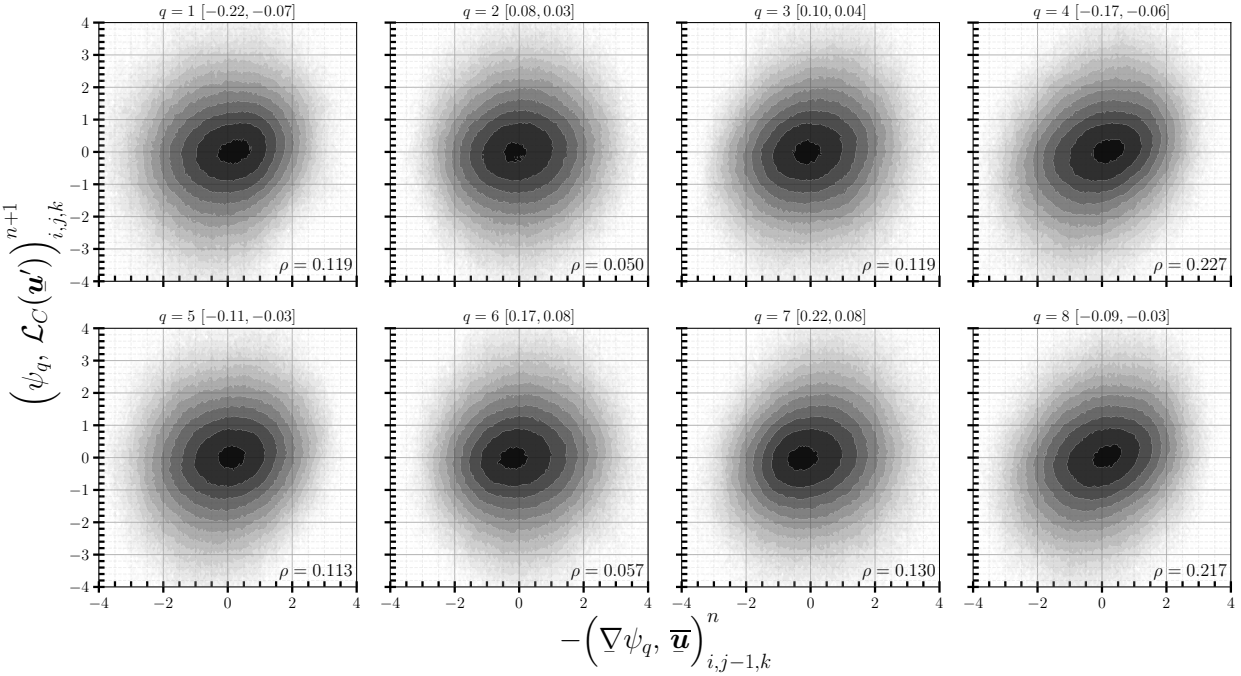


FIGURE A.153 Histograms of the t^n $(i, j-1, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

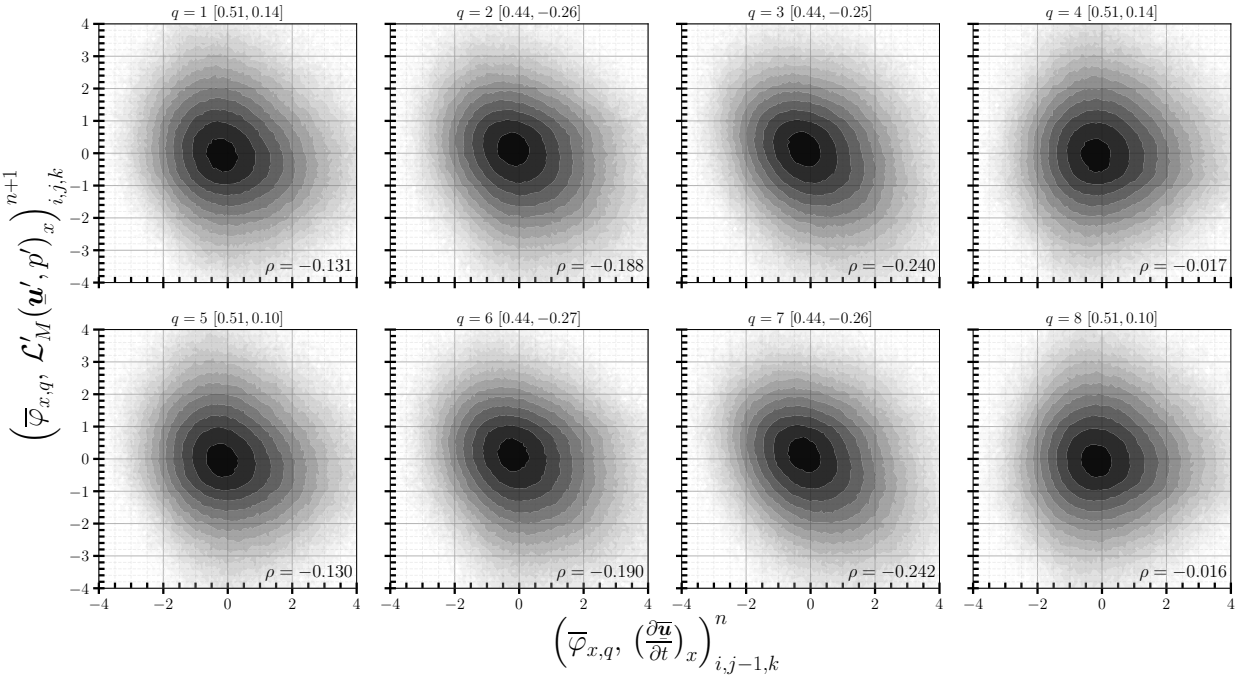


FIGURE A.154 Histograms of the t^n $(i, j-1, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

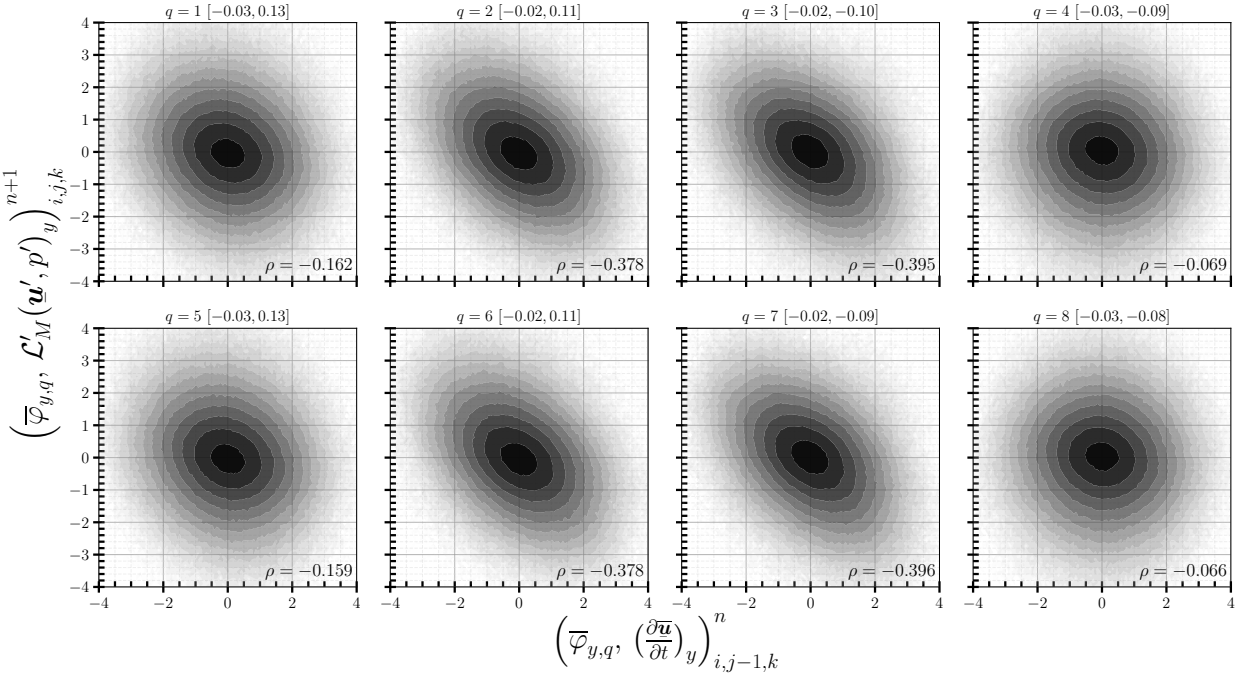


FIGURE A.155 Histograms of the t^n $(i, j-1, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

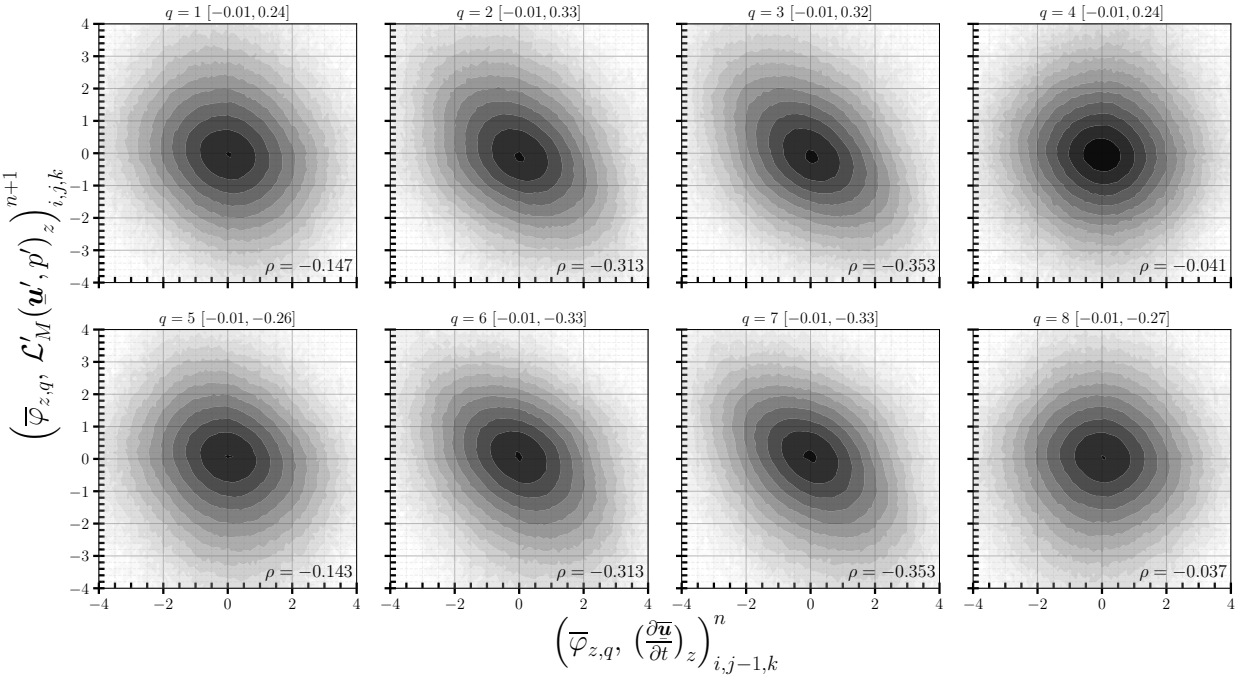


FIGURE A.156 Histograms of the t^n $(i, j-1, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

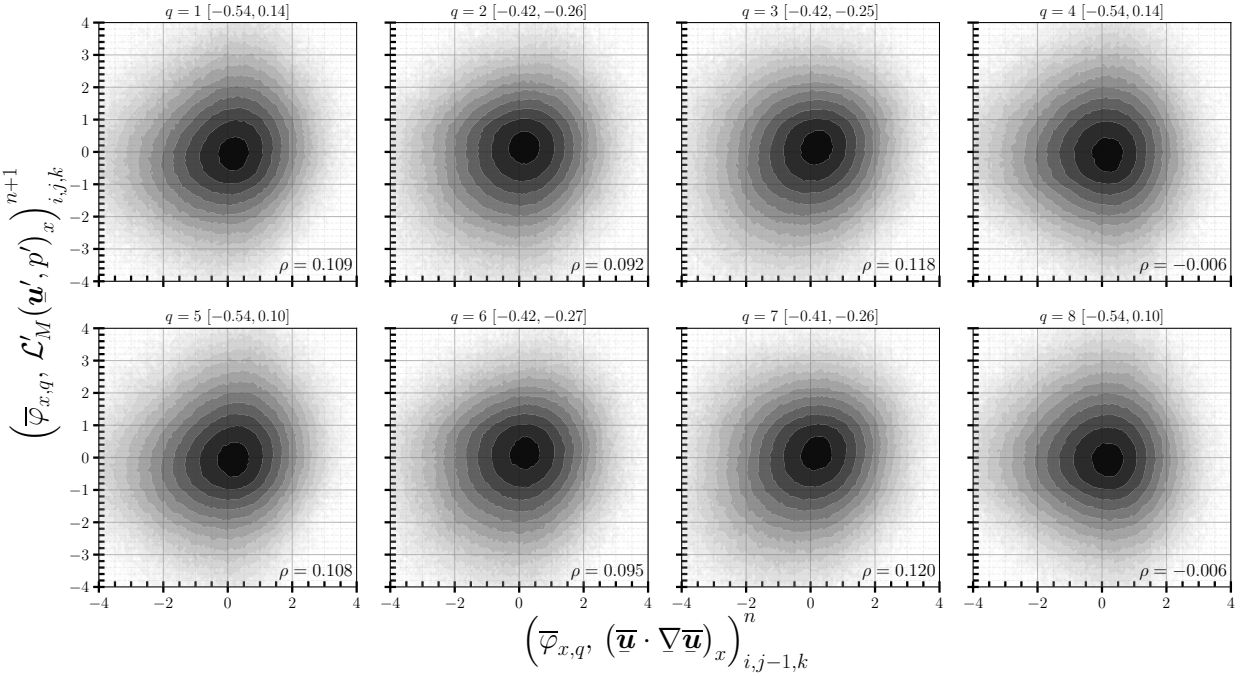


FIGURE A.157 Histograms of the t^n $(i, j-1, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

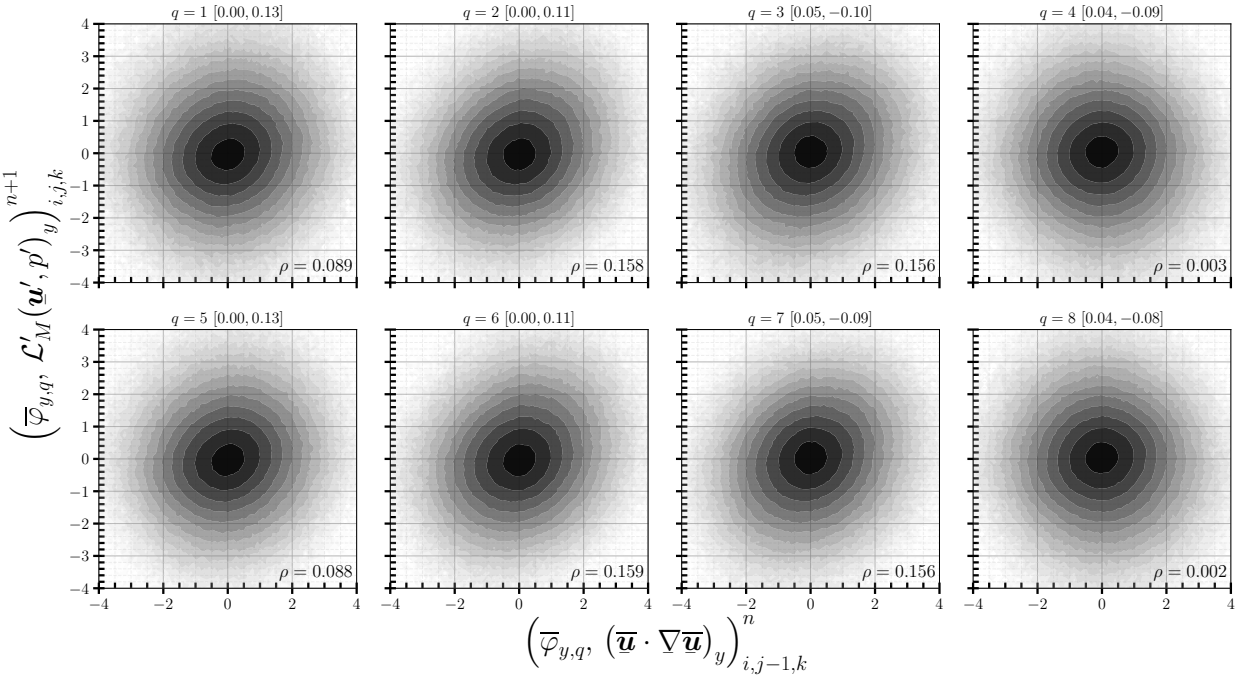


FIGURE A.158 Histograms of the t^n $(i, j-1, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

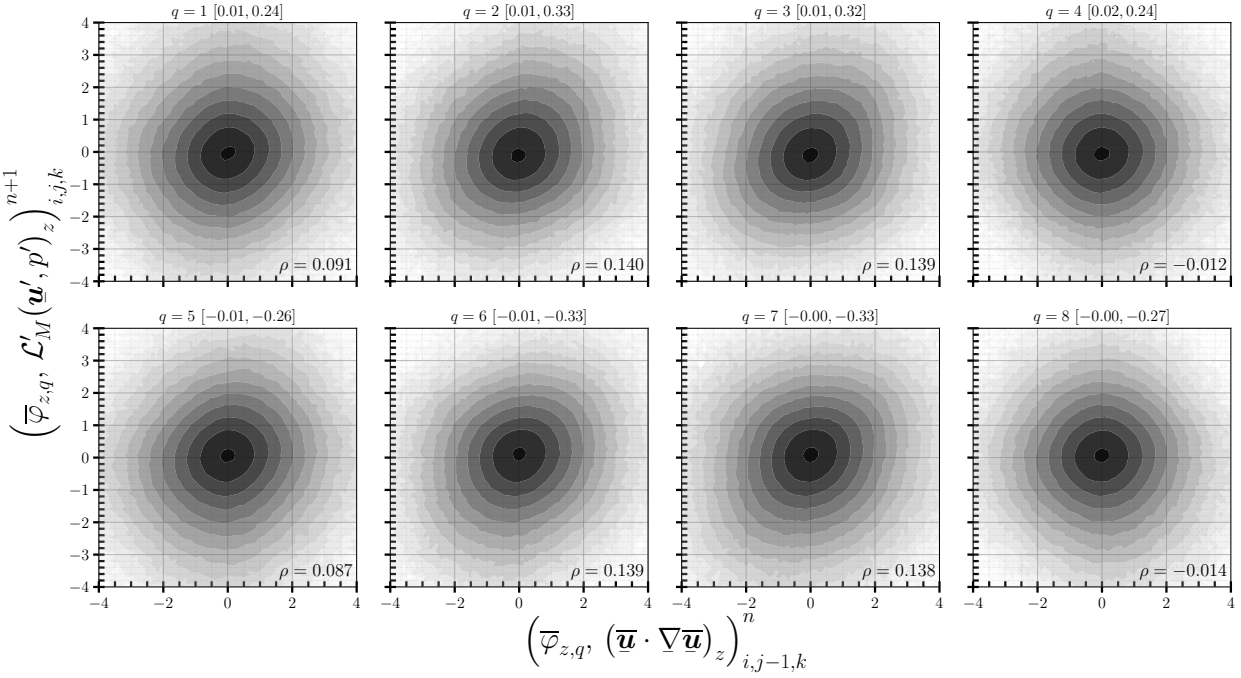


FIGURE A.159 Histograms of the t^n $(i, j-1, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

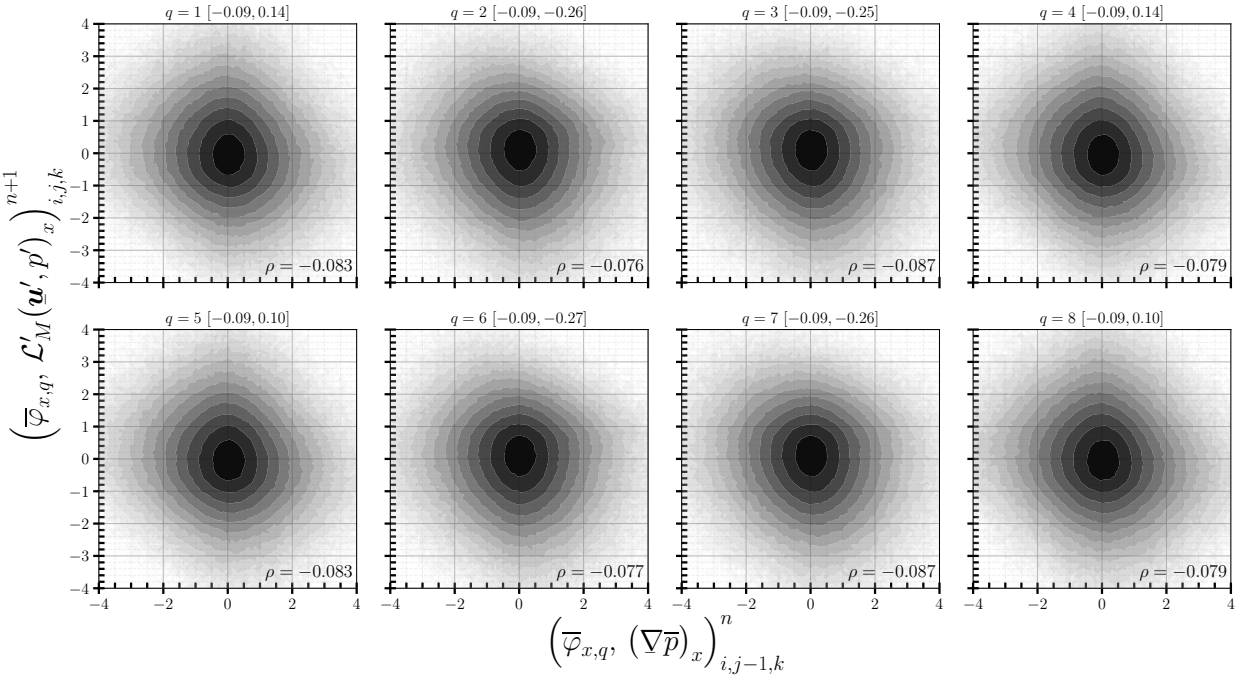


FIGURE A.160 Histograms of the t^n $(i, j-1, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

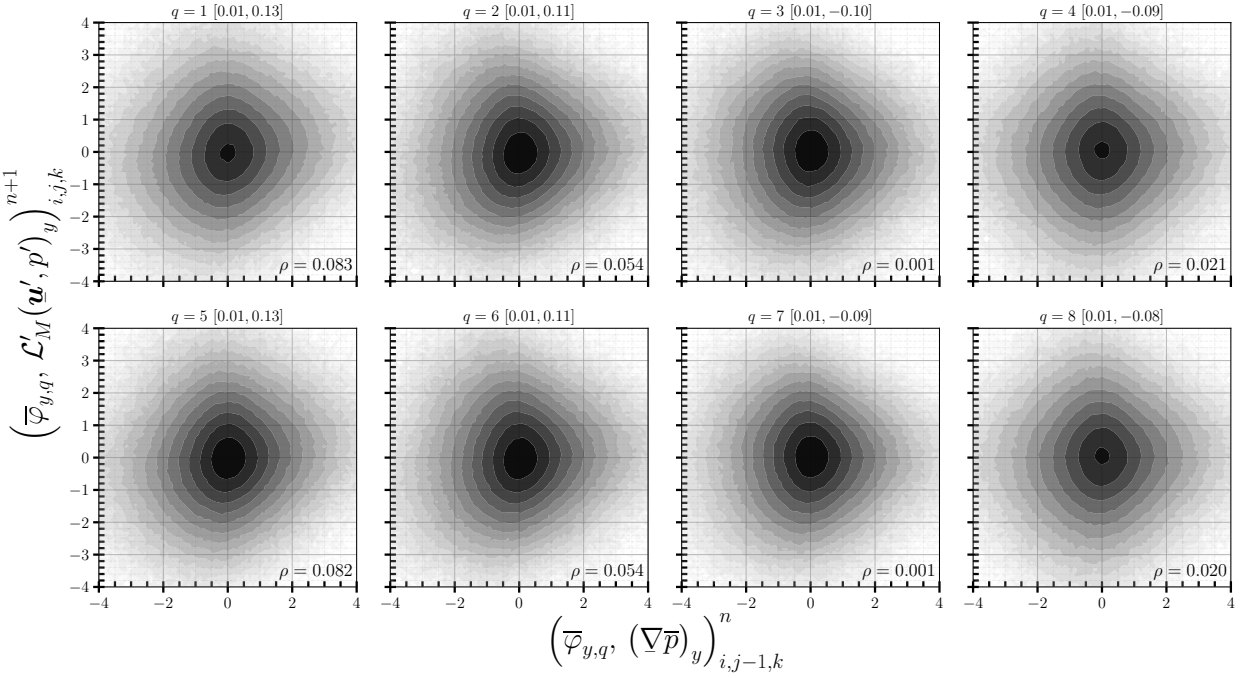


FIGURE A.161 Histograms of the t^n $(i, j-1, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

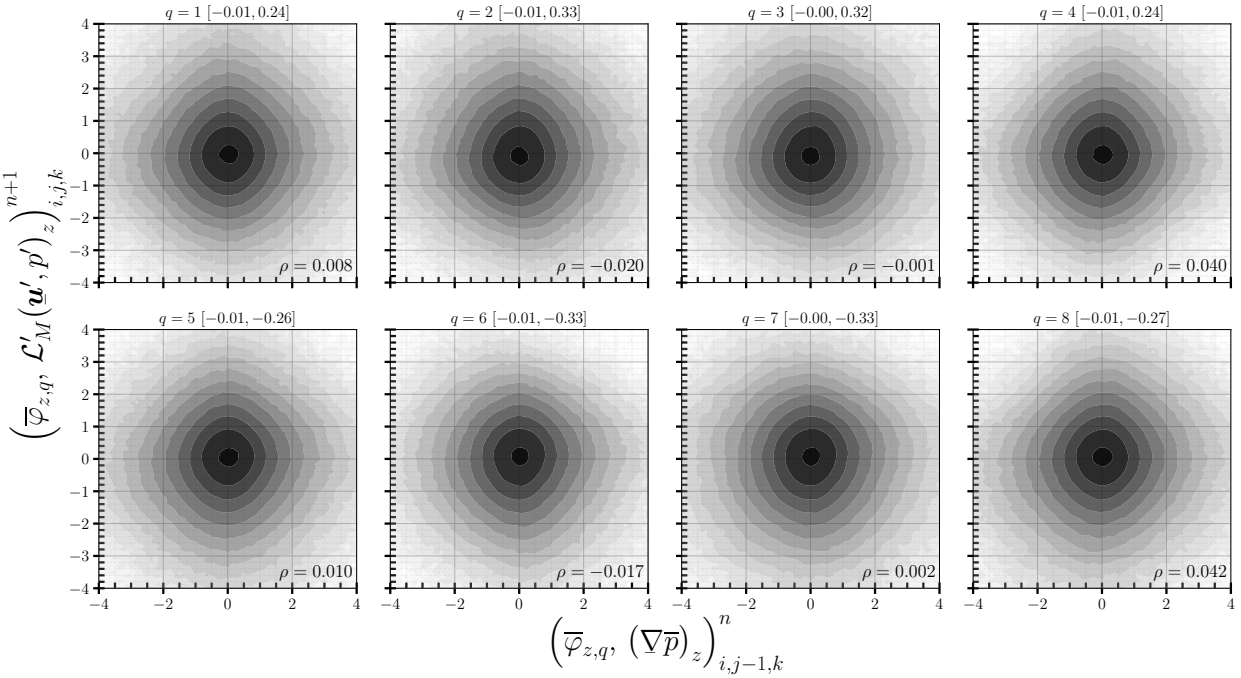


FIGURE A.162 Histograms of the t^n $(i, j-1, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

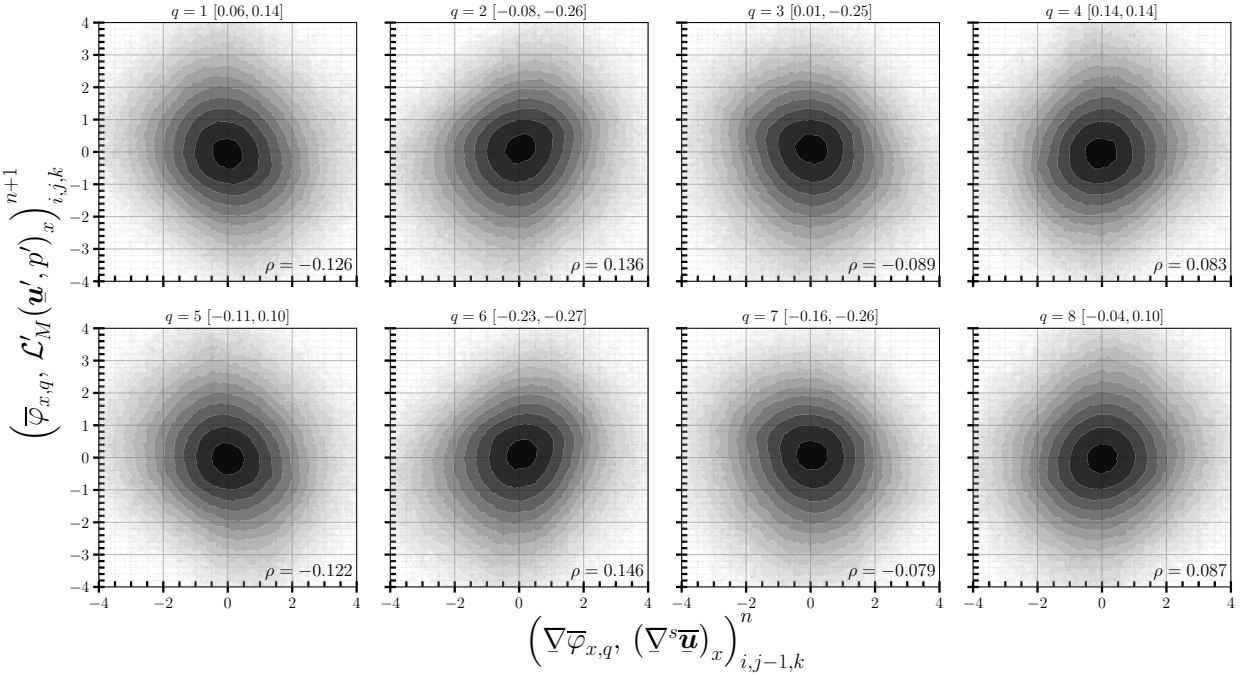


FIGURE A.163 Histograms of the t^n $(i, j-1, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

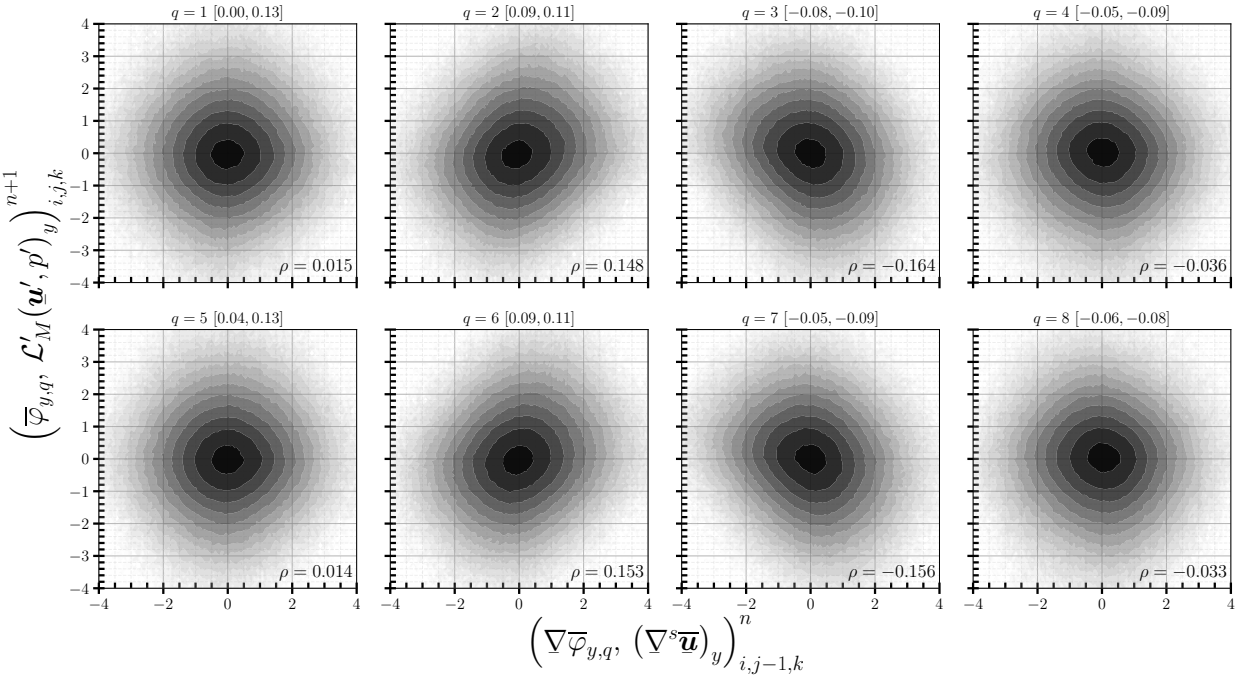


FIGURE A.164 Histograms of the t^n $(i, j-1, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

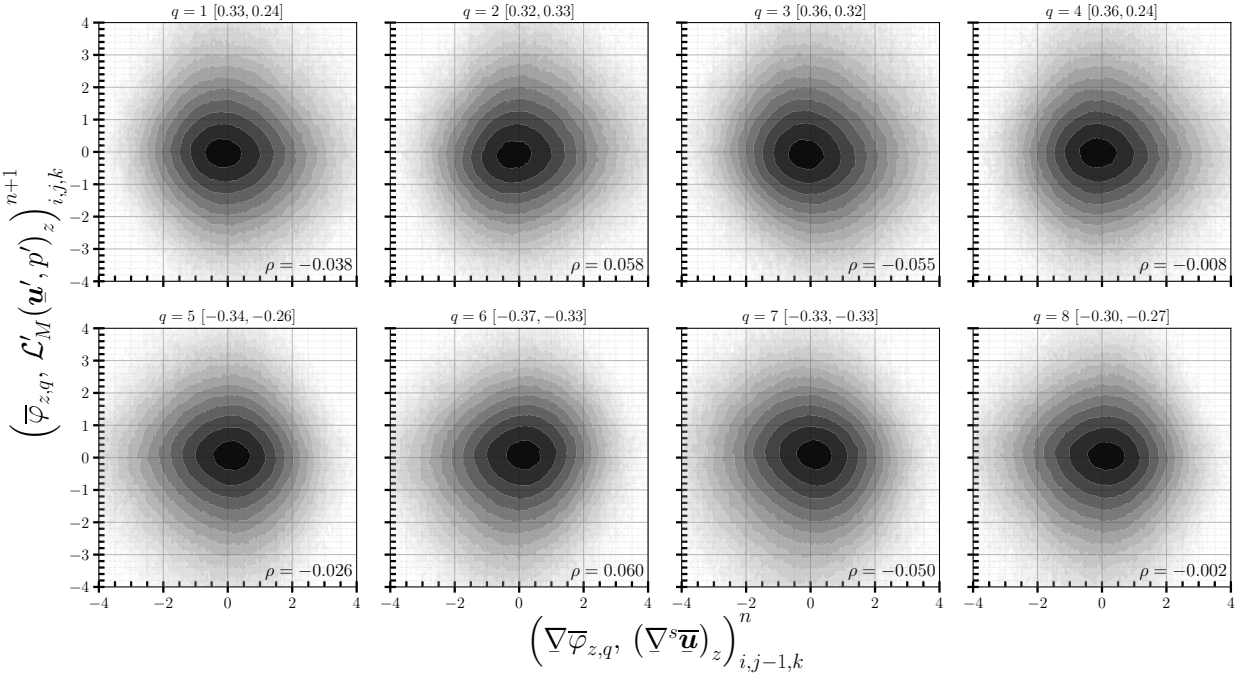


FIGURE A.165 Histograms of the t^n $(i, j-1, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.8 $(i, j+1, k)$ Histograms

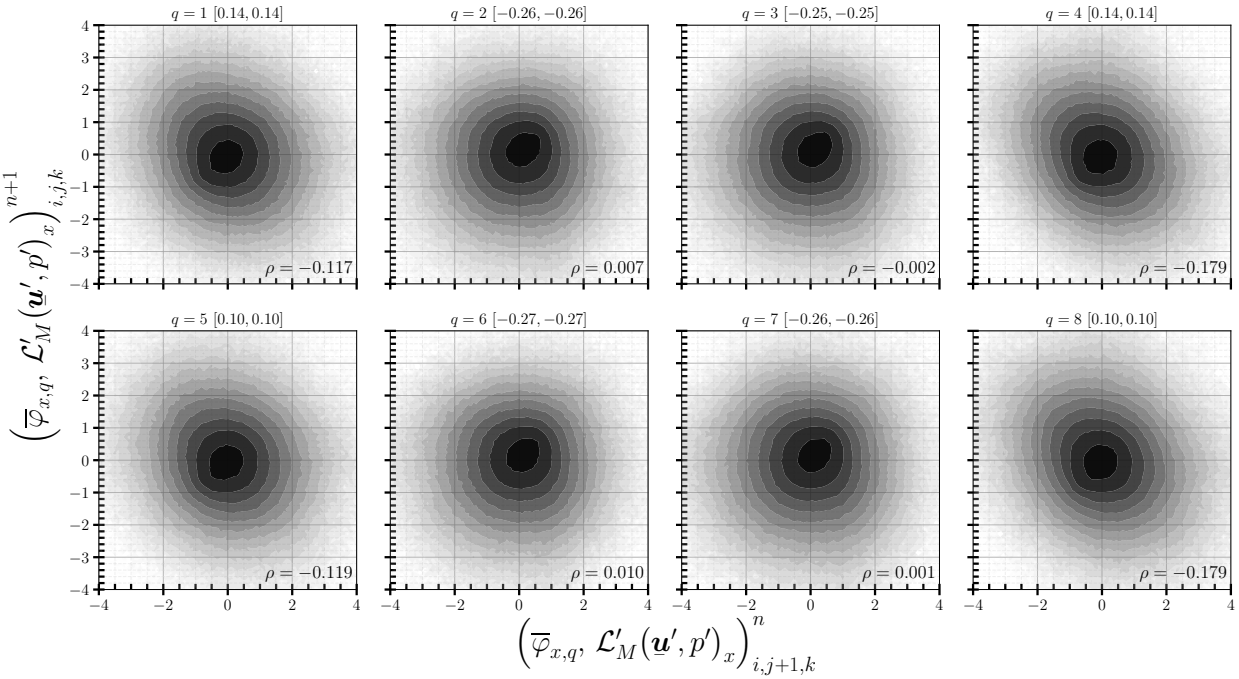


FIGURE A.166 Histograms of the t^n $(i, j+1, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

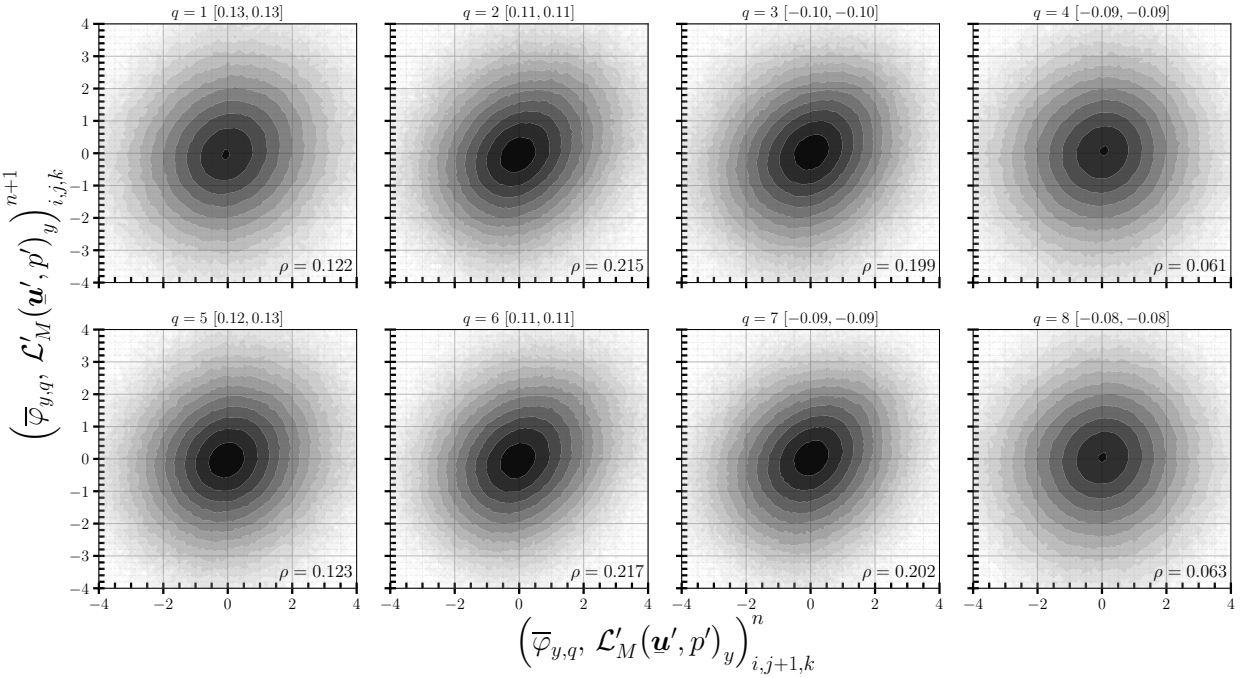


FIGURE A.167 Histograms of the t^n $(i, j+1, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

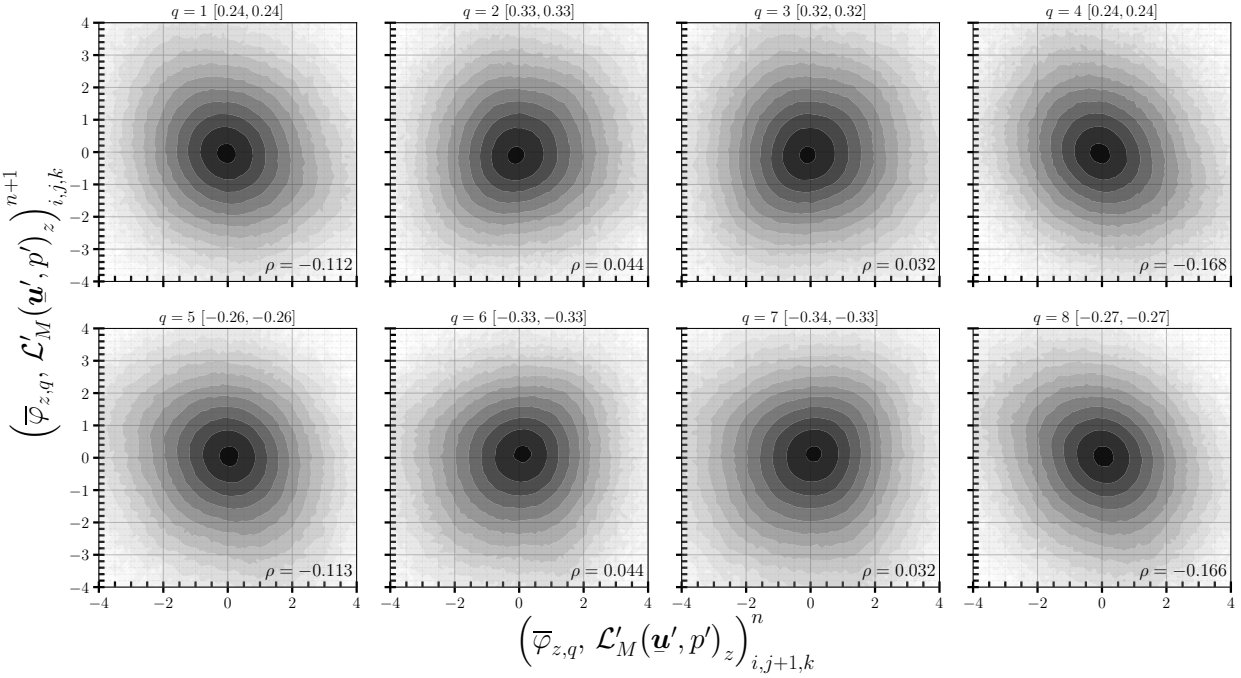


FIGURE A.168 Histograms of the t^n $(i, j+1, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

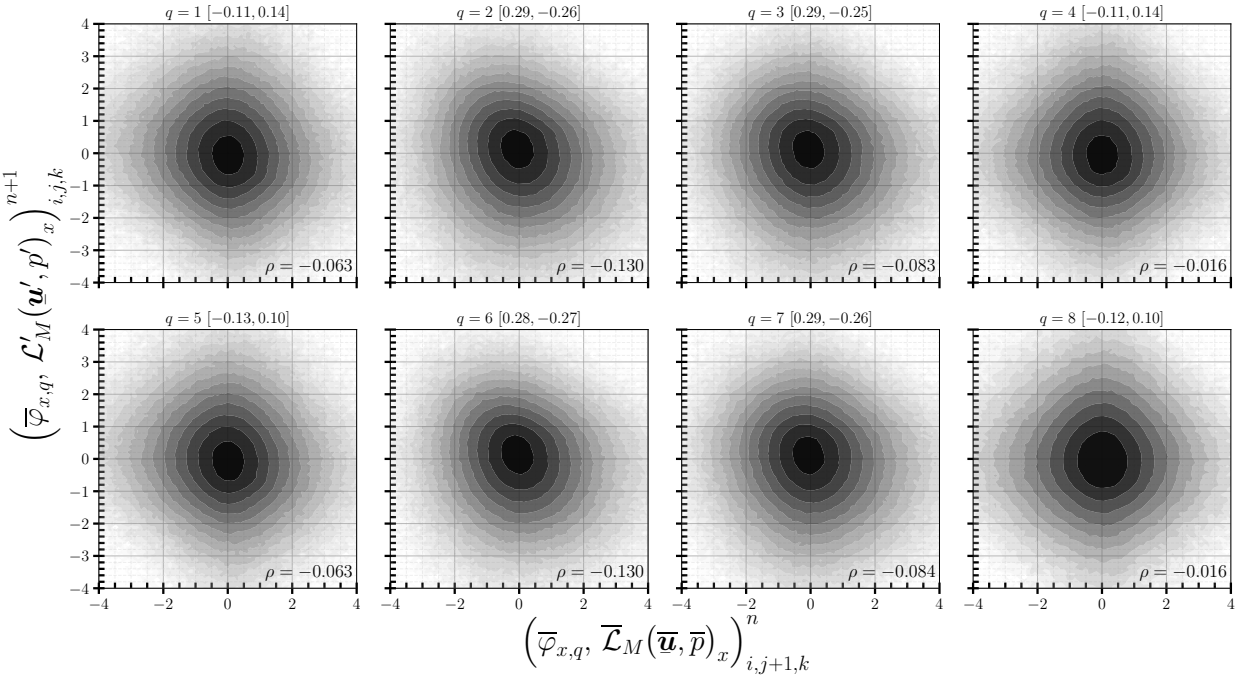


FIGURE A.169 Histograms of the t^n $(i, j+1, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

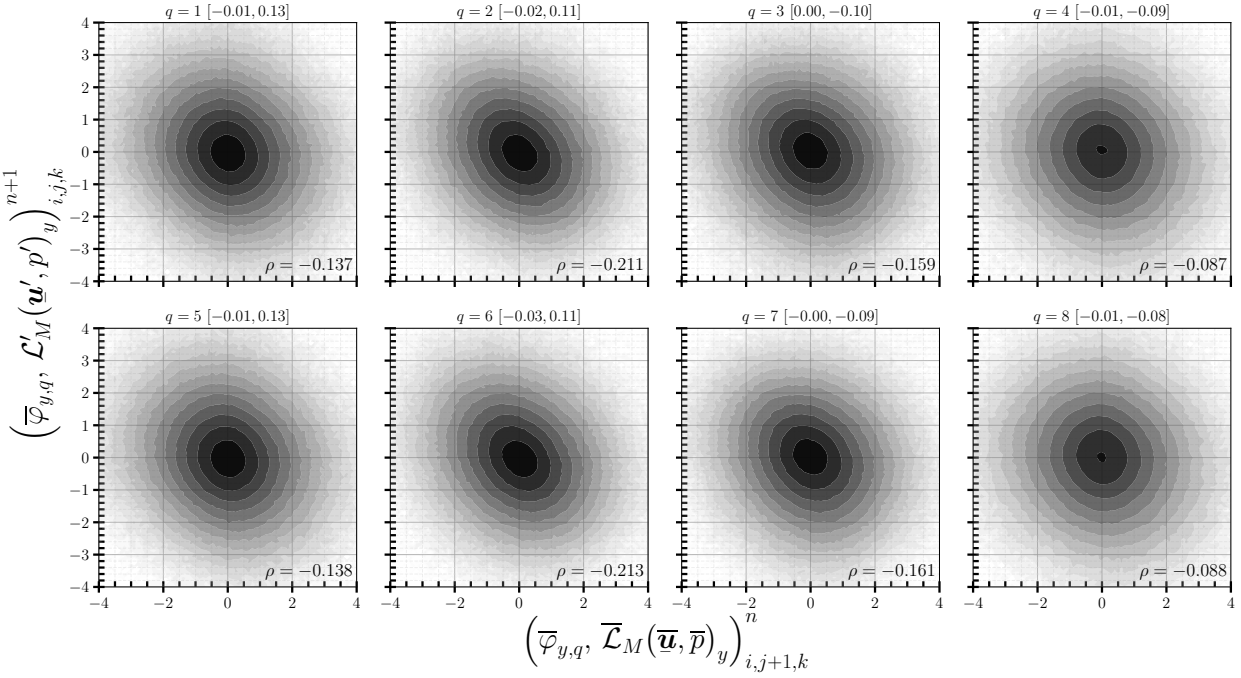


FIGURE A.170 Histograms of the t^n $(i, j+1, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

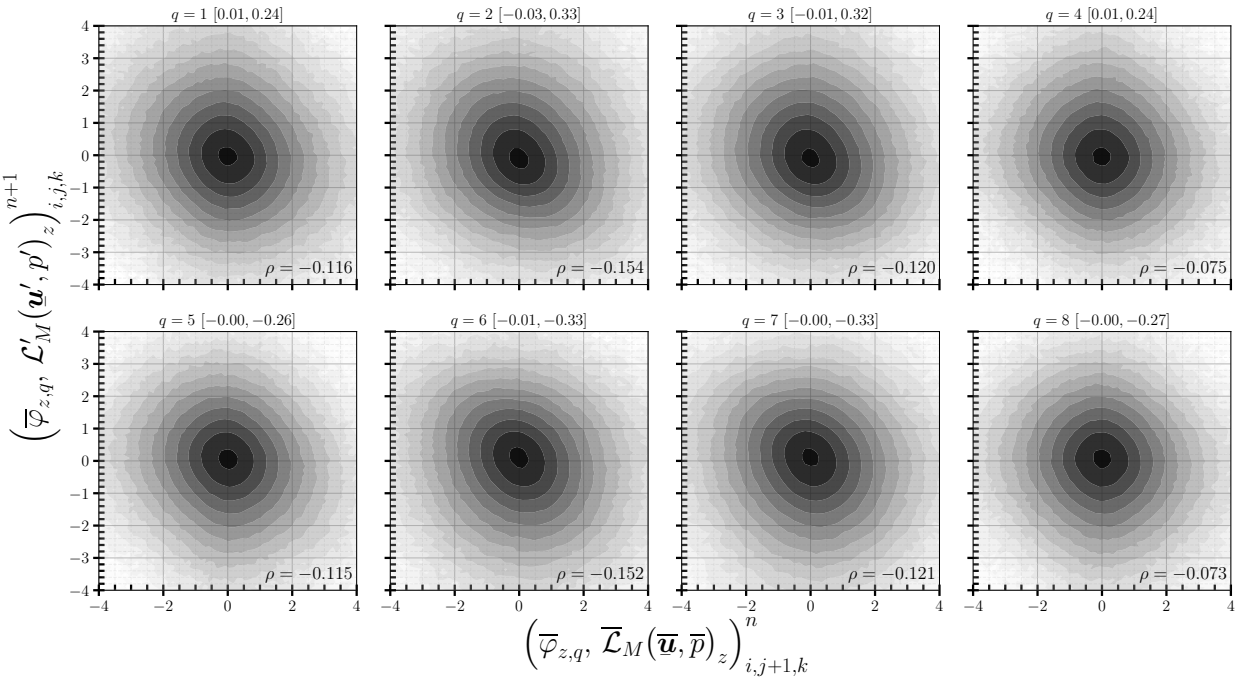


FIGURE A.171 Histograms of the t^n $(i, j+1, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

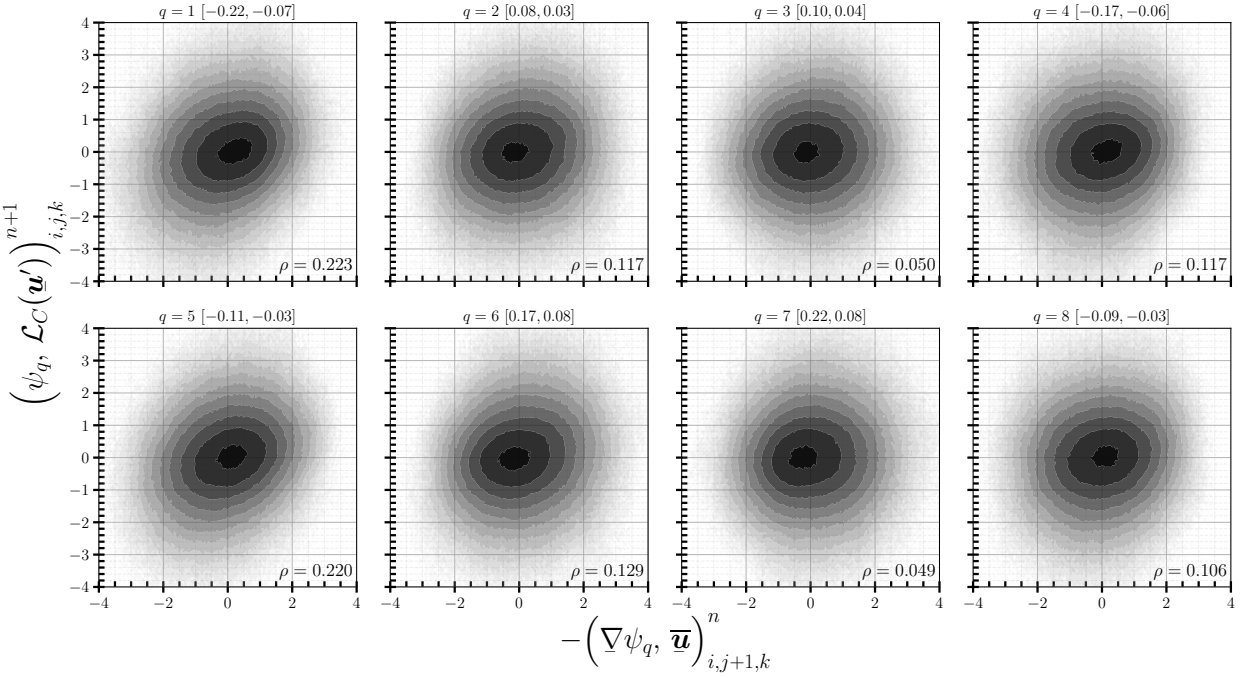


FIGURE A.172 Histograms of the t^n $(i, j+1, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

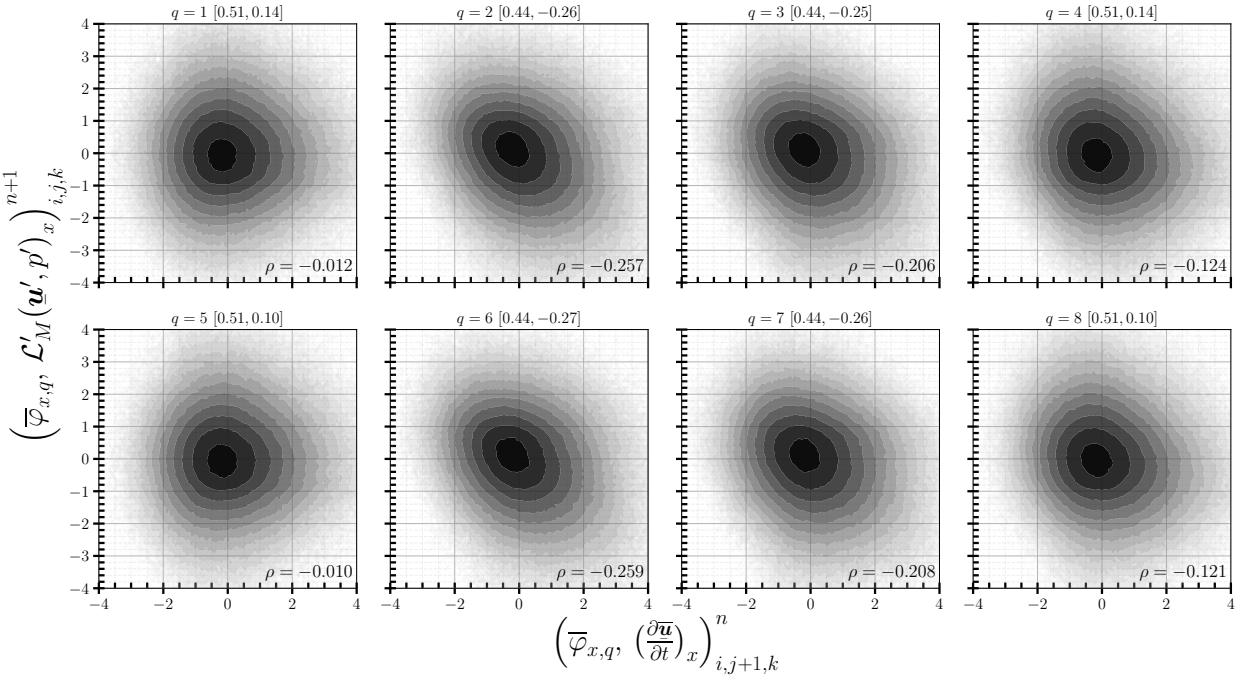


FIGURE A.173 Histograms of the t^n $(i, j+1, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

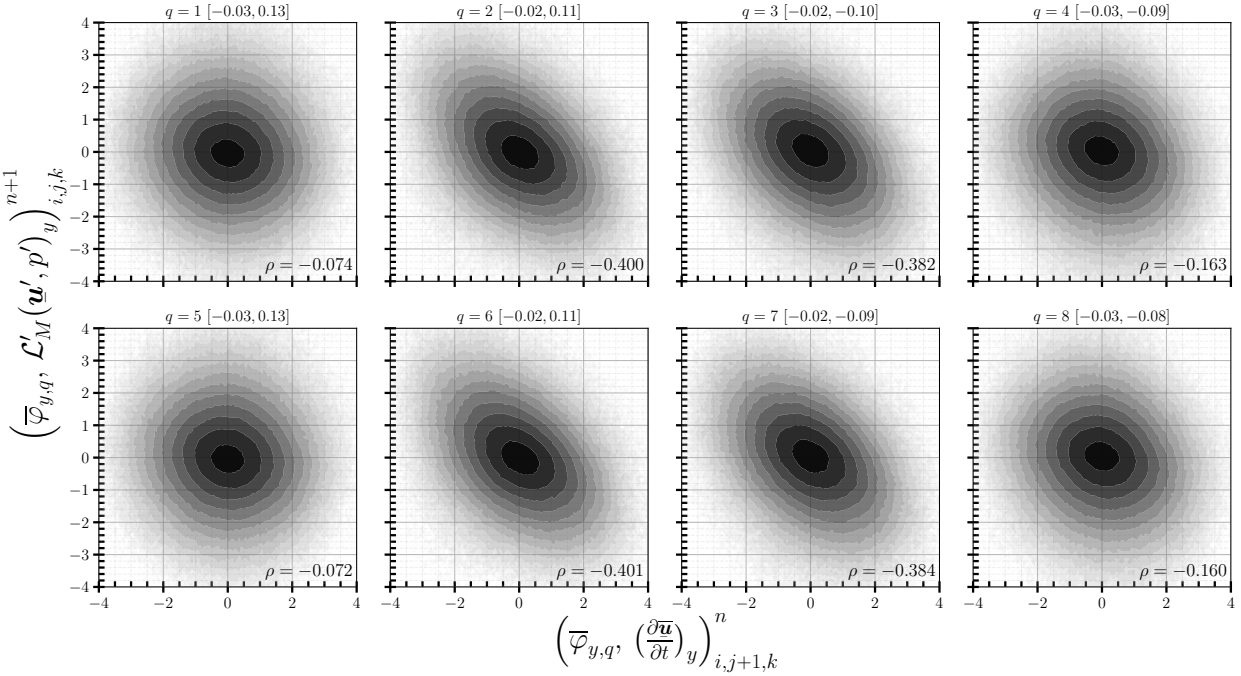


FIGURE A.174 Histograms of the t^n $(i, j+1, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

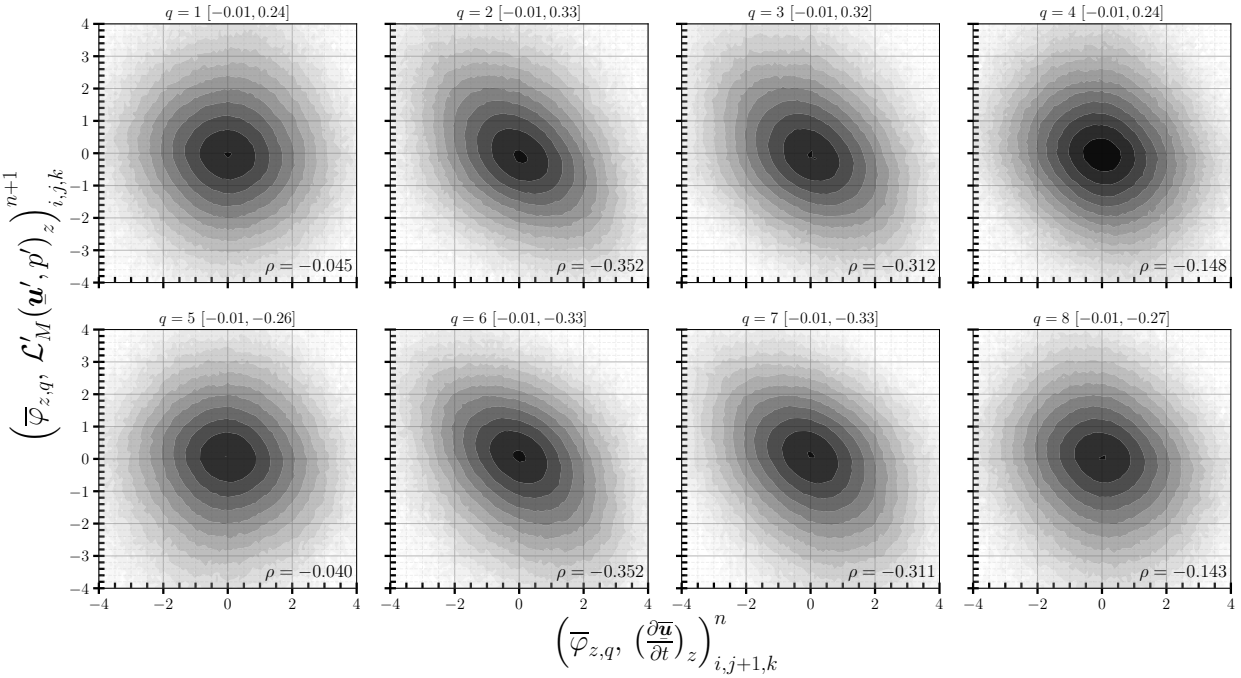


FIGURE A.175 Histograms of the t^n $(i, j+1, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

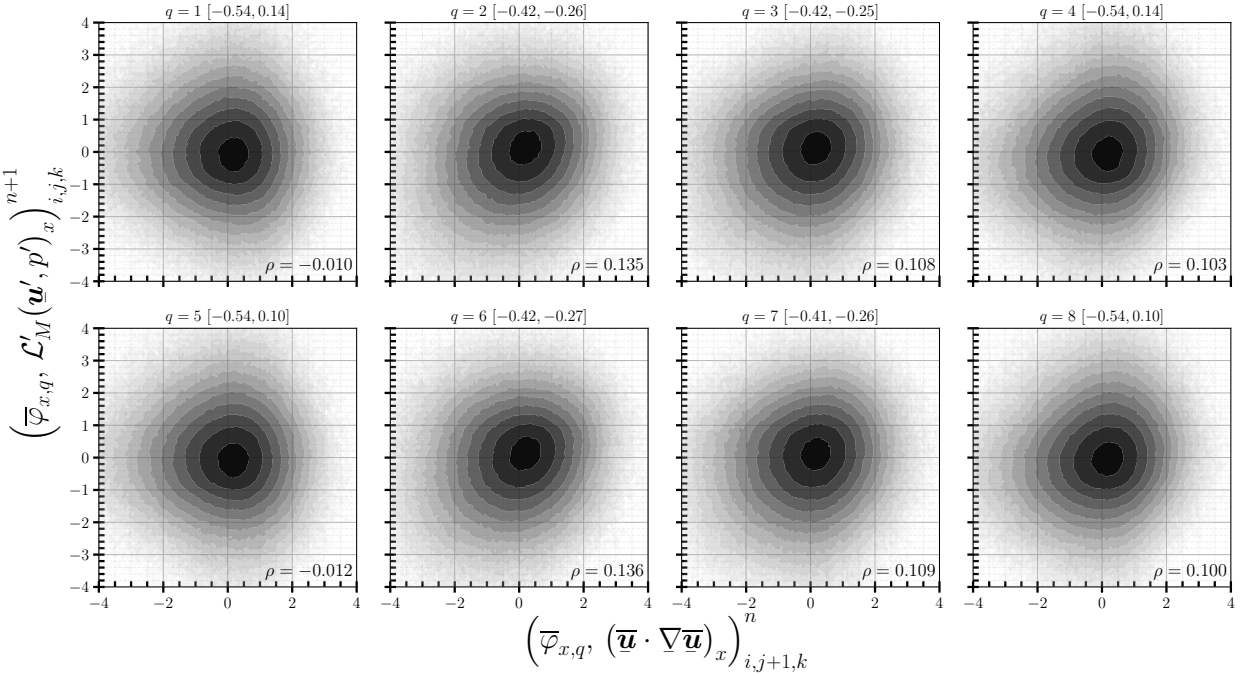


FIGURE A.176 Histograms of the t^n $(i, j+1, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

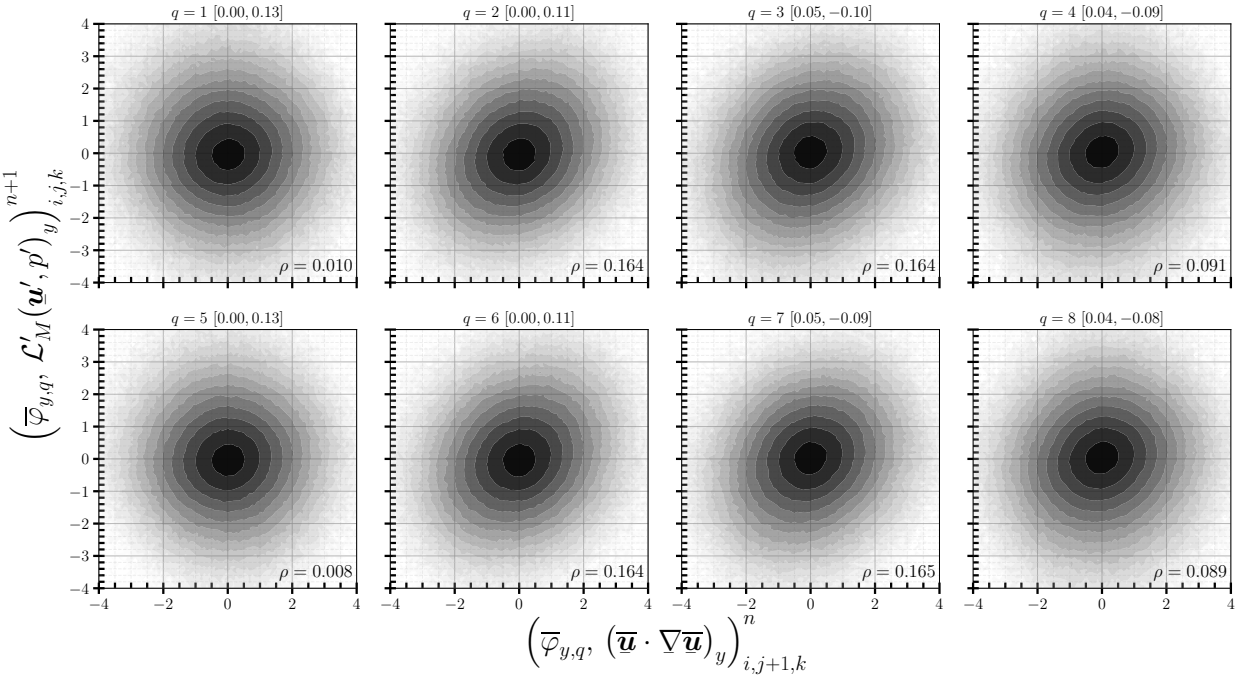


FIGURE A.177 Histograms of the t^n $(i, j+1, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

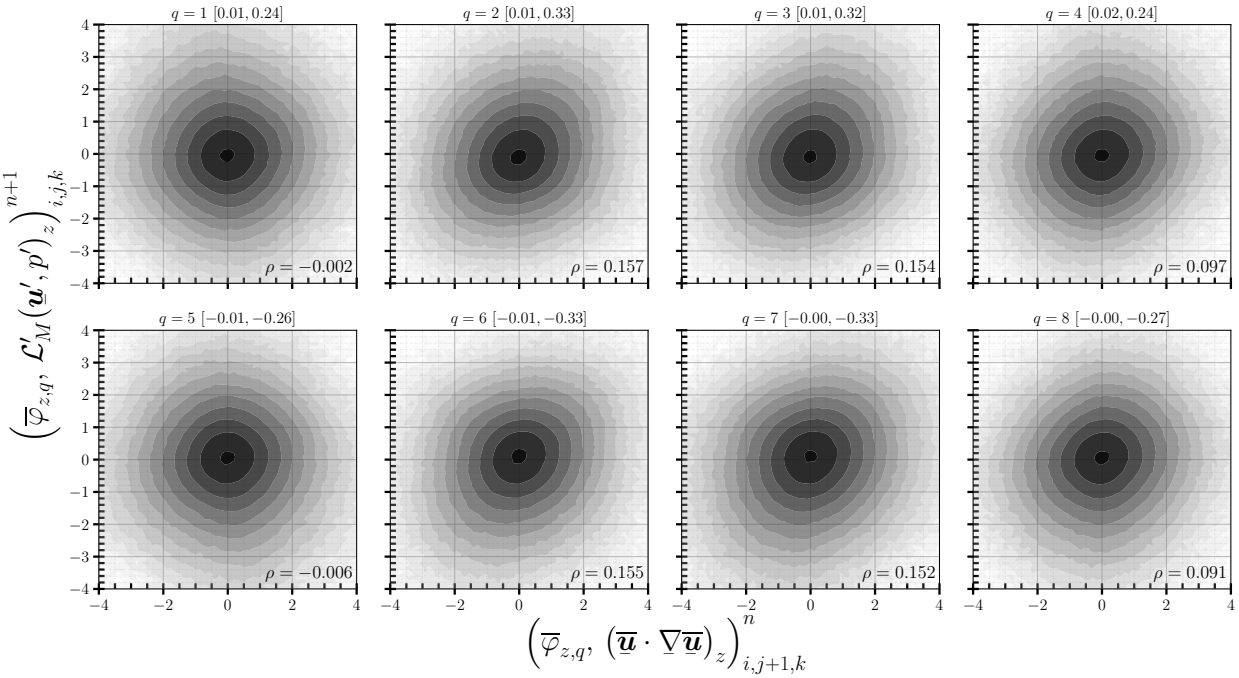


FIGURE A.178 Histograms of the t^n $(i, j+1, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

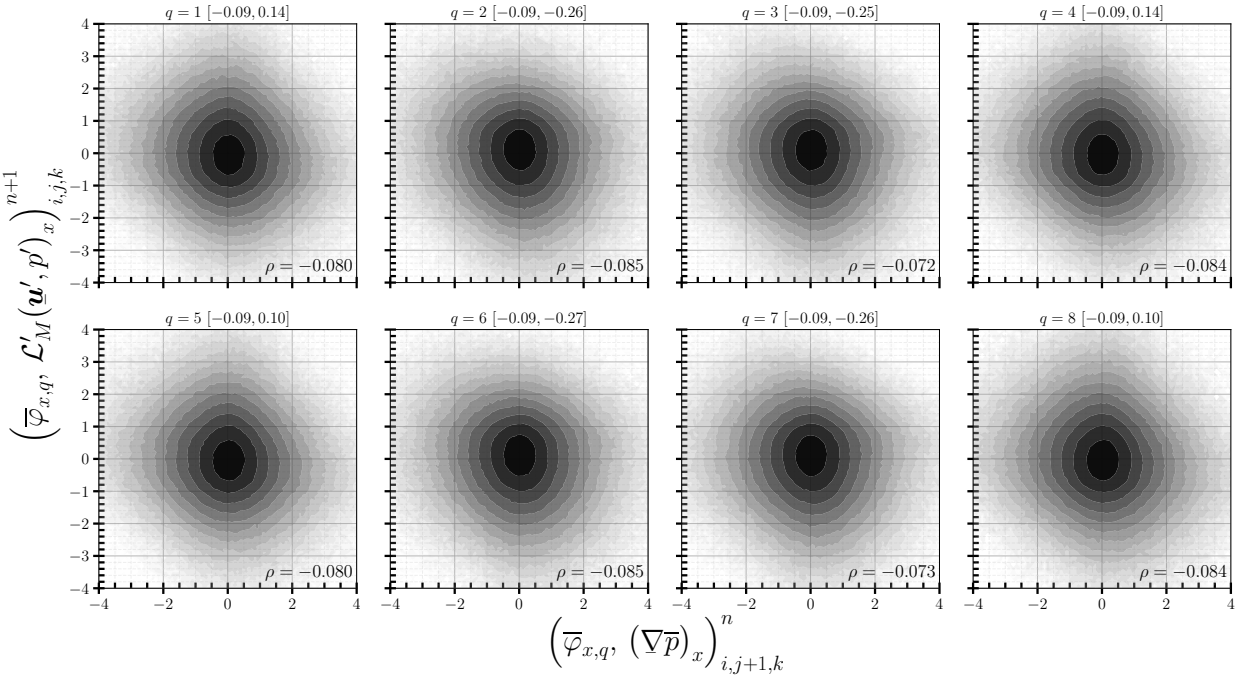


FIGURE A.179 Histograms of the t^n $(i, j+1, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

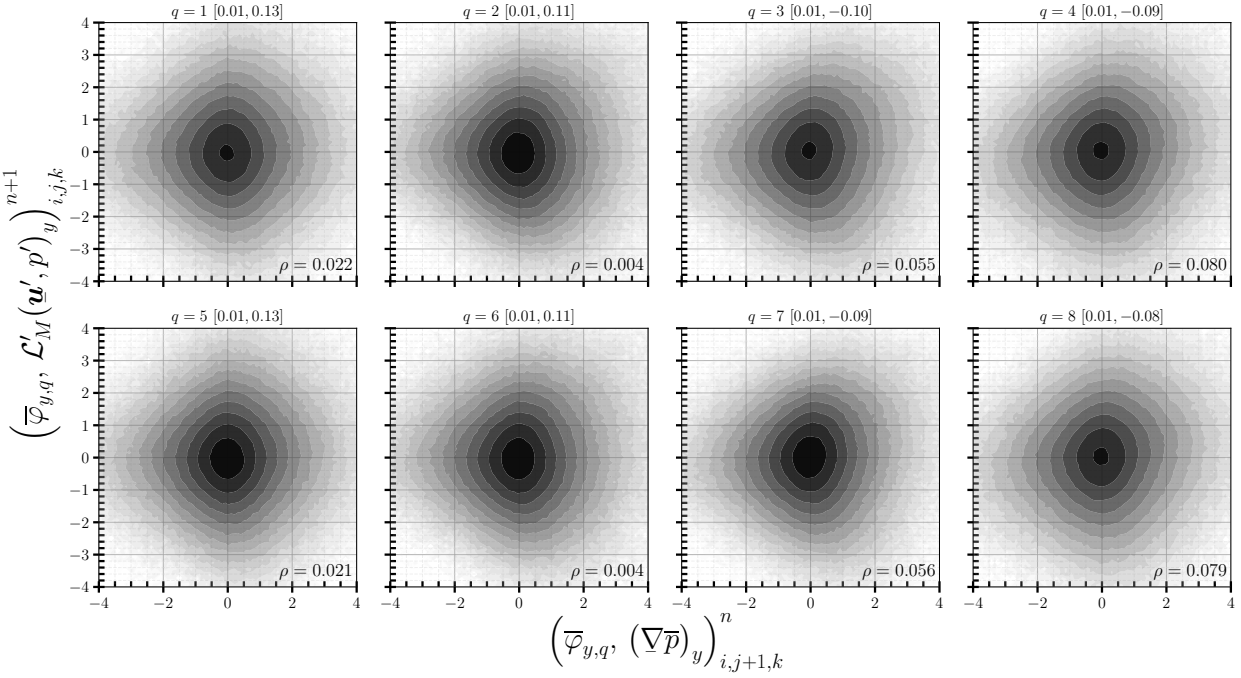


FIGURE A.180 Histograms of the t^n $(i, j+1, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

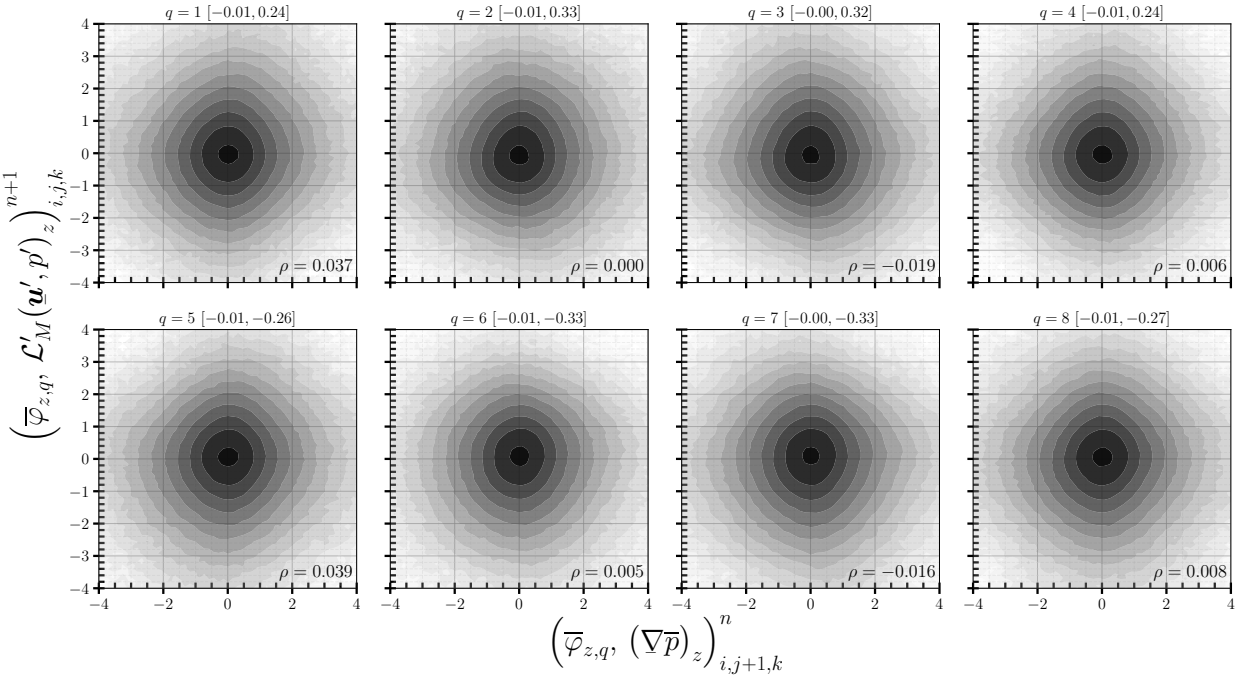


FIGURE A.181 Histograms of the t^n $(i, j + 1, k)$-element *z*-momentum pressure gradient term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

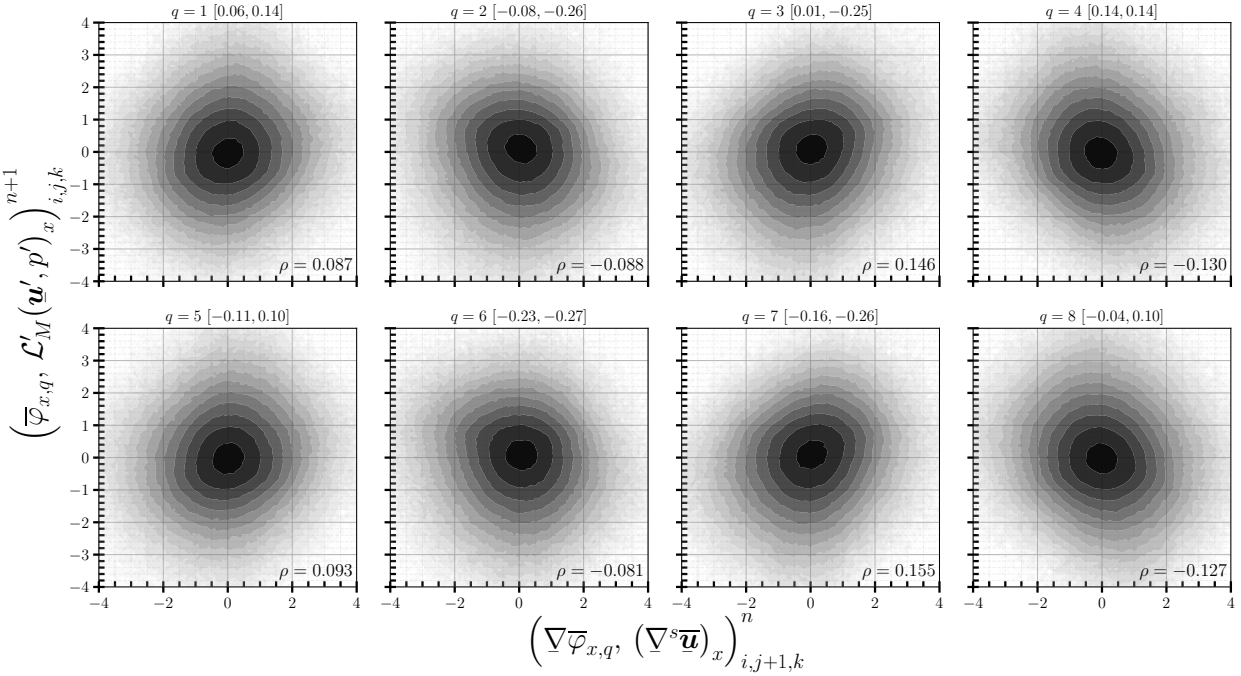


FIGURE A.182 Histograms of the t^n $(i, j + 1, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

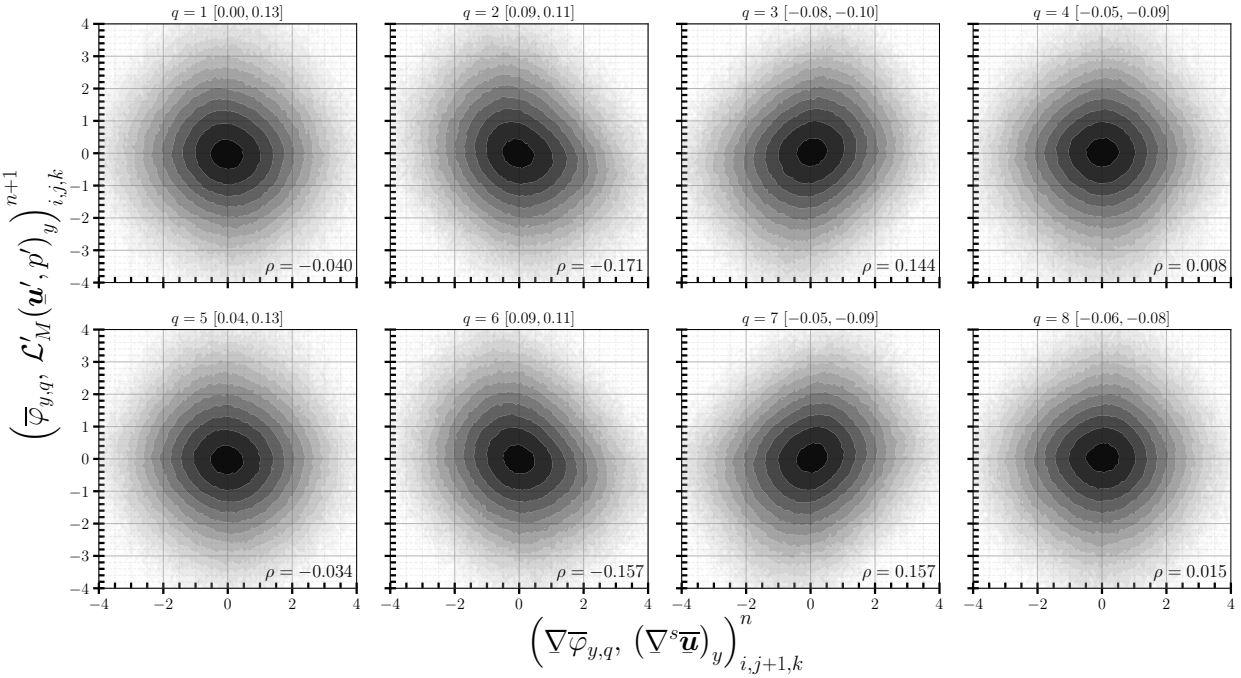


FIGURE A.183 Histograms of the t^n $(i, j+1, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

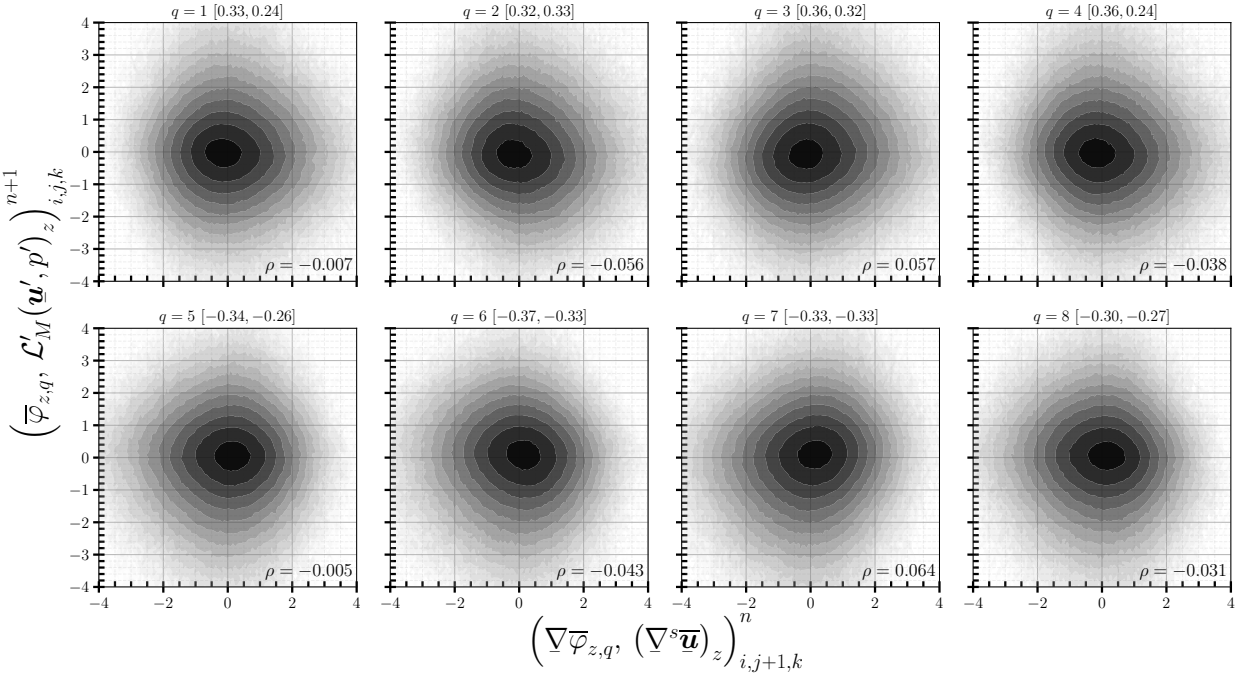


FIGURE A.184 Histograms of the t^n $(i, j+1, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.2.9 $(i, j, k \pm 1)$ Histograms

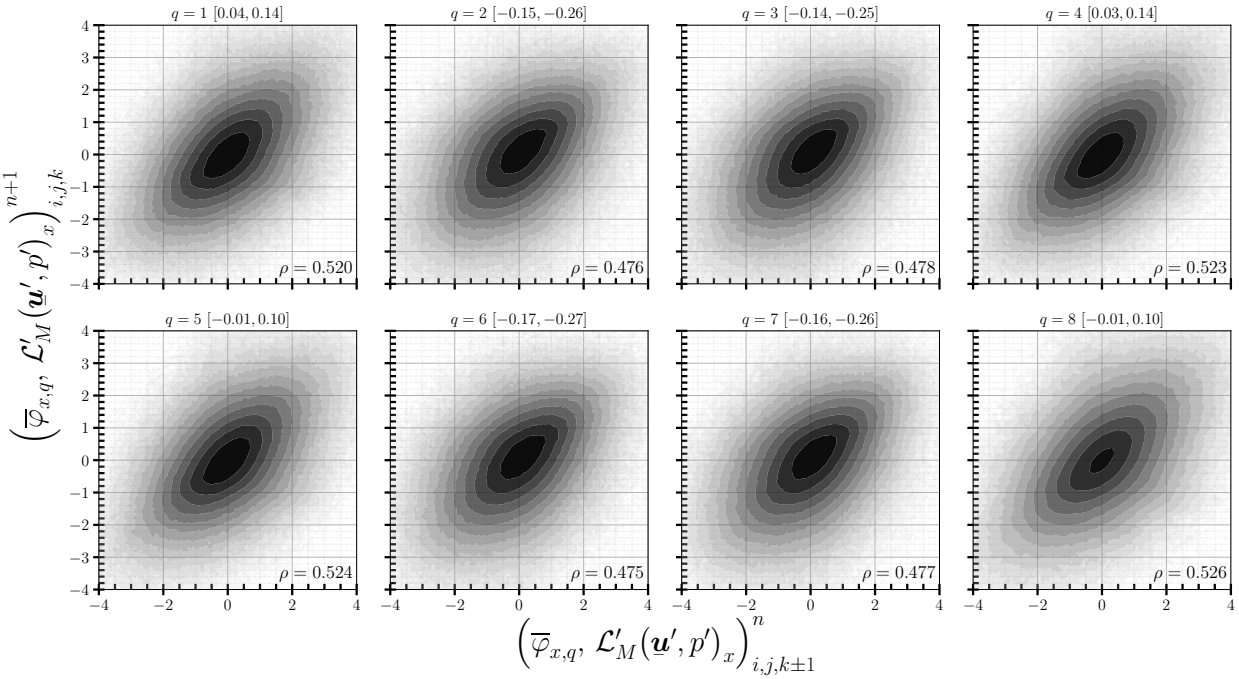


FIGURE A.185 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

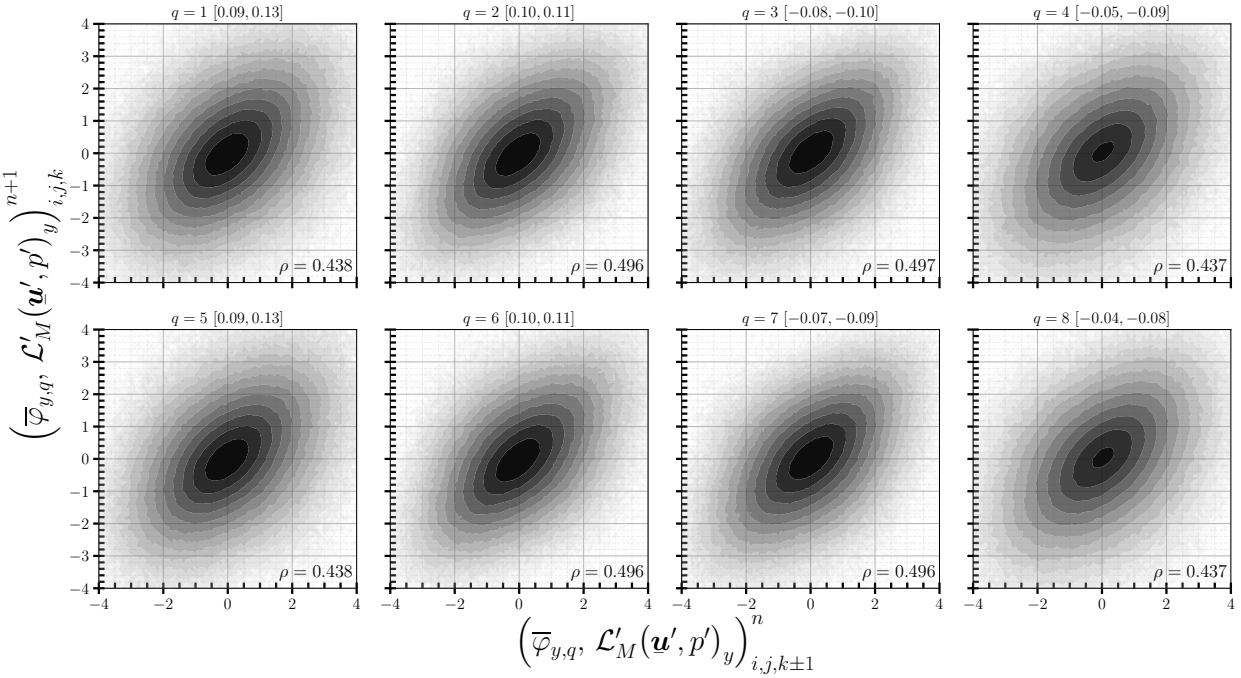


FIGURE A.186 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

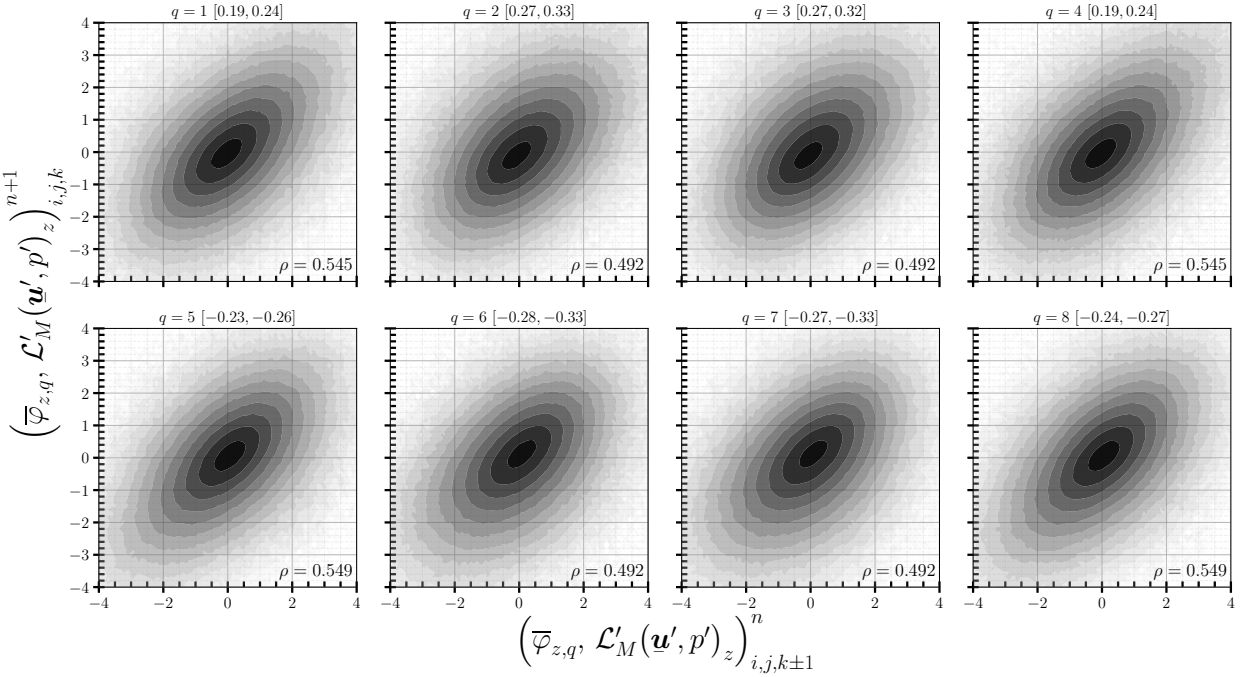


FIGURE A.187 Histograms of the t^n $(i, j, k \pm 1)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

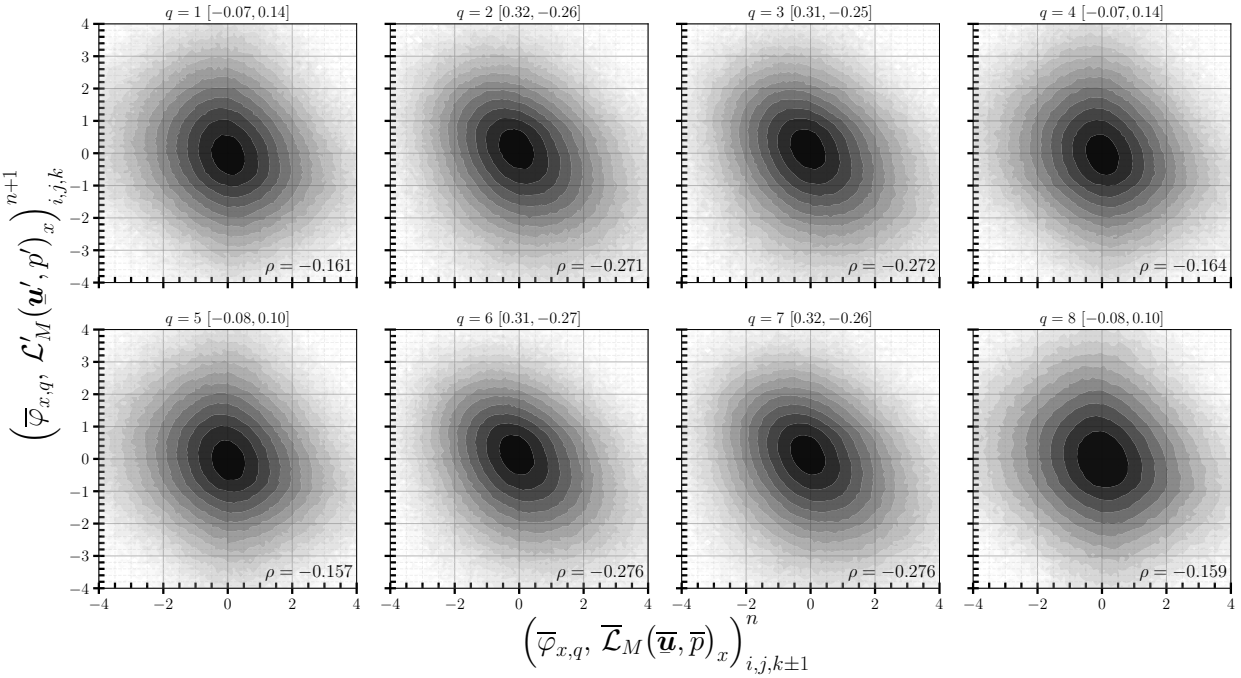


FIGURE A.188 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

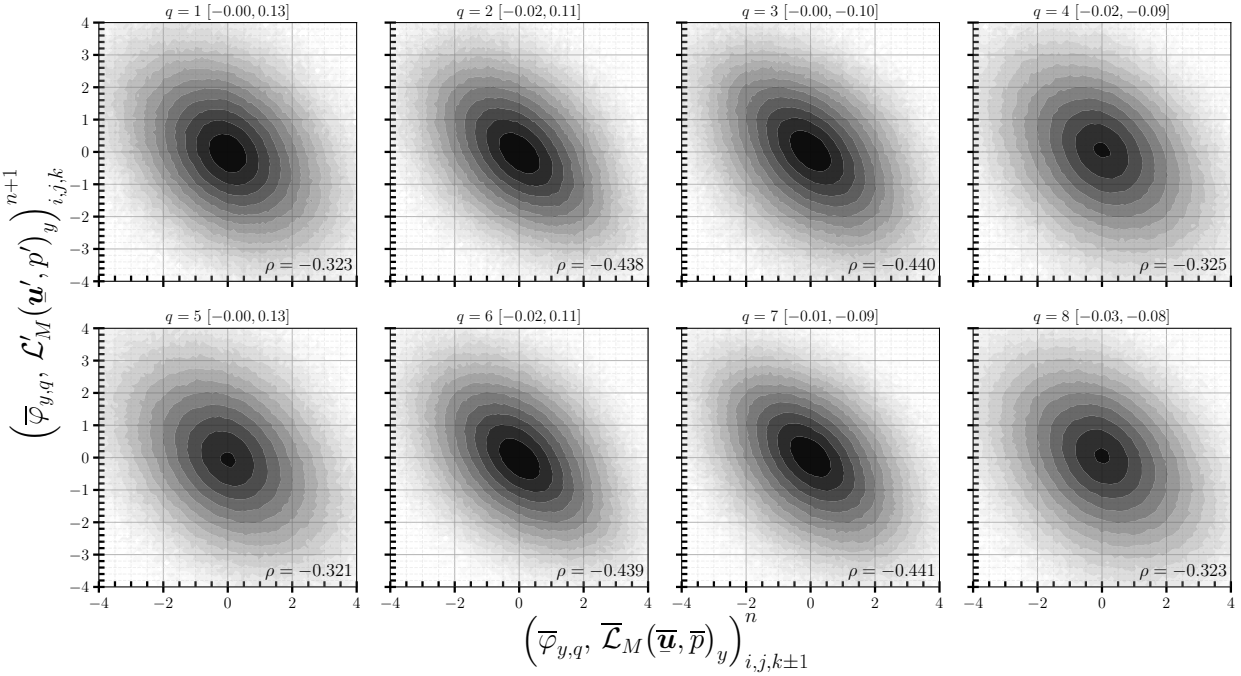


FIGURE A.189 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

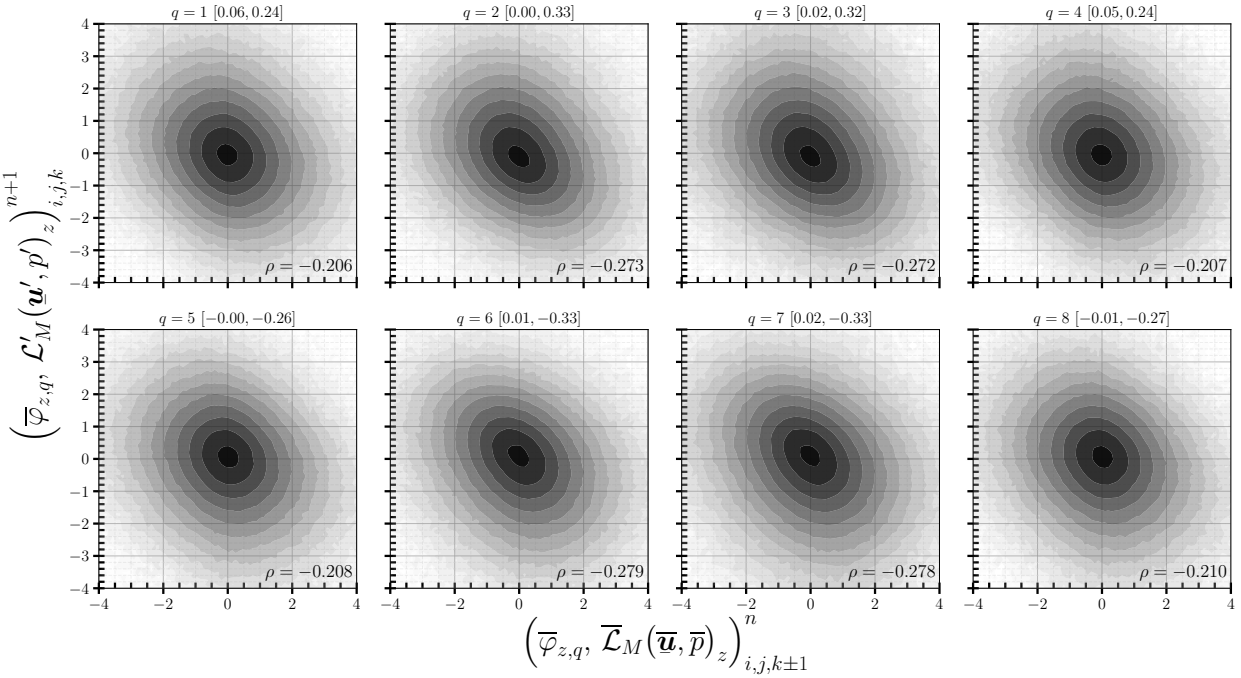


FIGURE A.190 Histograms of the t^n $(i, j, k \pm 1)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

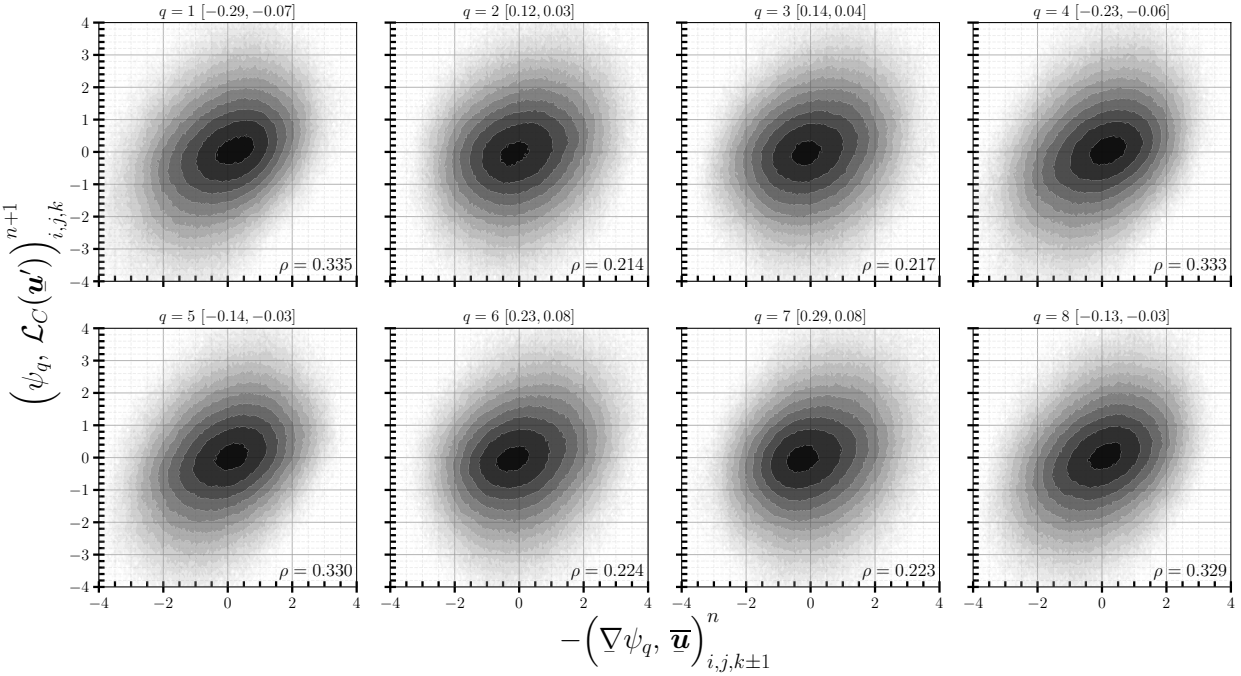


FIGURE A.191 Histograms of the t^n $(i, j, k \pm 1)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

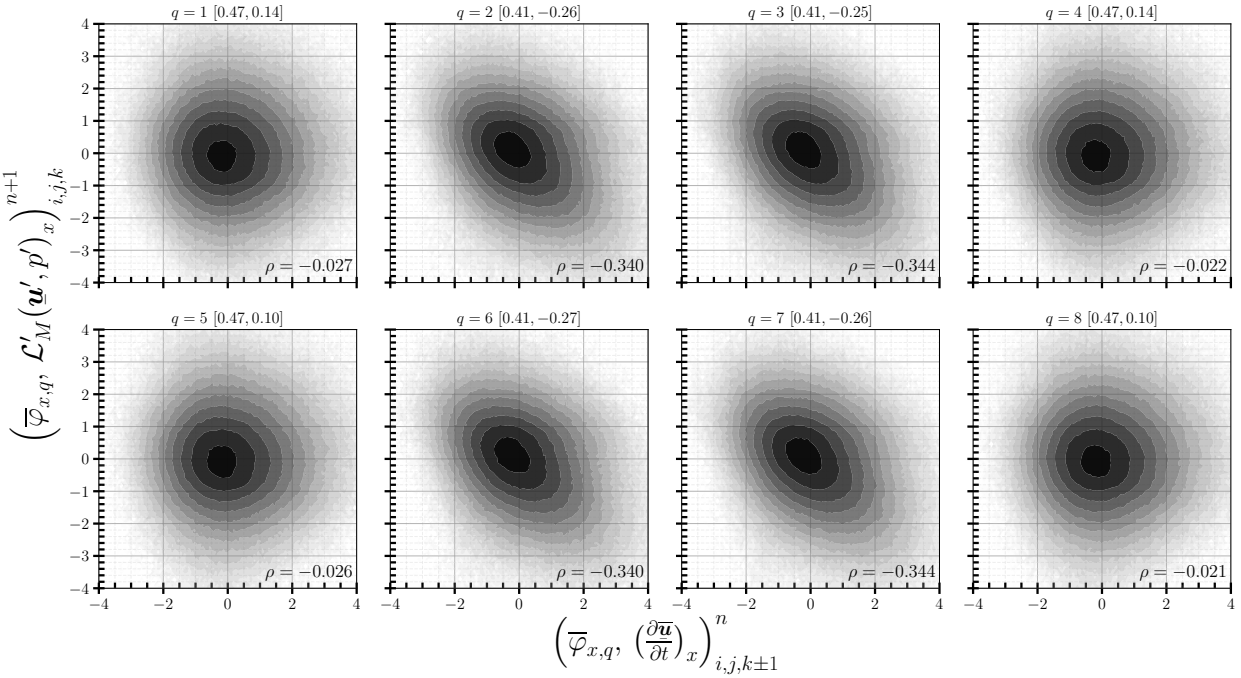


FIGURE A.192 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

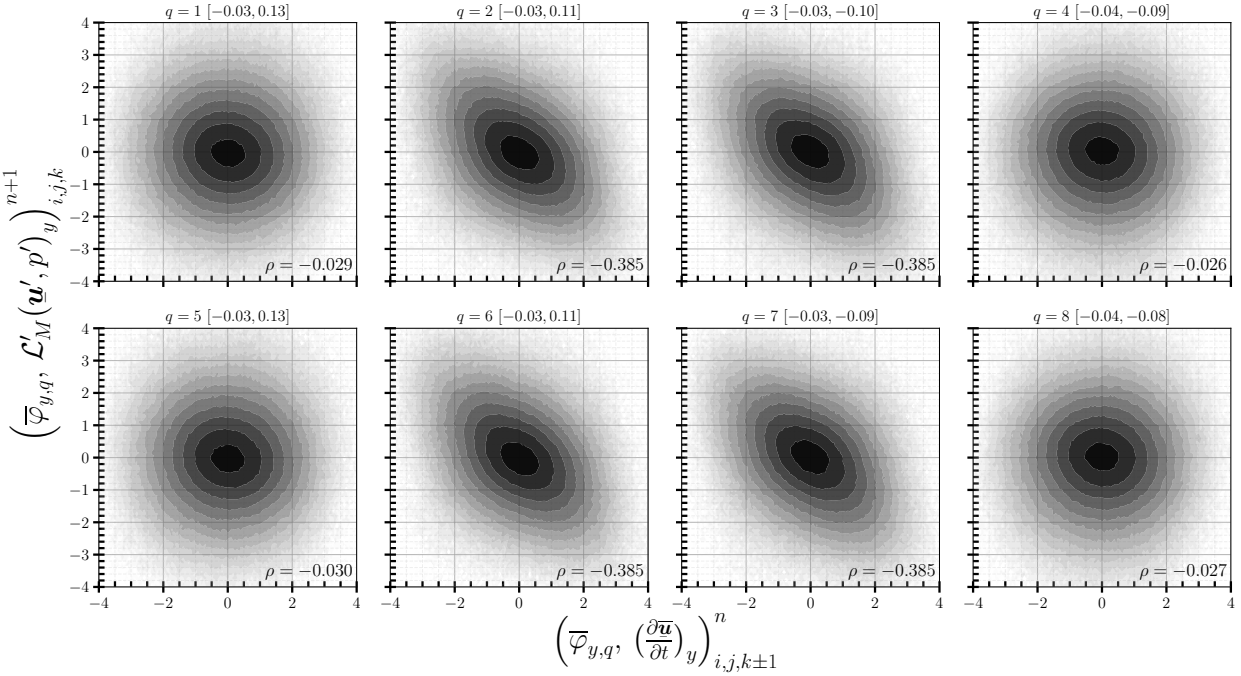


FIGURE A.193 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

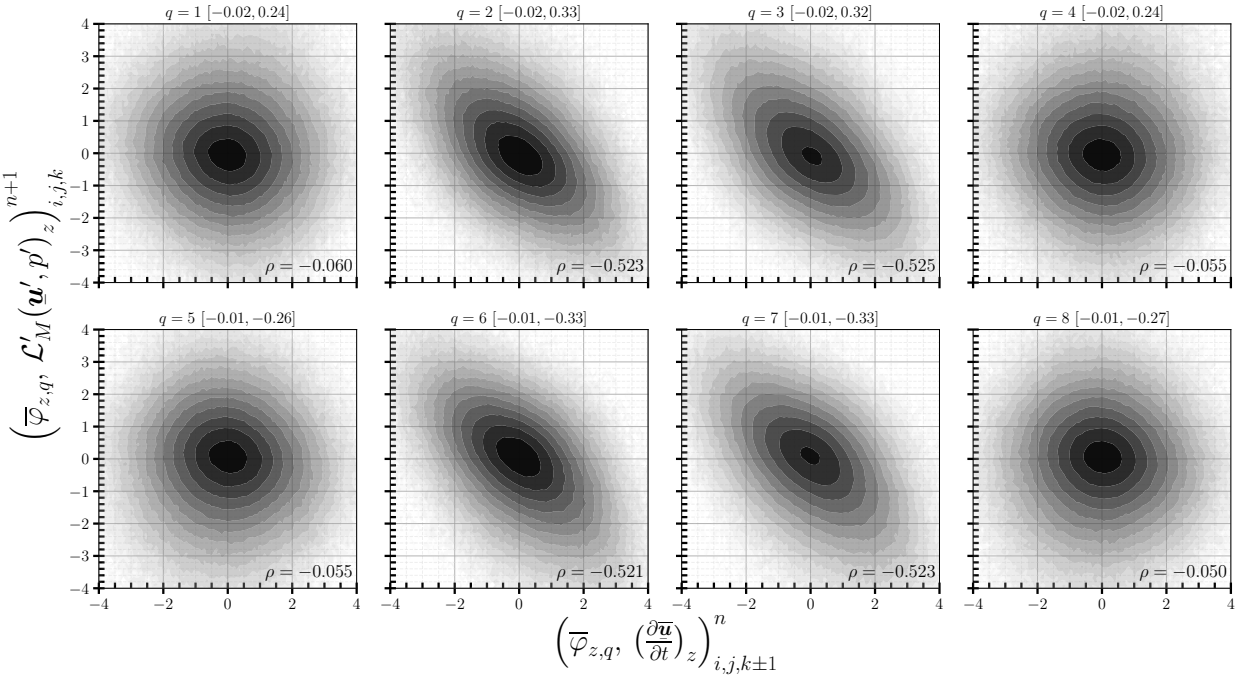


FIGURE A.194 Histograms of the t^n $(i, j, k \pm 1)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

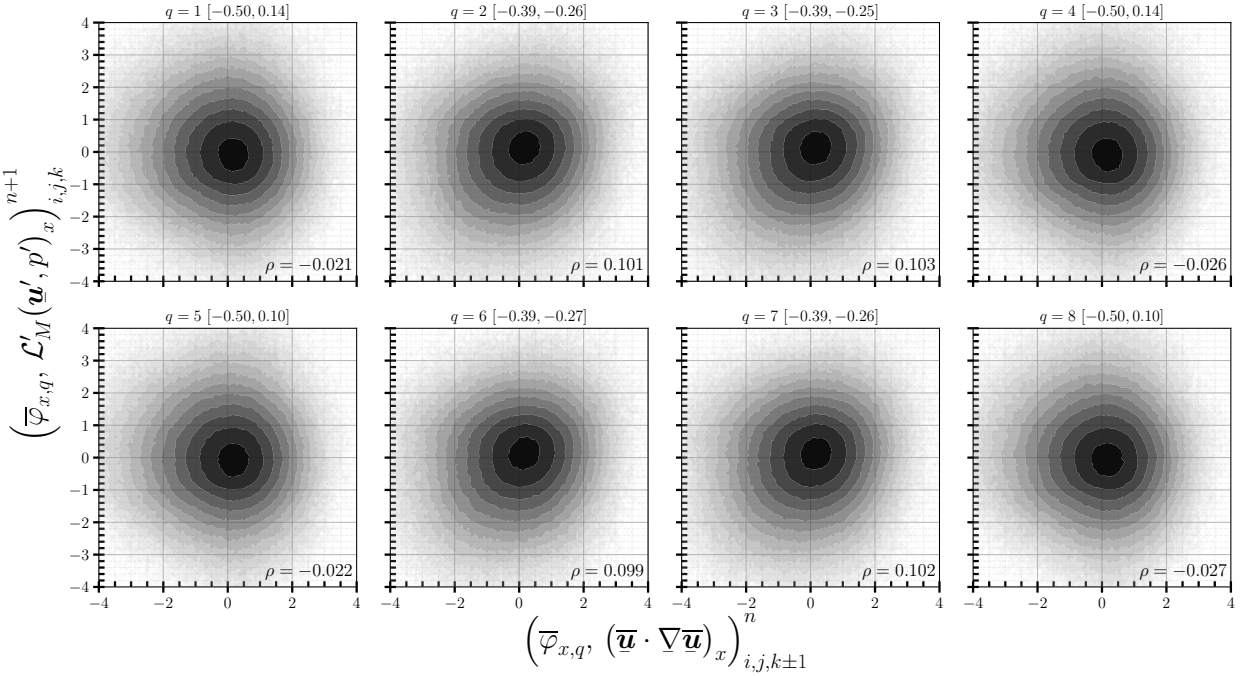


FIGURE A.195 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

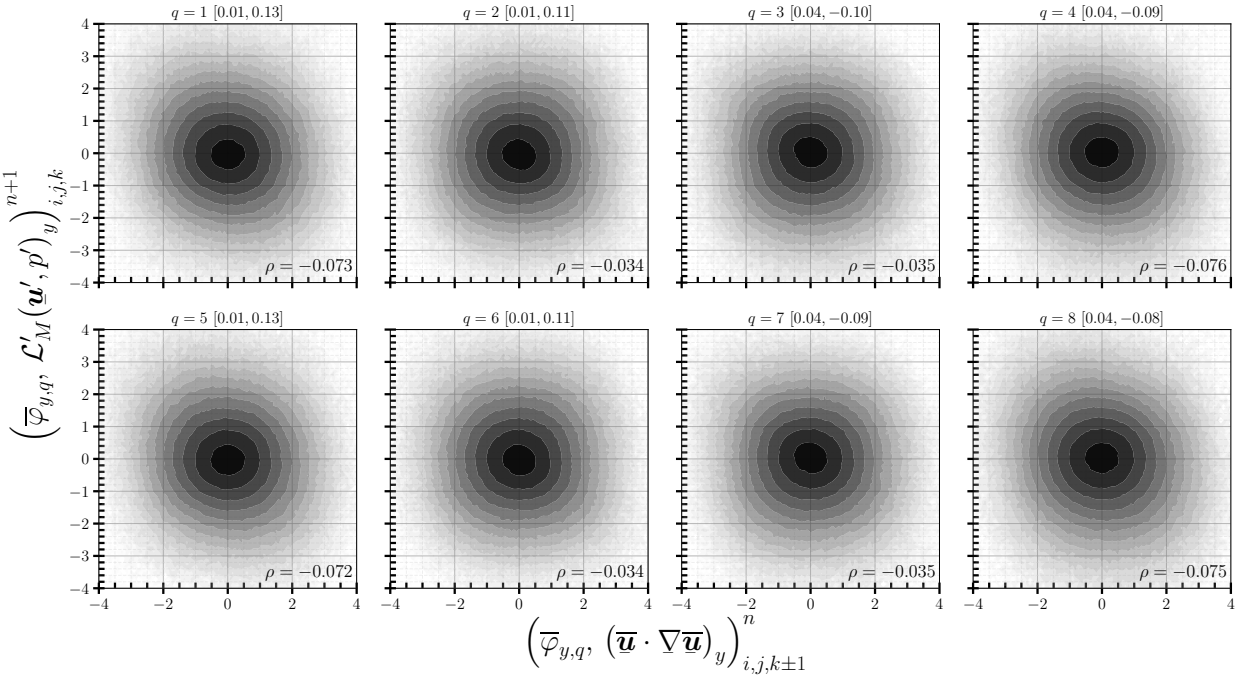


FIGURE A.196 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

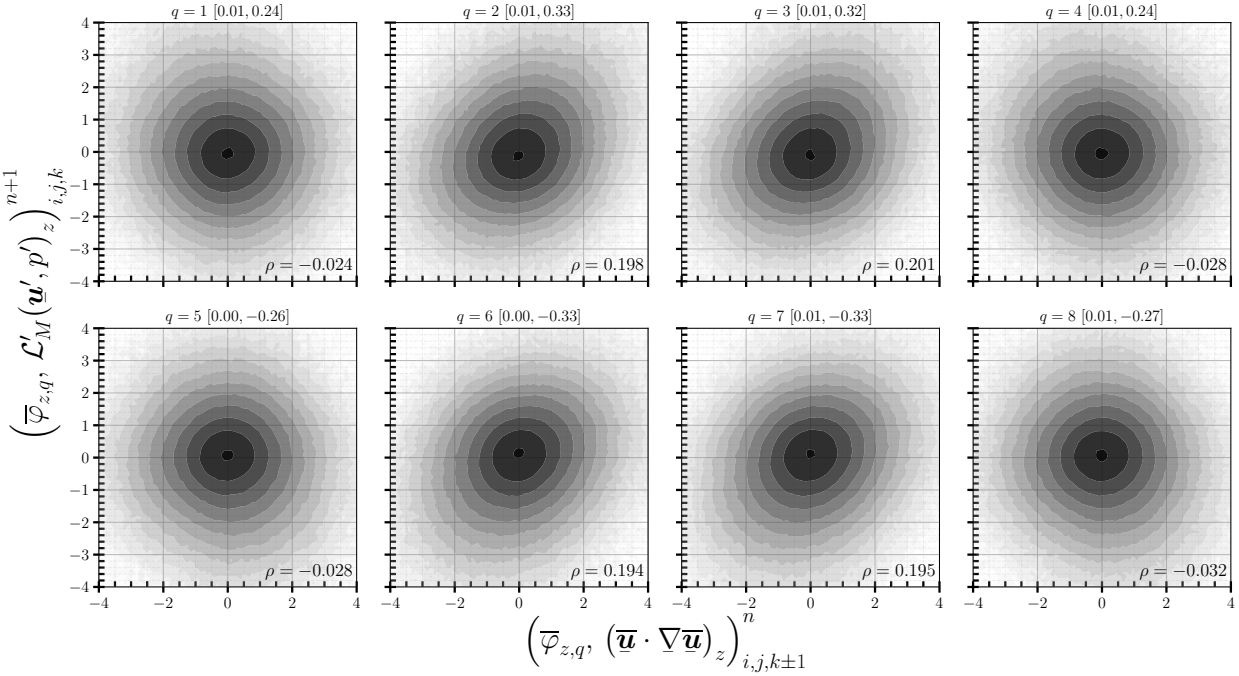


FIGURE A.197 Histograms of the t^n $(i, j, k \pm 1)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

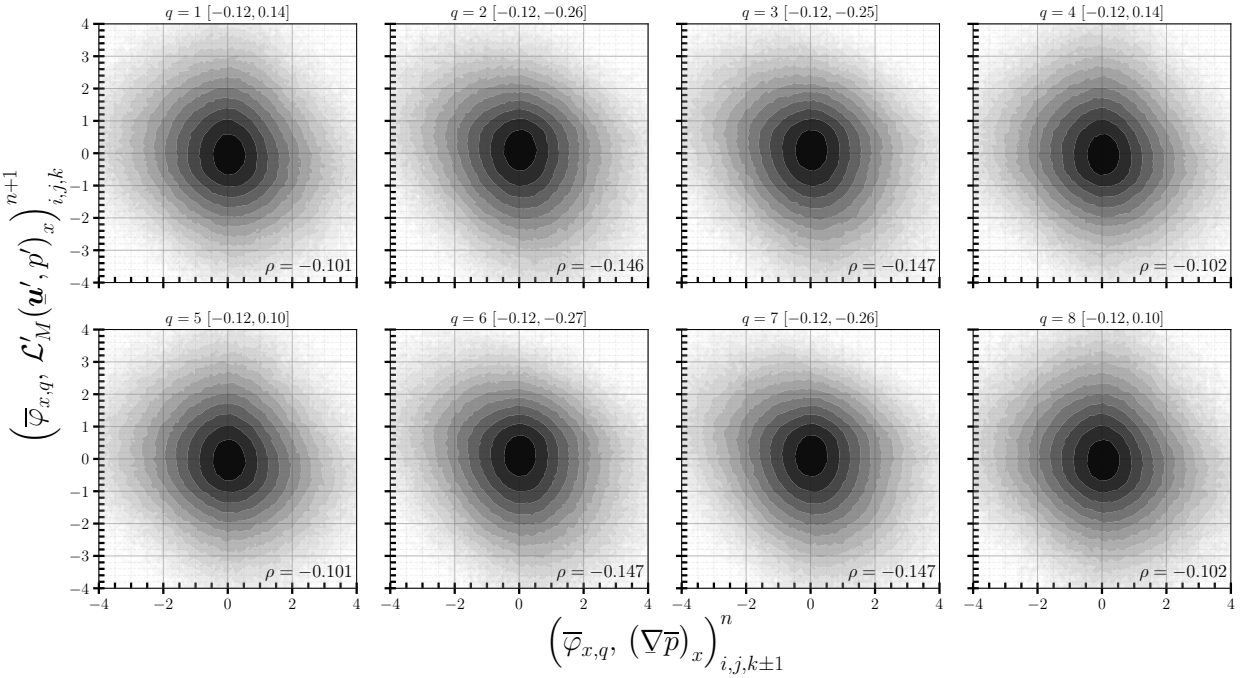


FIGURE A.198 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

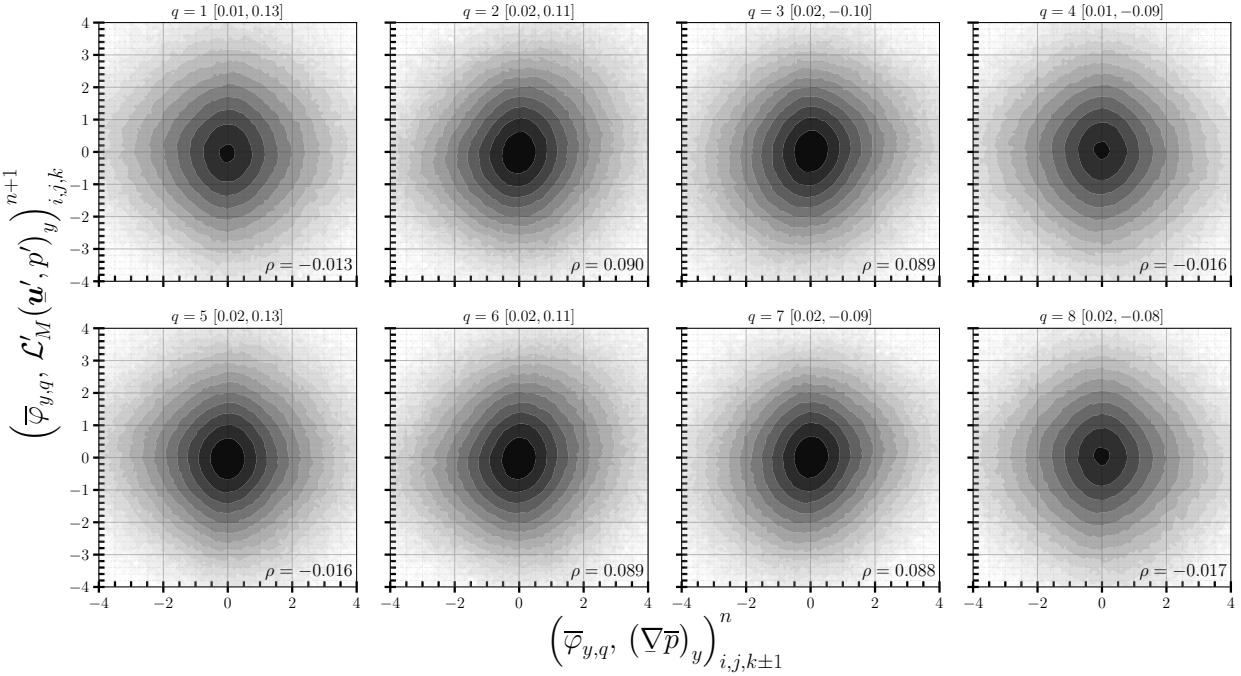


FIGURE A.199 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

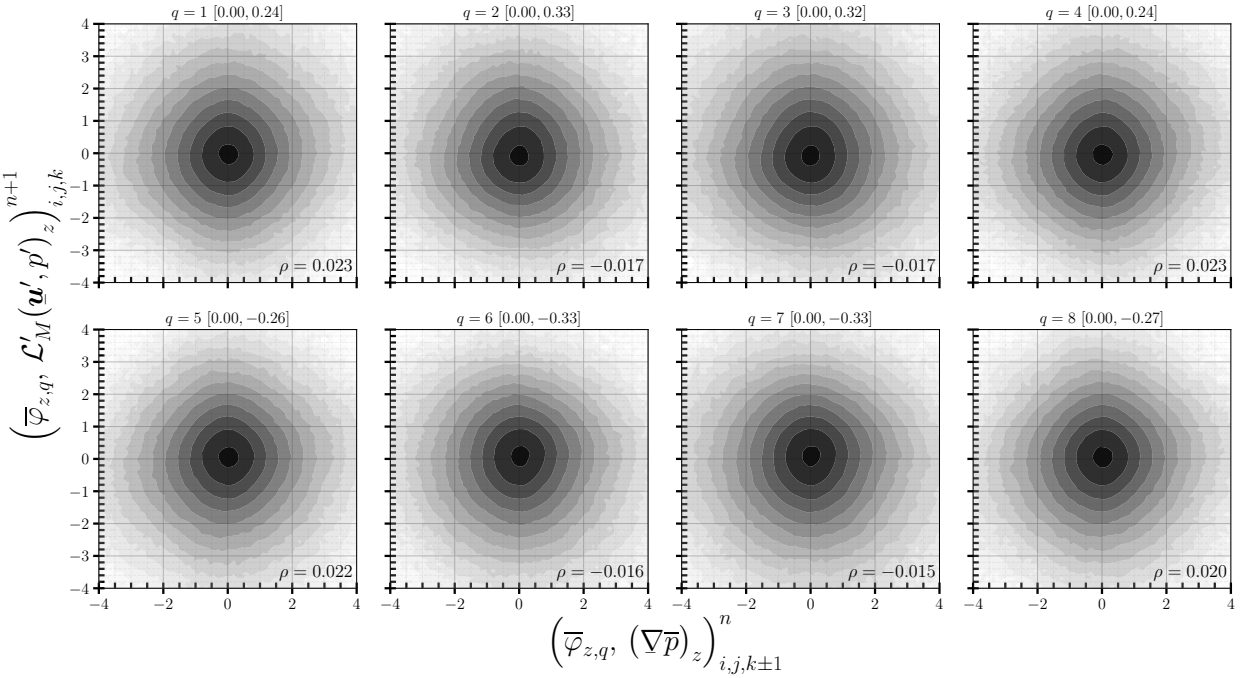


FIGURE A.200 Histograms of the t^n $(i, j, k \pm 1)$-element *z*-momentum pressure gradient term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

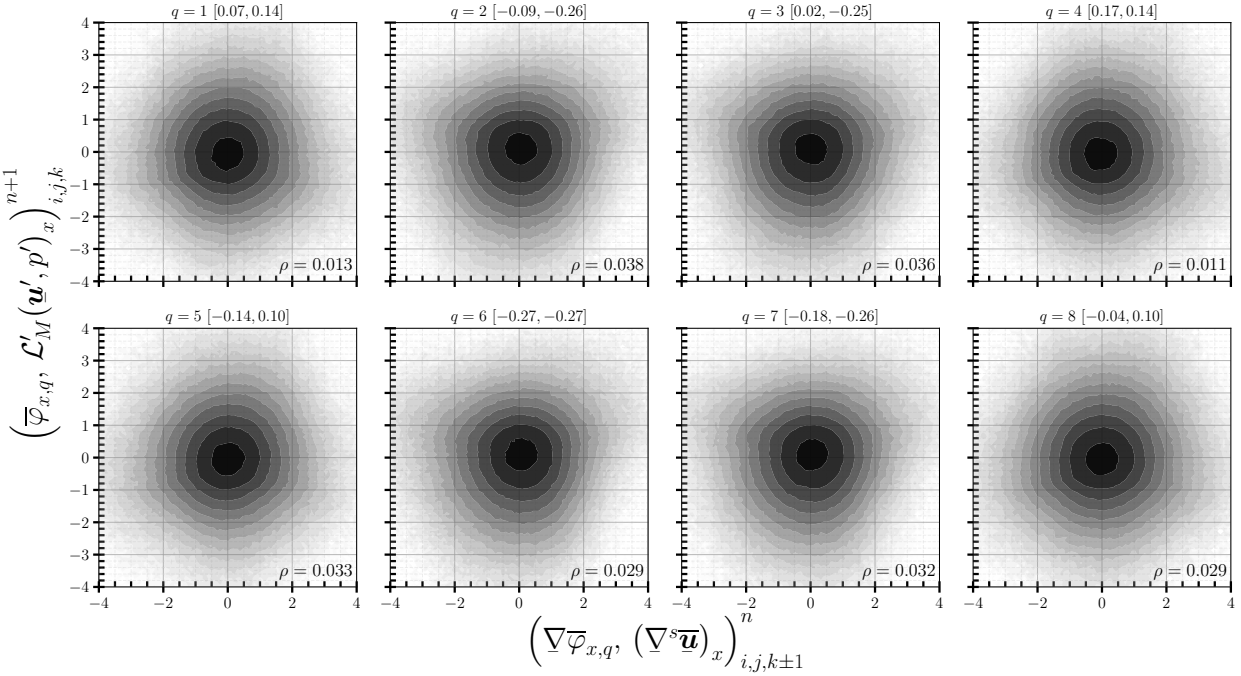


FIGURE A.201 Histograms of the t^n $(i, j, k \pm 1)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

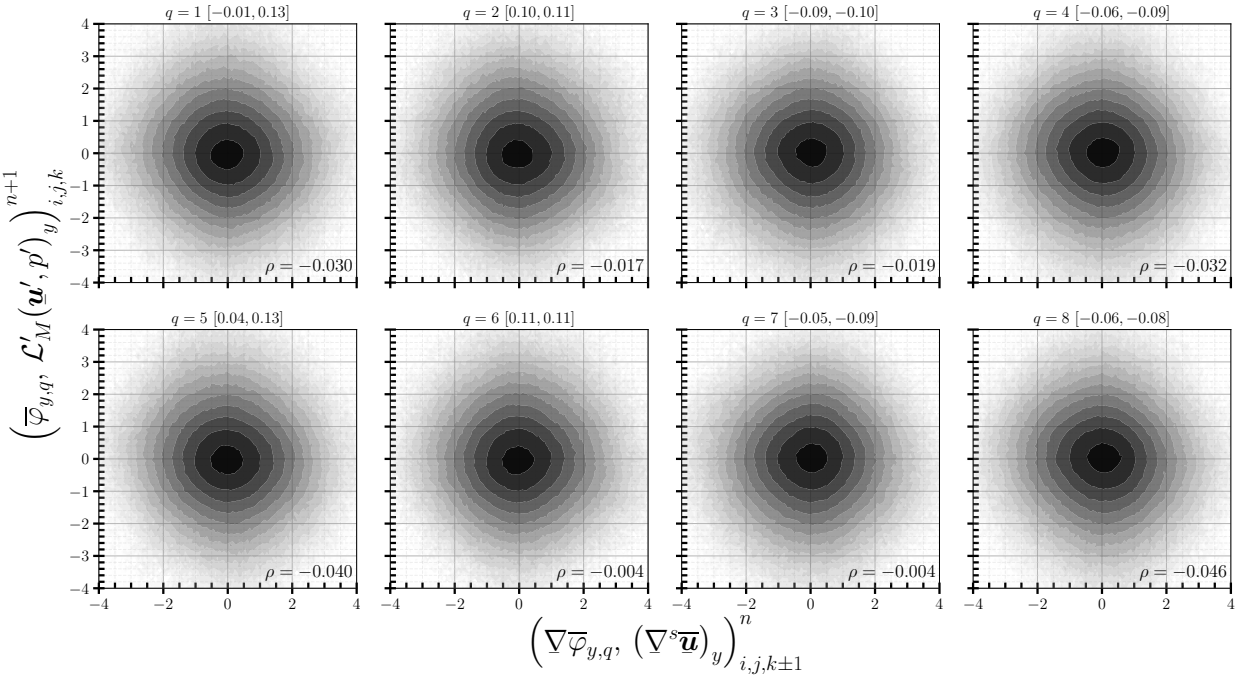


FIGURE A.202 Histograms of the t^n $(i, j, k \pm 1)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

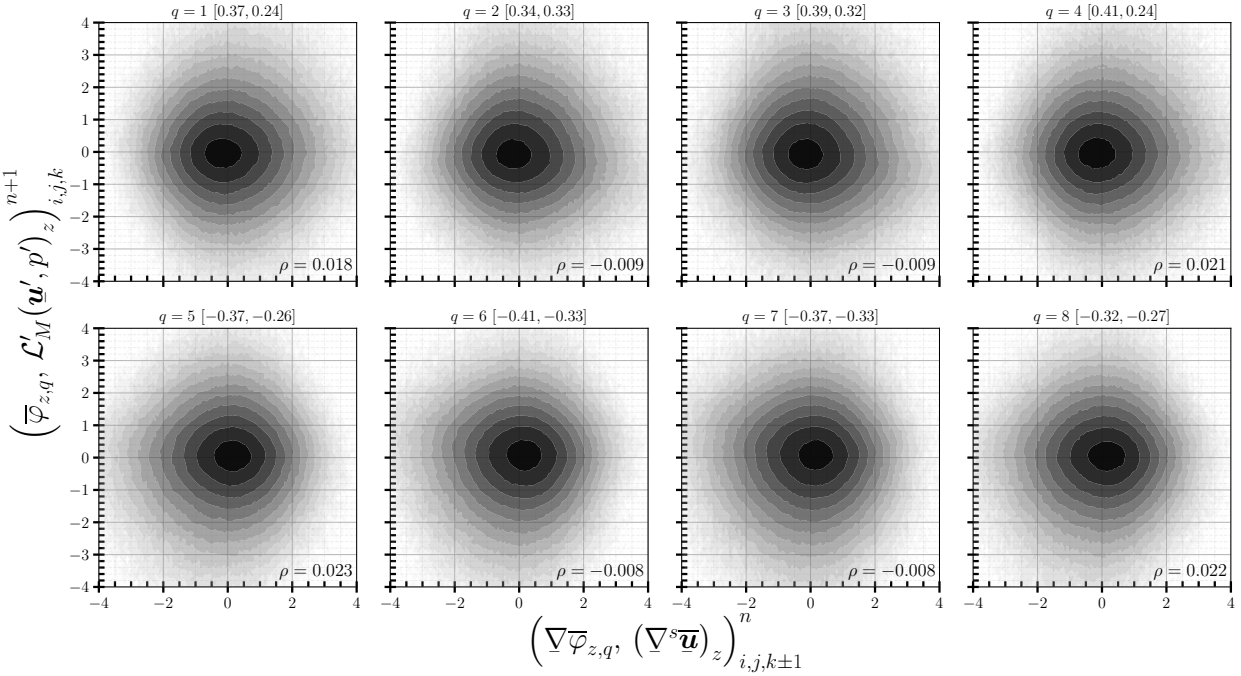


FIGURE A.203 Histograms of the t^n $(i, j, k \pm 1)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3 t^{n-1} - t^{n+1} Histograms

A.3.1 (i, j, k) Histograms

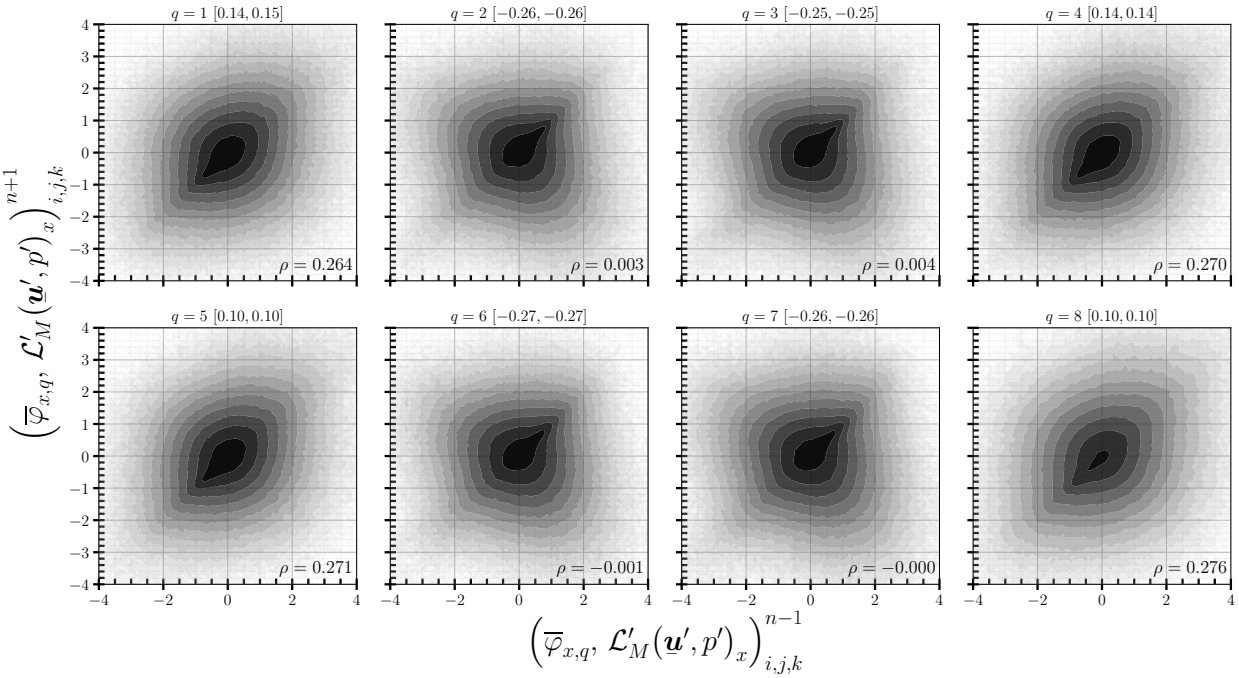


FIGURE A.204 Histograms of the t^{n-1} (i, j, k)-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

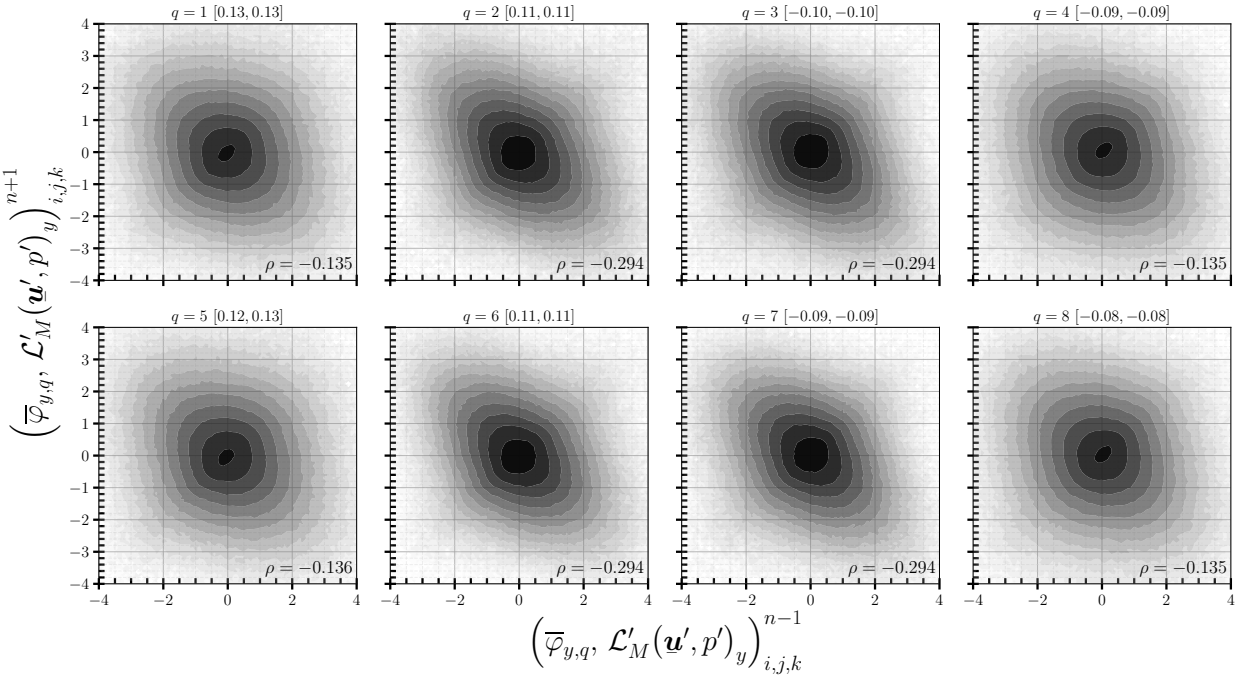


FIGURE A.205 Histograms of the t^{n-1} (i, j, k)-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

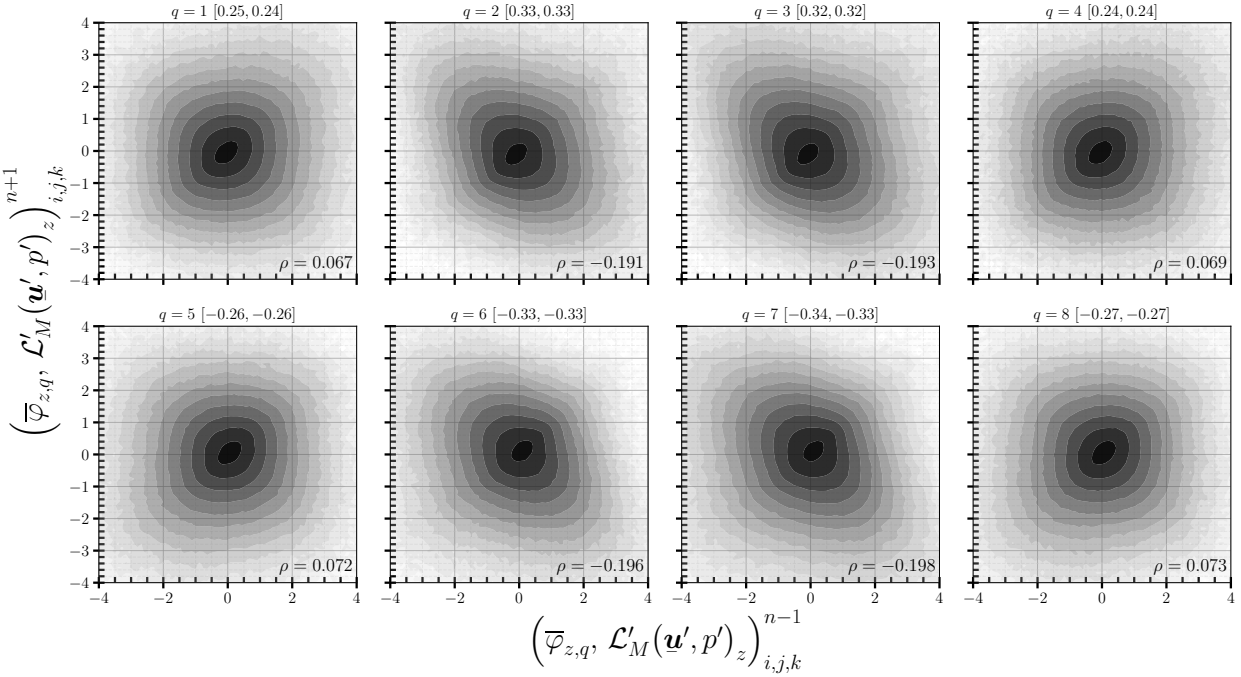


FIGURE A.206 Histograms of the t^{n-1} (i, j, k)-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

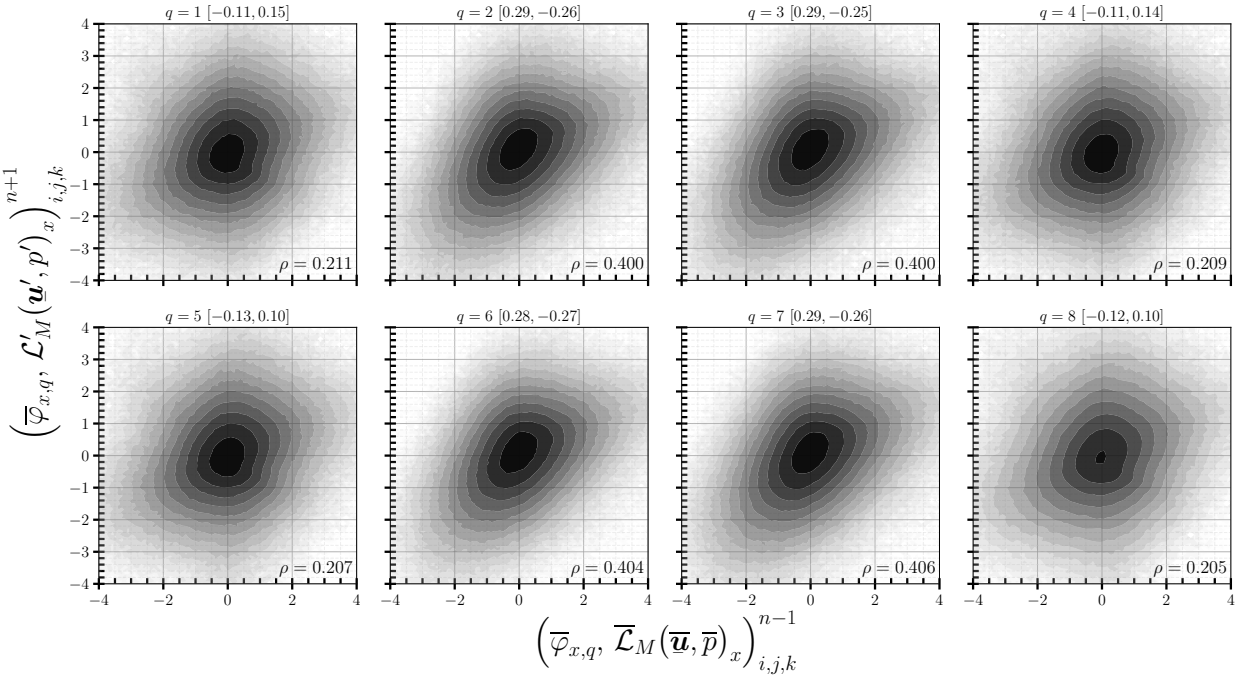


FIGURE A.207 Histograms of the t^{n-1} (i, j, k)-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

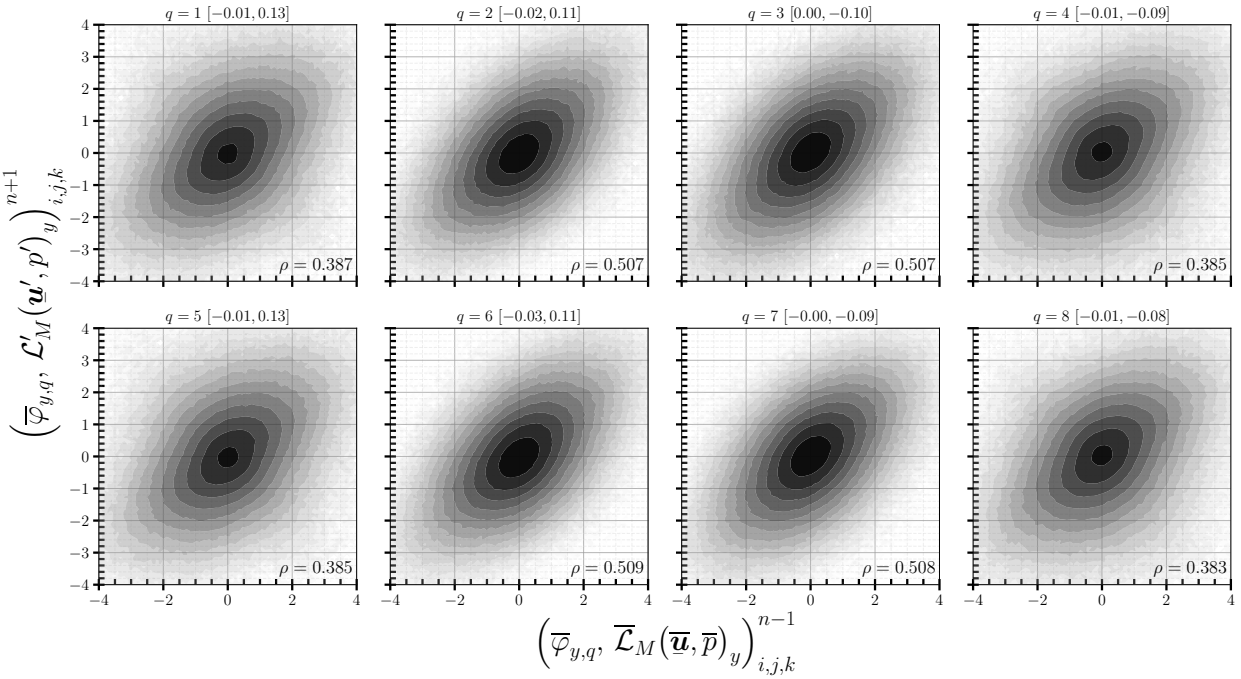


FIGURE A.208 Histograms of the t^{n-1} (i, j, k)-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

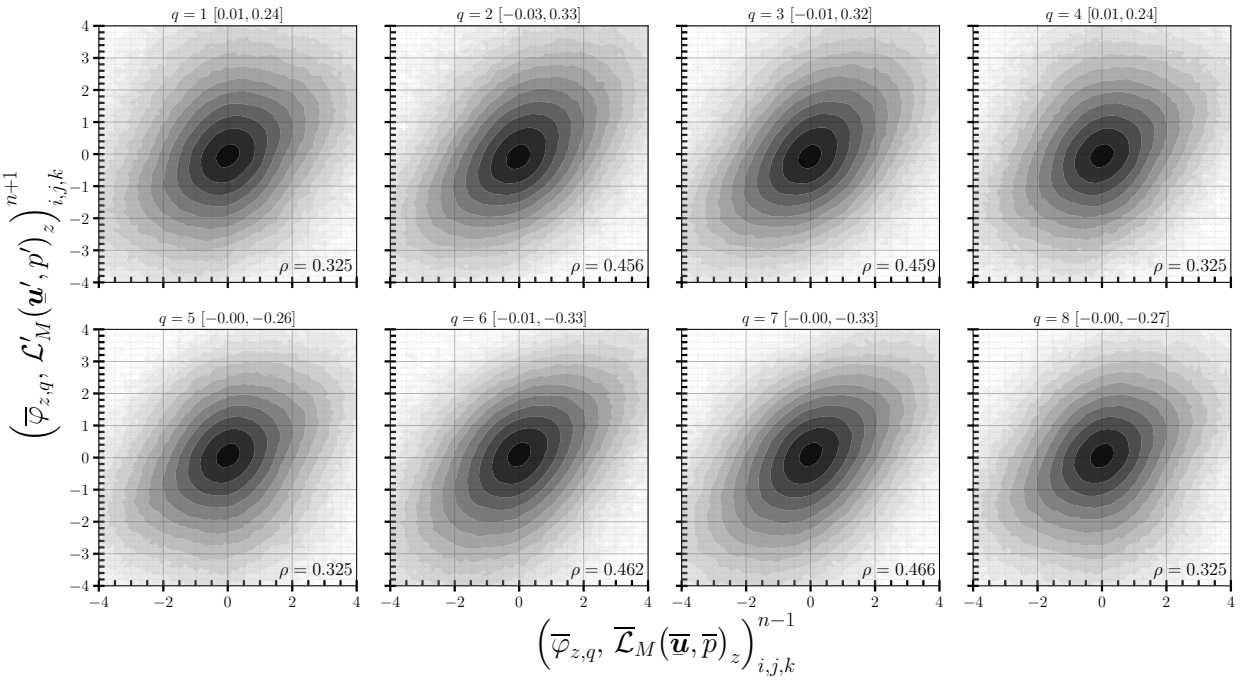


FIGURE A.209 Histograms of the t^{n-1} (i, j, k)-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

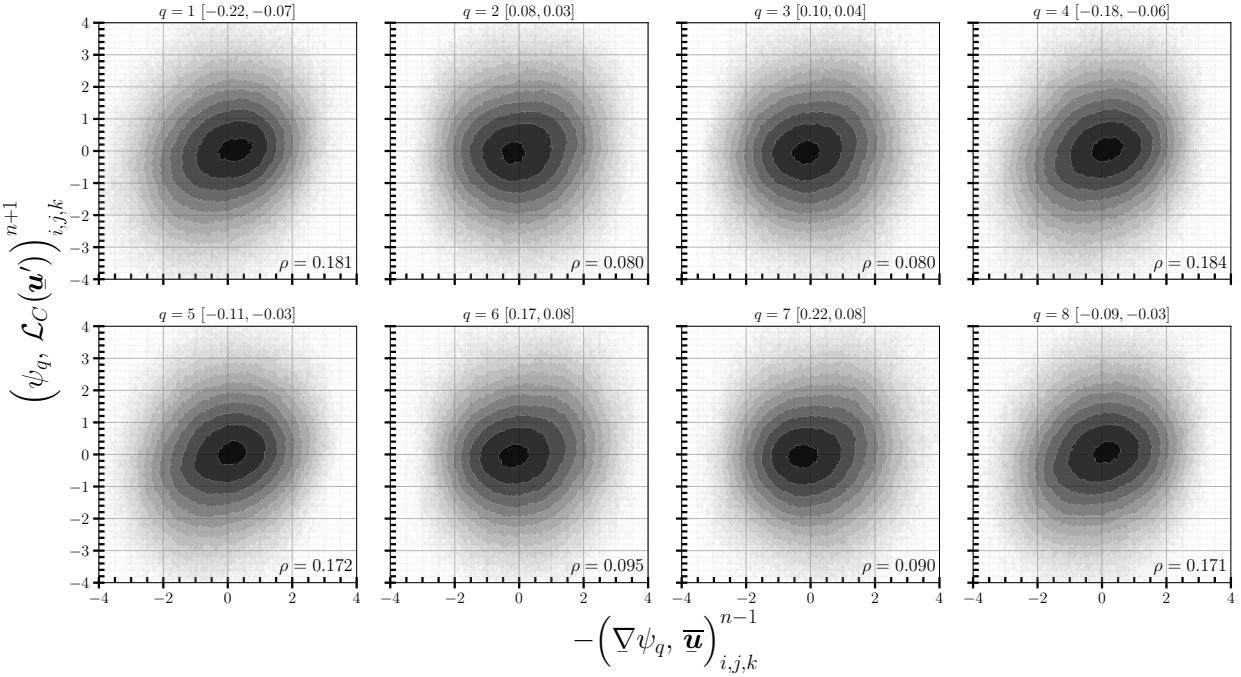


FIGURE A.210 Histograms of the t^{n-1} (i, j, k)-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

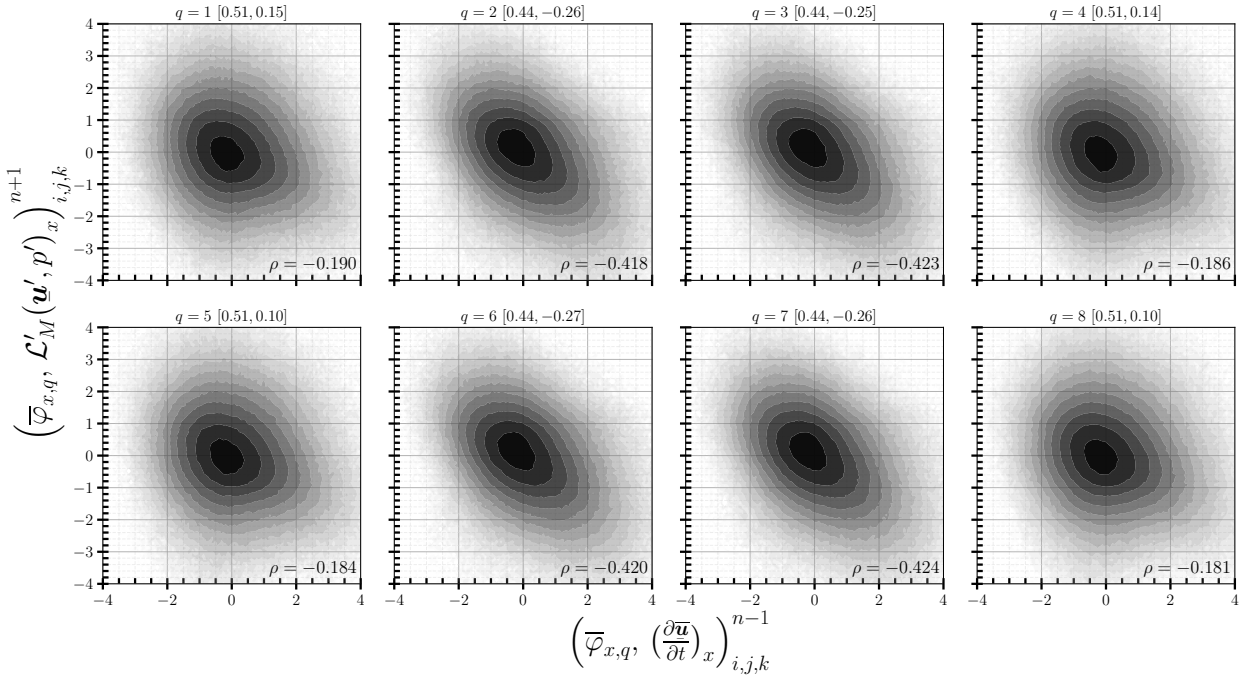


FIGURE A.211 Histograms of the t^{n-1} (i, j, k)-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

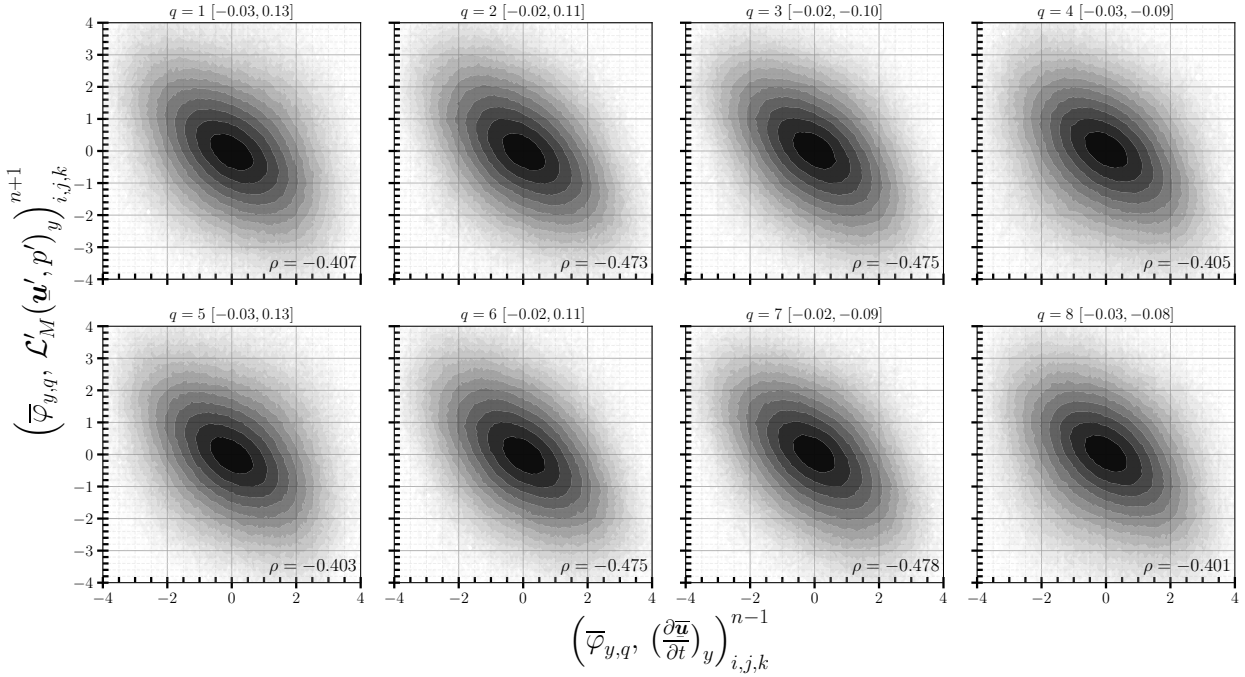


FIGURE A.212 Histograms of the t^{n-1} (i, j, k)-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

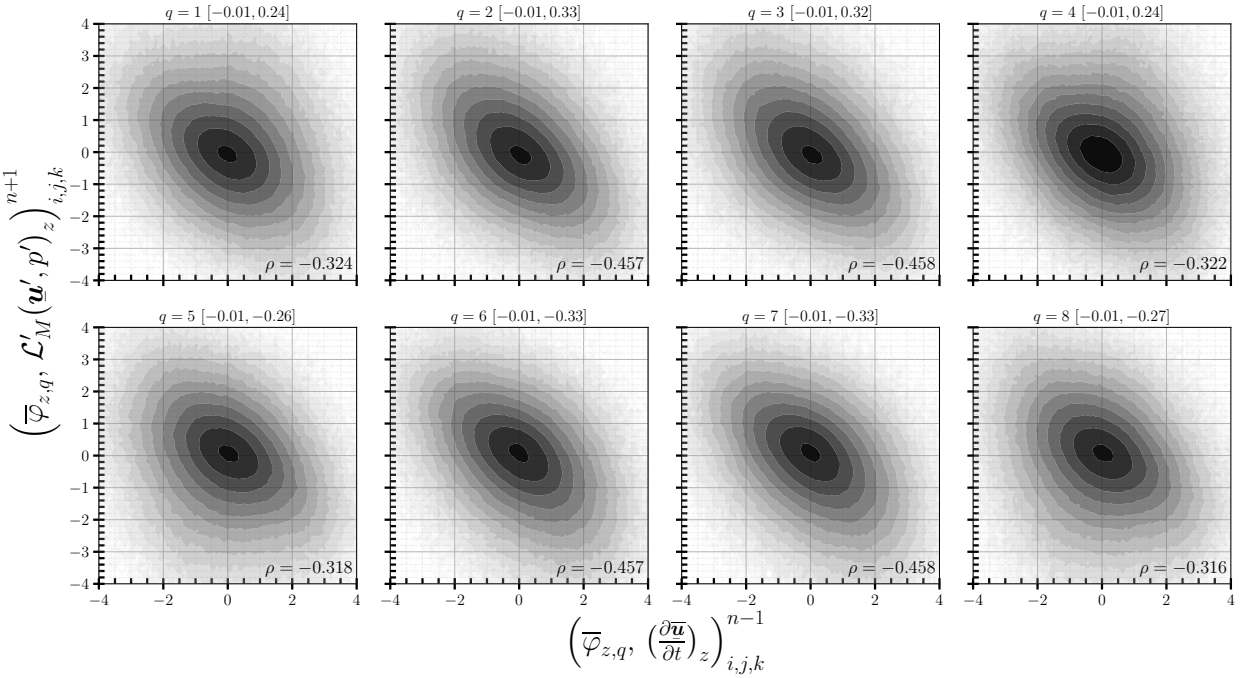


FIGURE A.213 Histograms of the t^{n-1} (i, j, k)-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

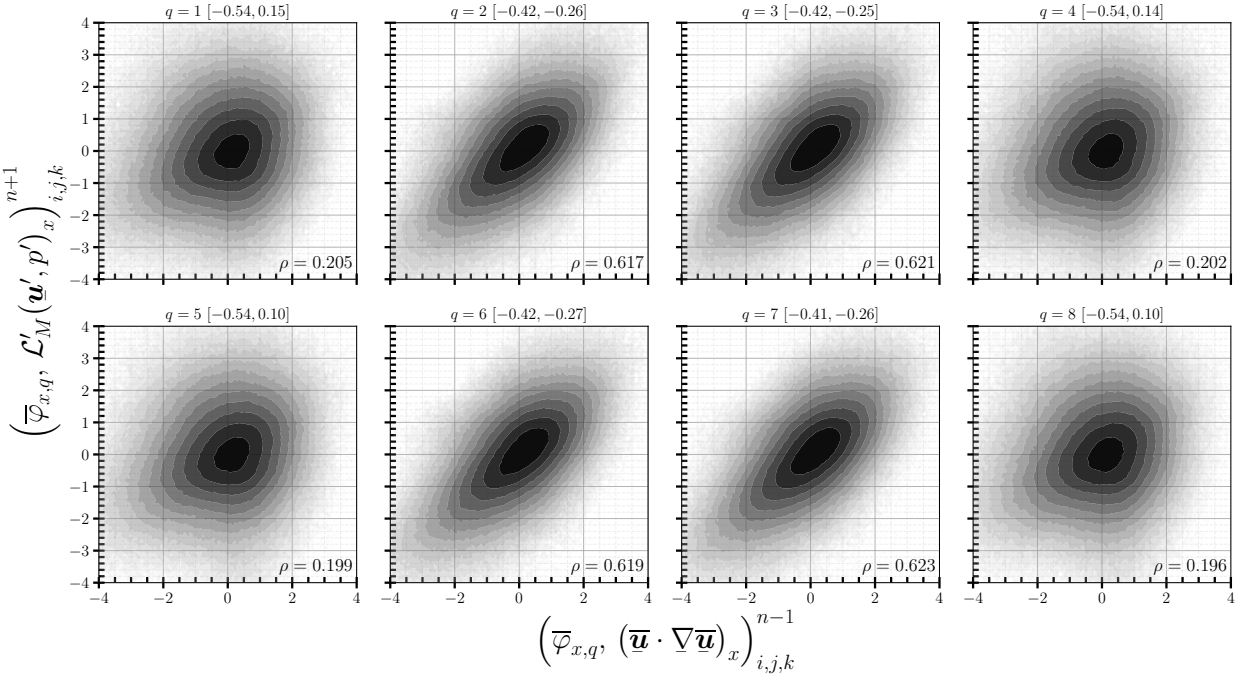


FIGURE A.214 Histograms of the t^{n-1} (i, j, k)-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

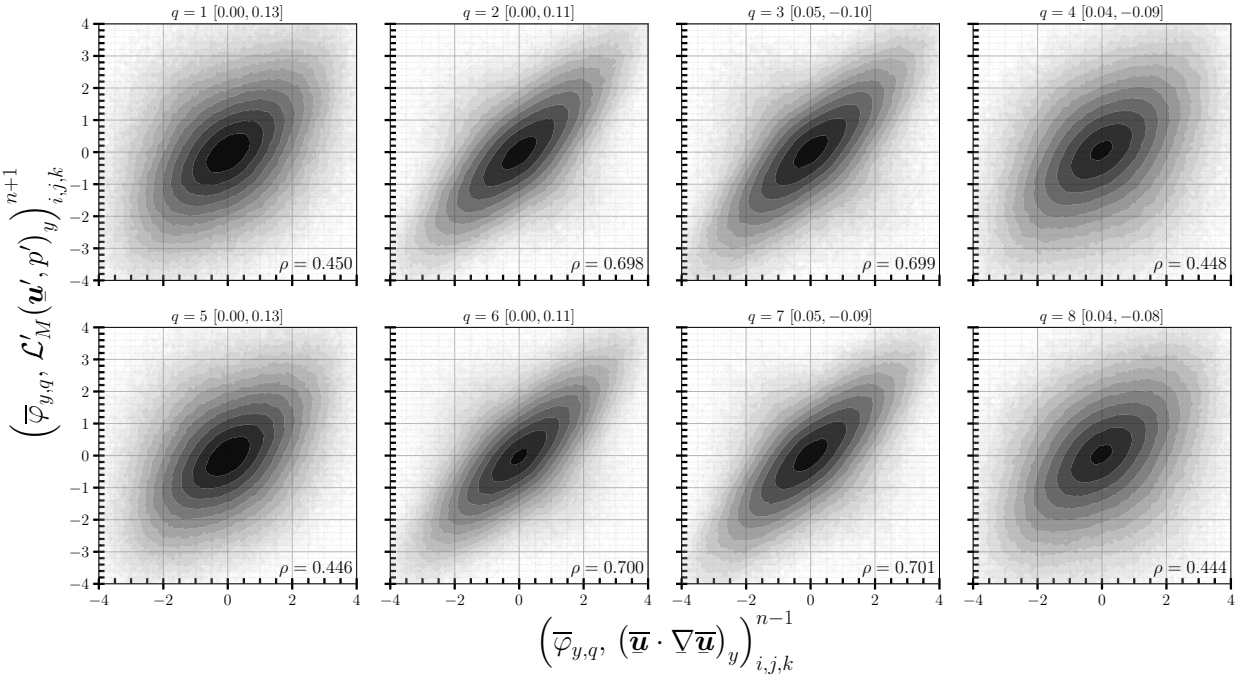


FIGURE A.215 Histograms of the t^{n-1} (i, j, k)-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

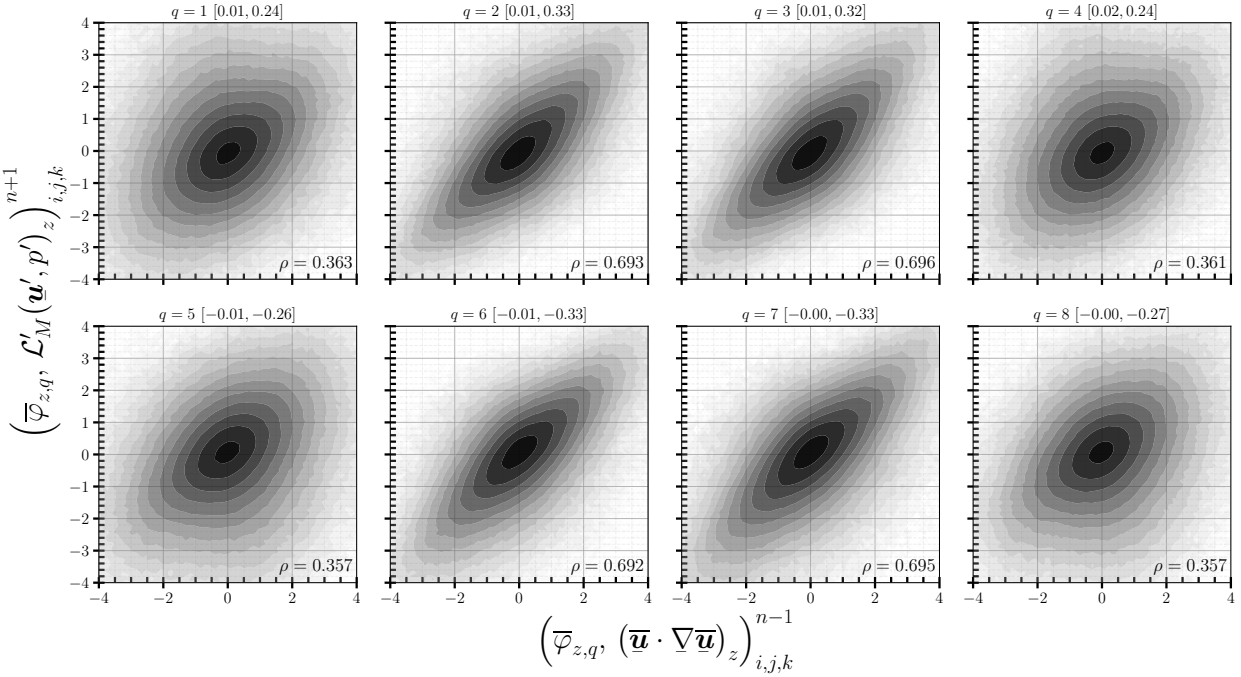


FIGURE A.216 Histograms of the t^{n-1} (i, j, k)-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

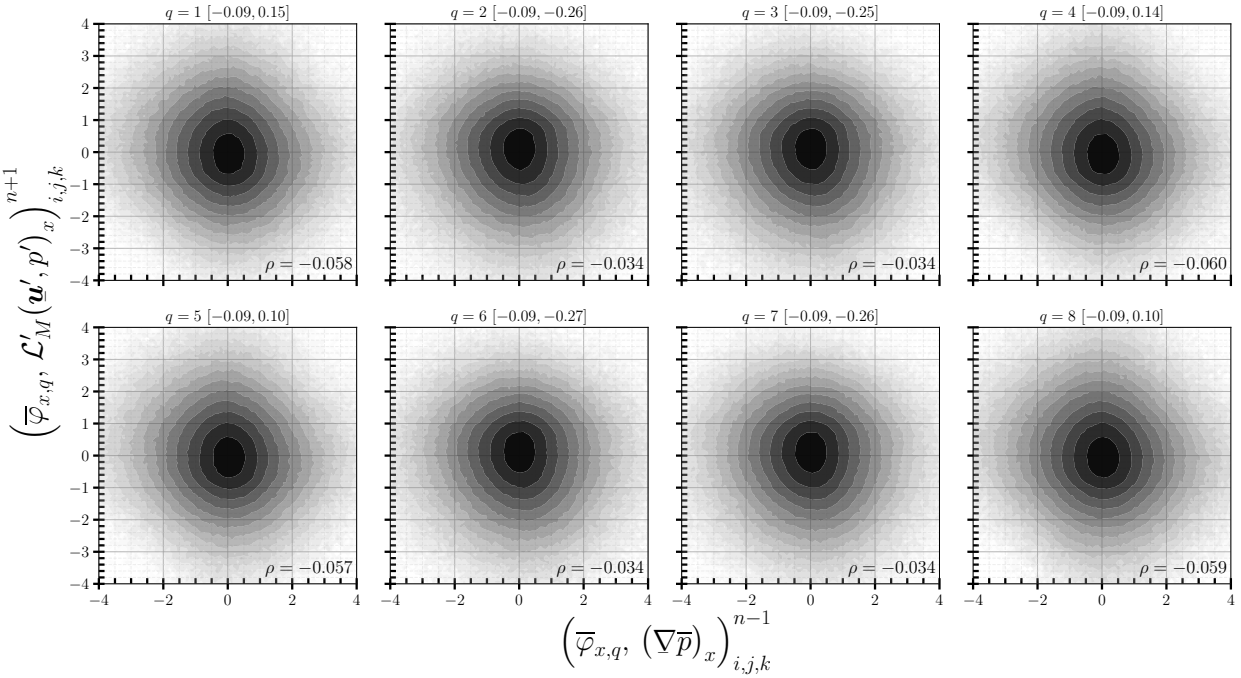


FIGURE A.217 Histograms of the t^{n-1} (i, j, k)-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

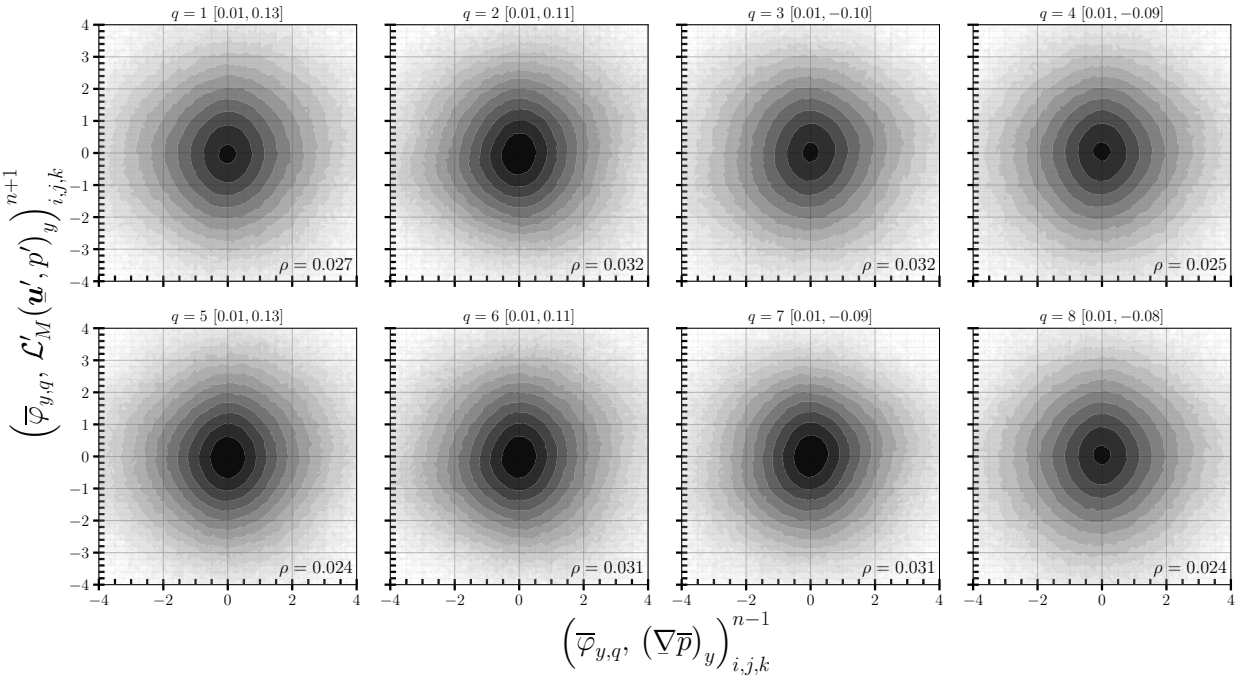


FIGURE A.218 Histograms of the t^{n-1} (i, j, k)-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

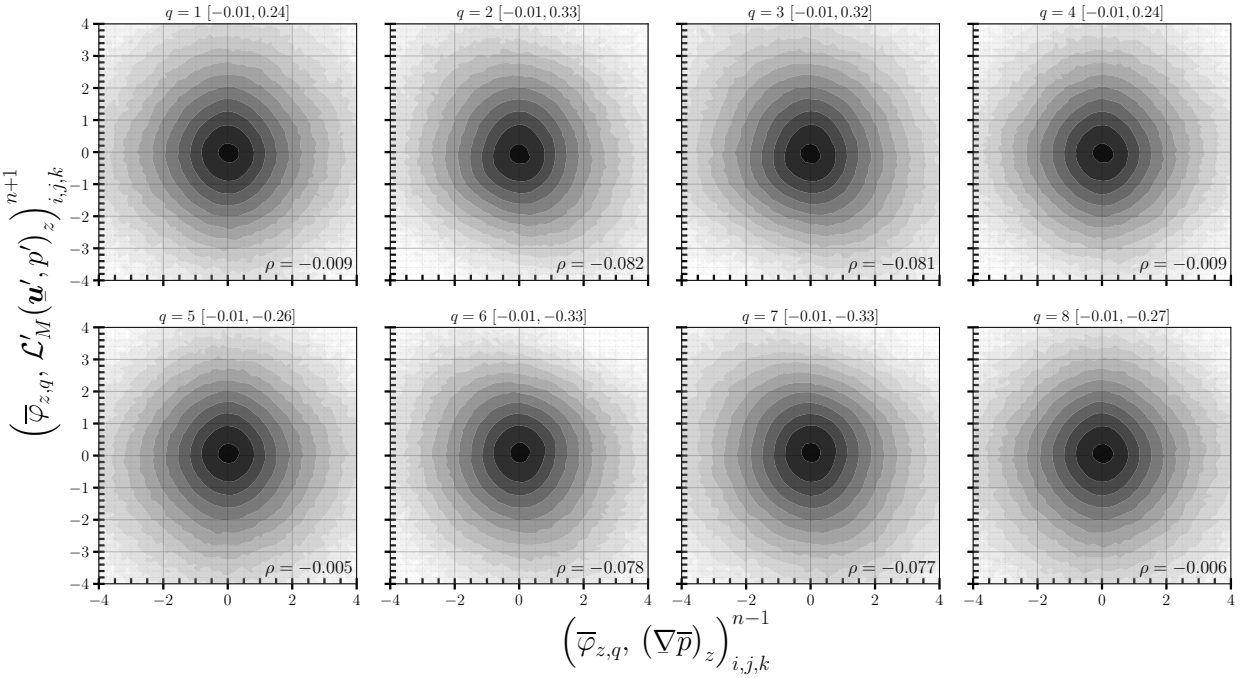


FIGURE A.219 Histograms of the t^{n-1} (i, j, k)-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

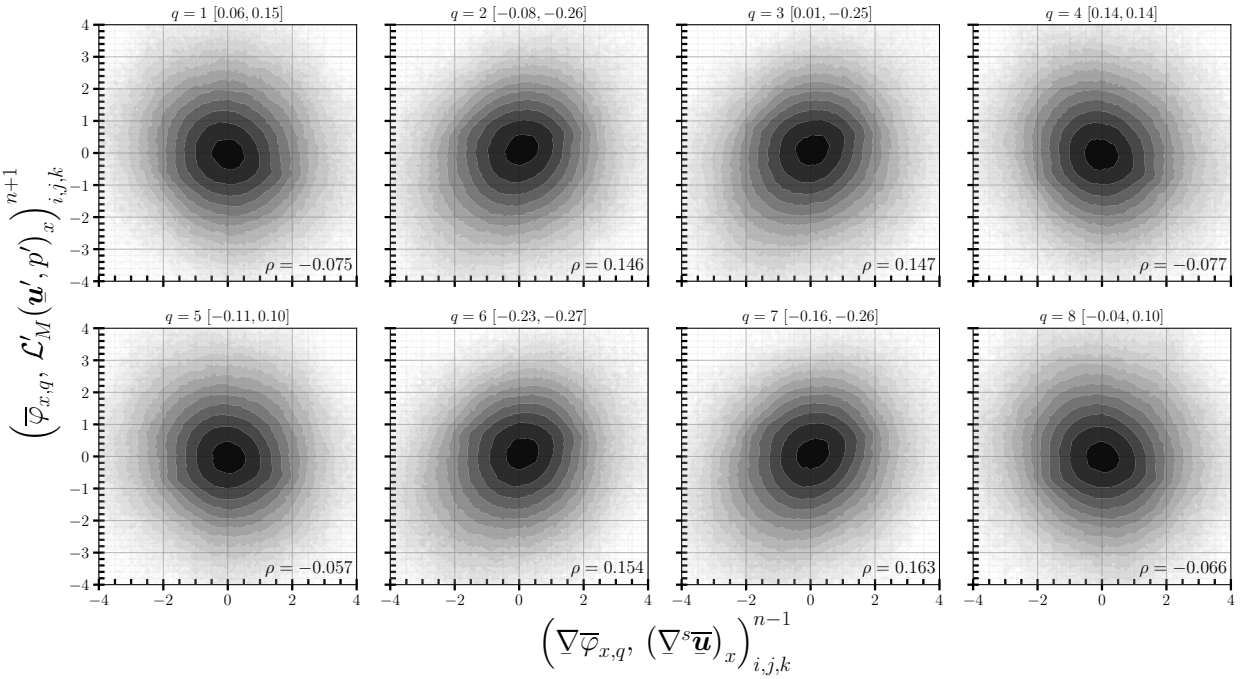


FIGURE A.220 Histograms of the t^{n-1} (i, j, k)-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

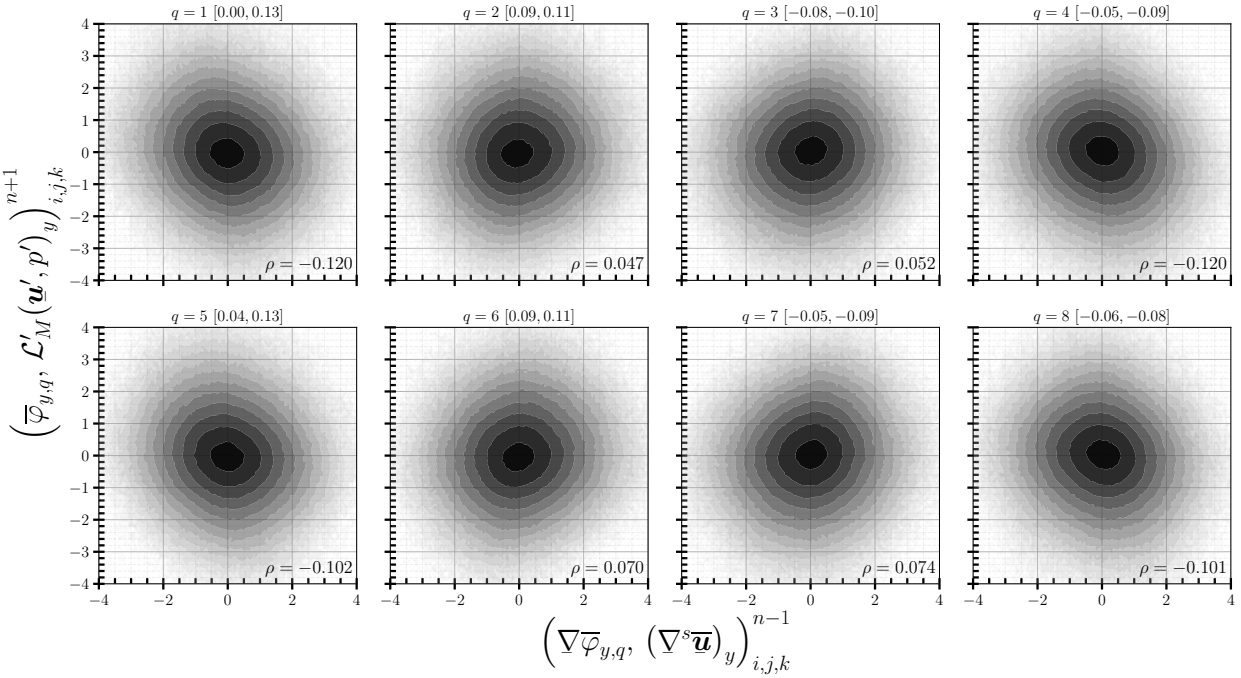


FIGURE A.221 Histograms of the t^{n-1} (i, j, k)-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

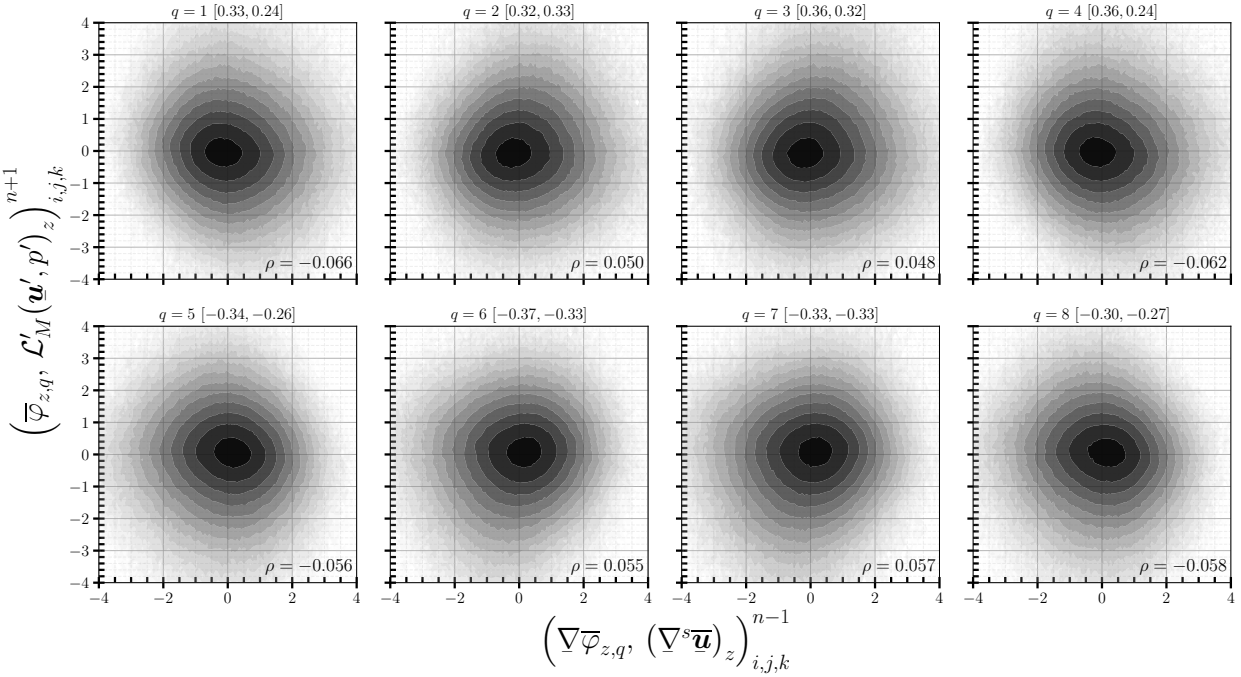


FIGURE A.222 Histograms of the t^{n-1} (i, j, k)-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.2 $(i-1, j, k)$ Histograms

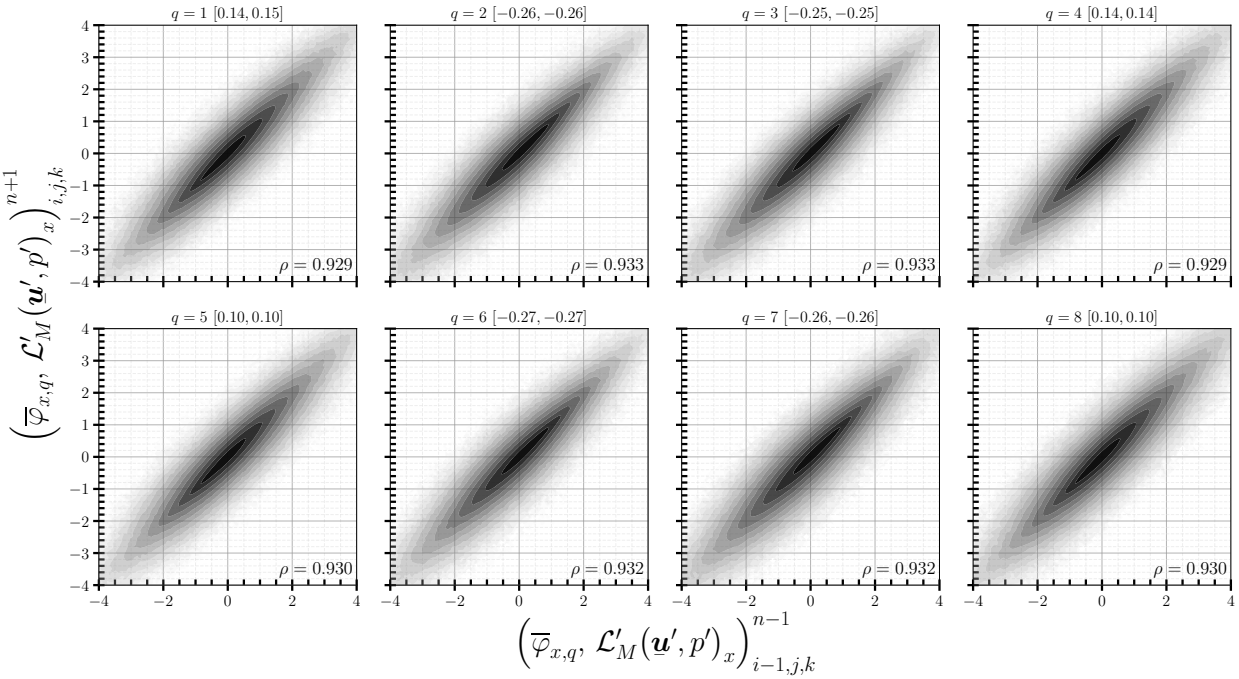


FIGURE A.223 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

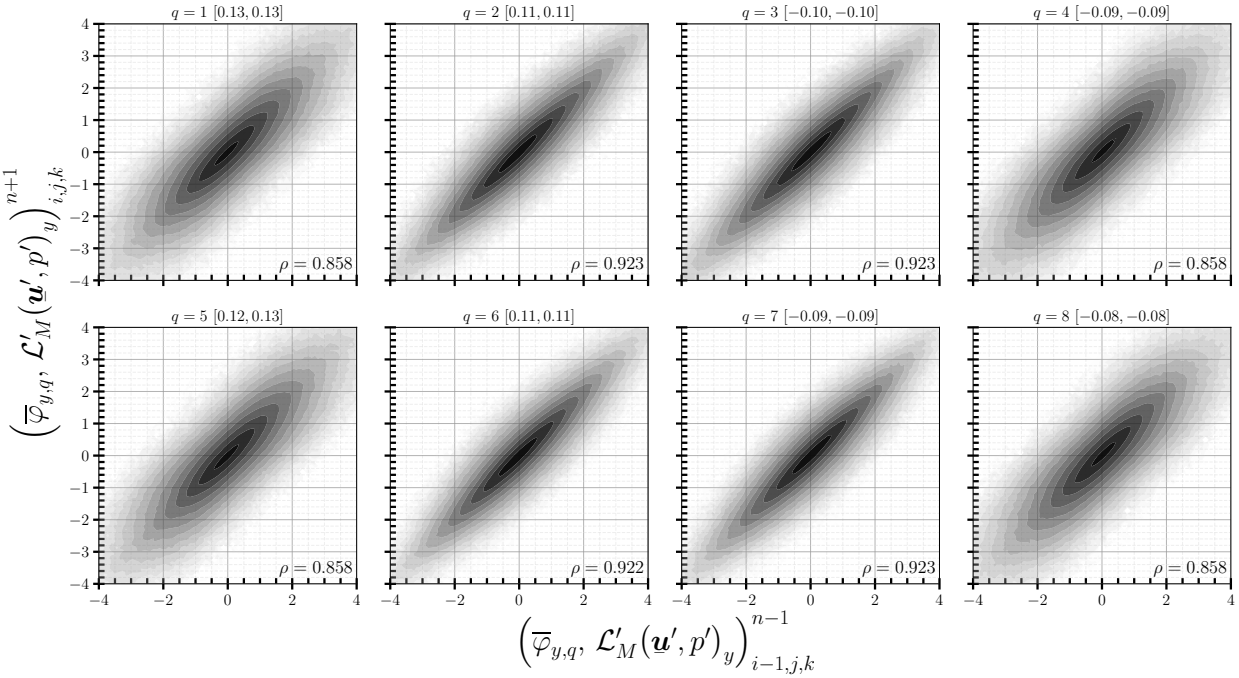


FIGURE A.224 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

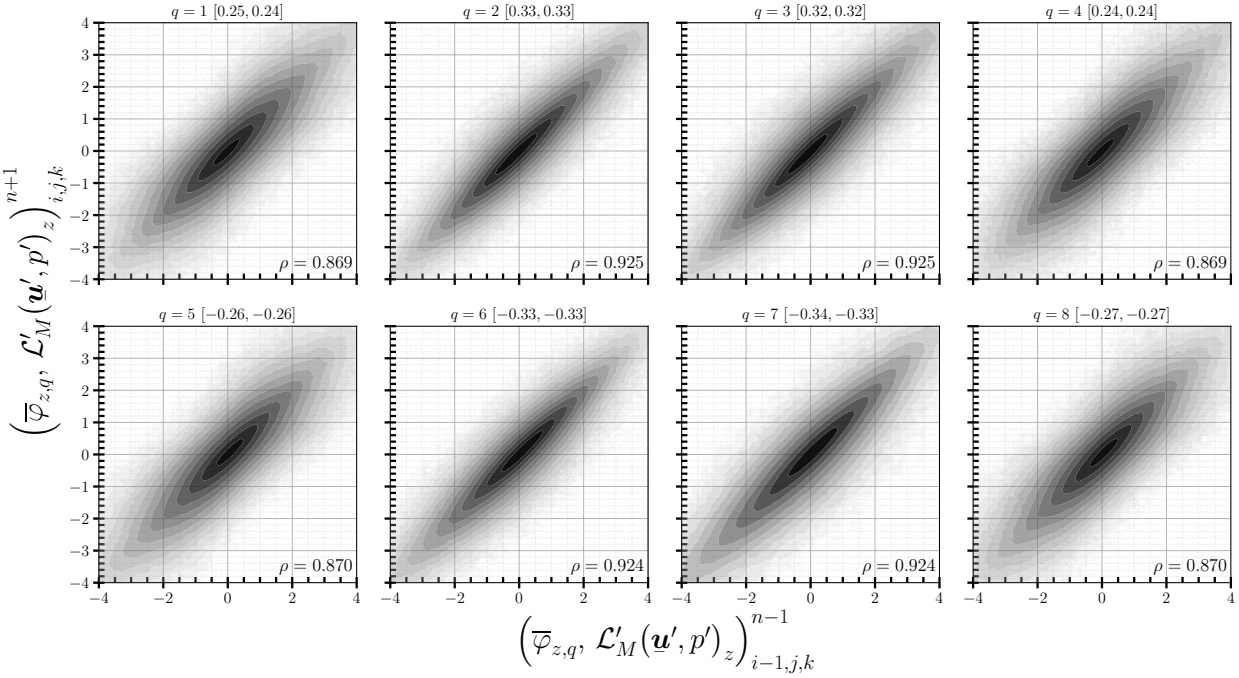


FIGURE A.225 Histograms of the t^{n-1} $(i-1, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

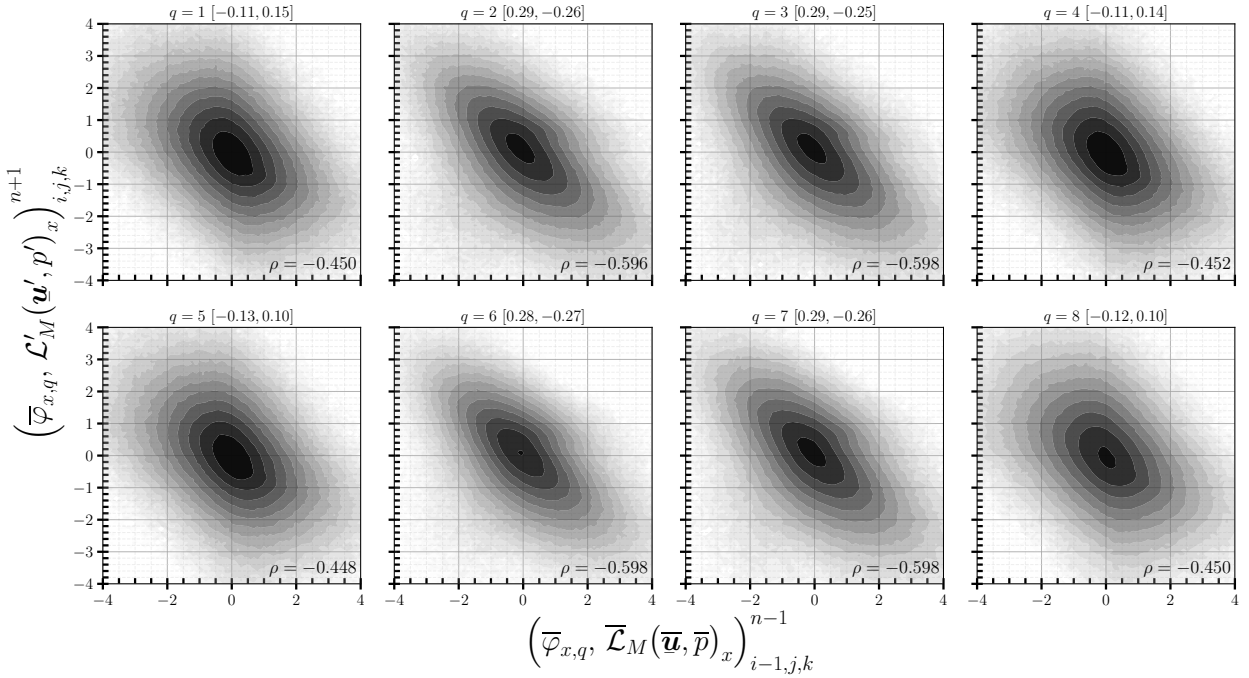


FIGURE A.226 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

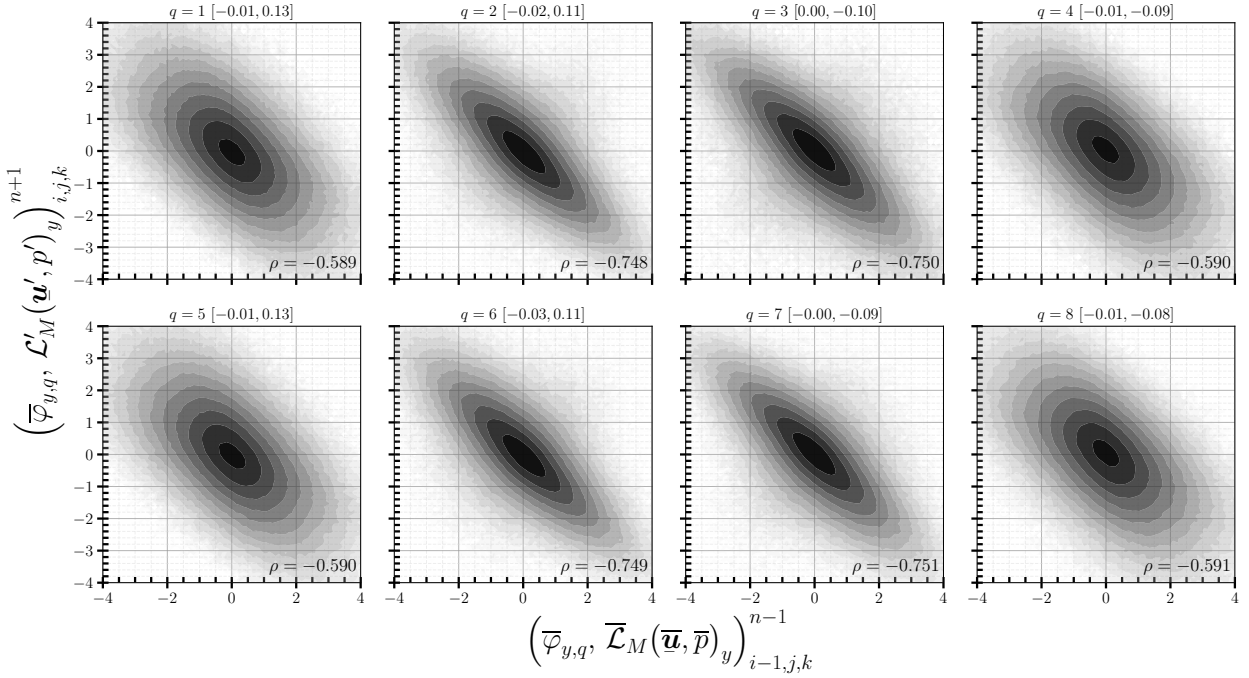


FIGURE A.227 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

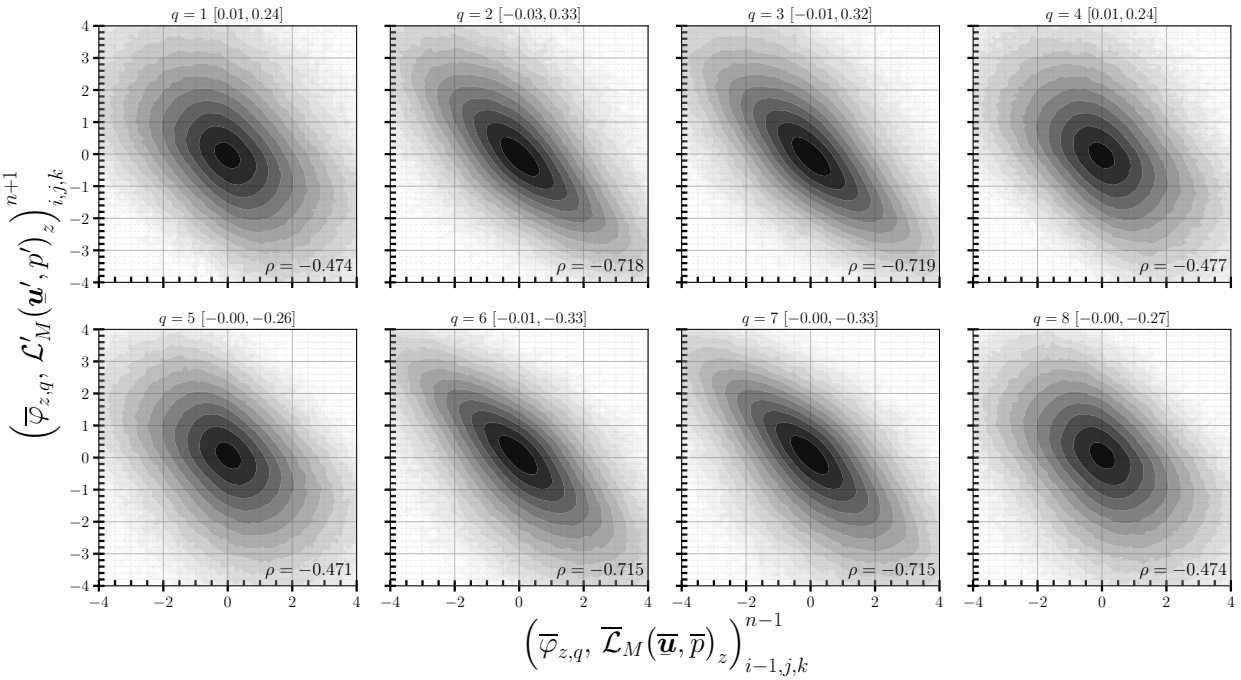


FIGURE A.228 Histograms of the t^{n-1} $(\mathrm{i}-1, \mathrm{j}, \mathrm{k})$-element z-momentum coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

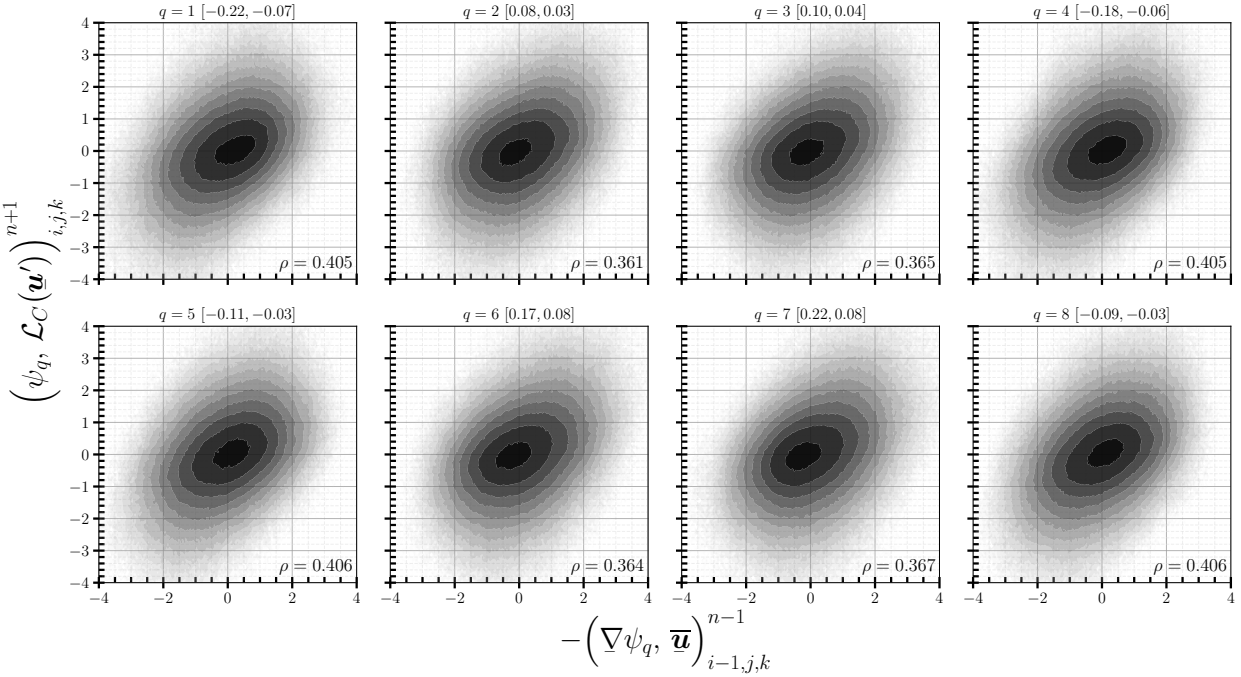


FIGURE A.229 Histograms of the t^{n-1} $(\mathrm{i}-1, \mathrm{j}, \mathrm{k})$-element continuity coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

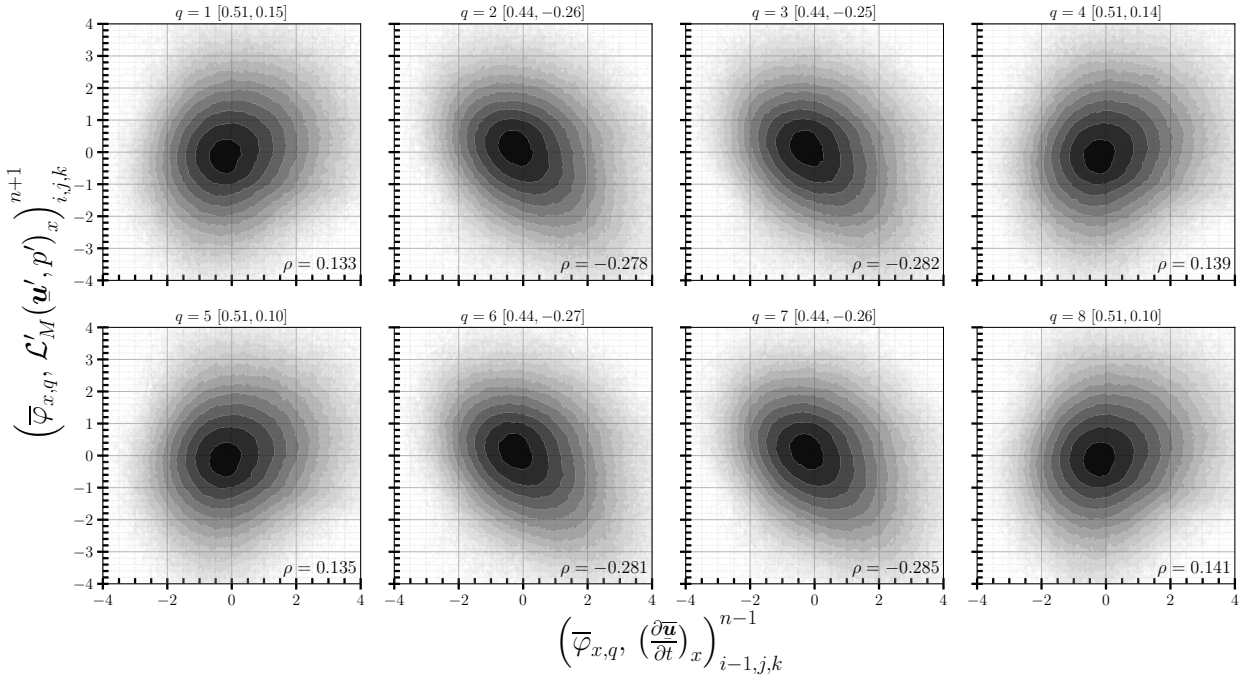


FIGURE A.230 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

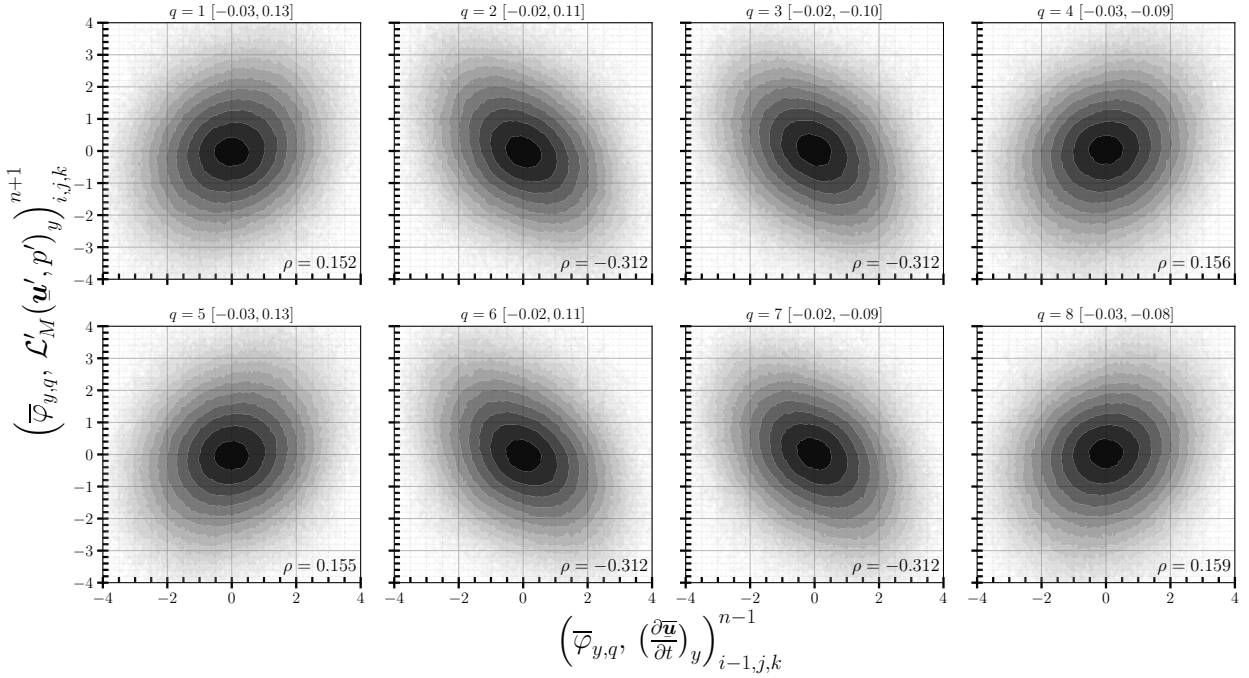


FIGURE A.231 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

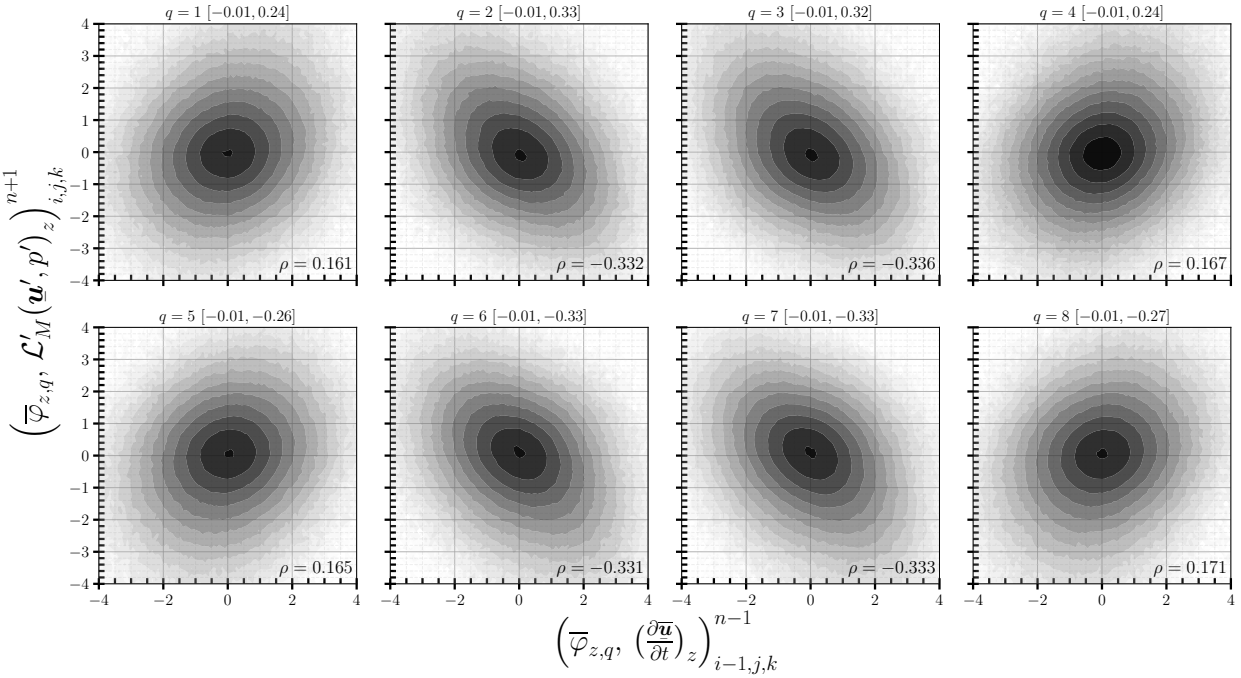


FIGURE A.232 Histograms of the t^{n-1} $(i-1, j, k)$-element *z*-momentum time derivative term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

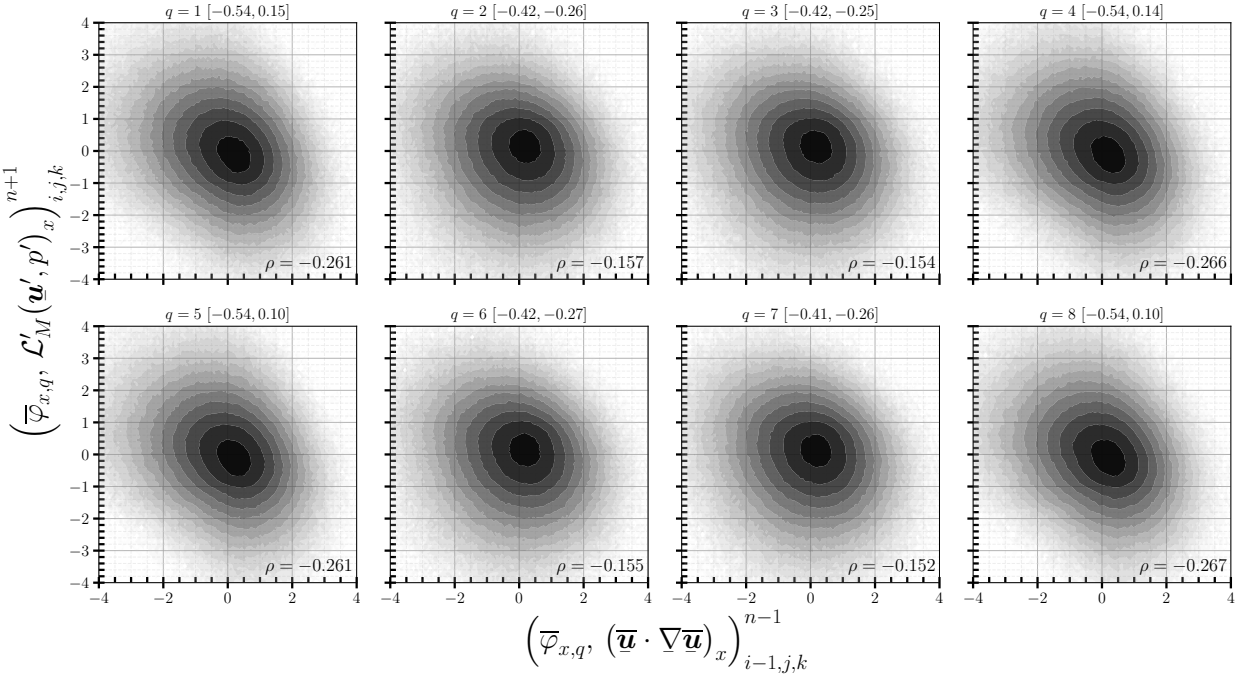


FIGURE A.233 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

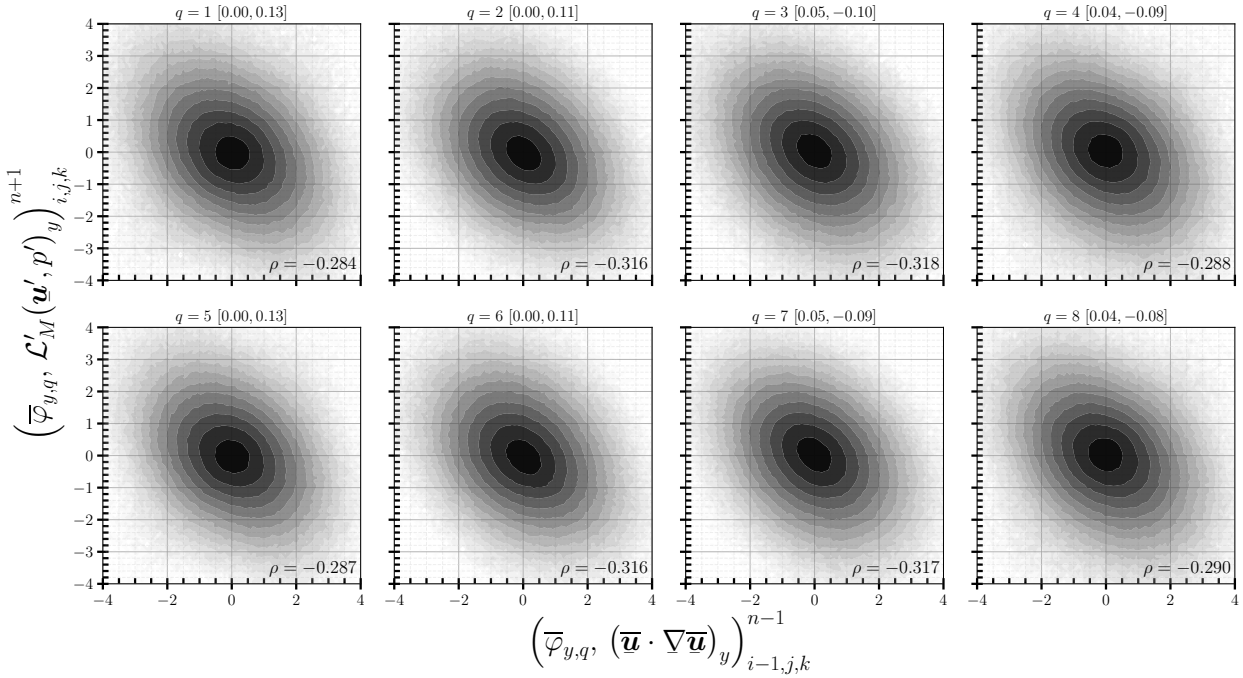


FIGURE A.234 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

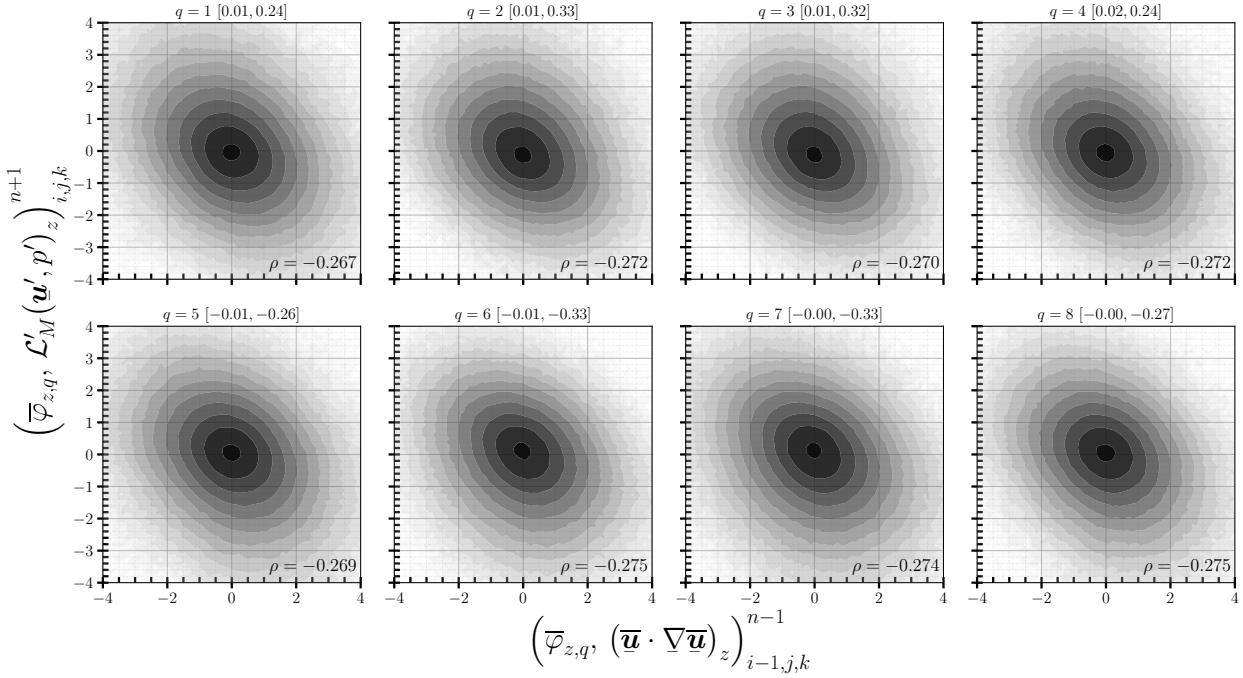


FIGURE A.235 Histograms of the t^{n-1} $(i-1, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

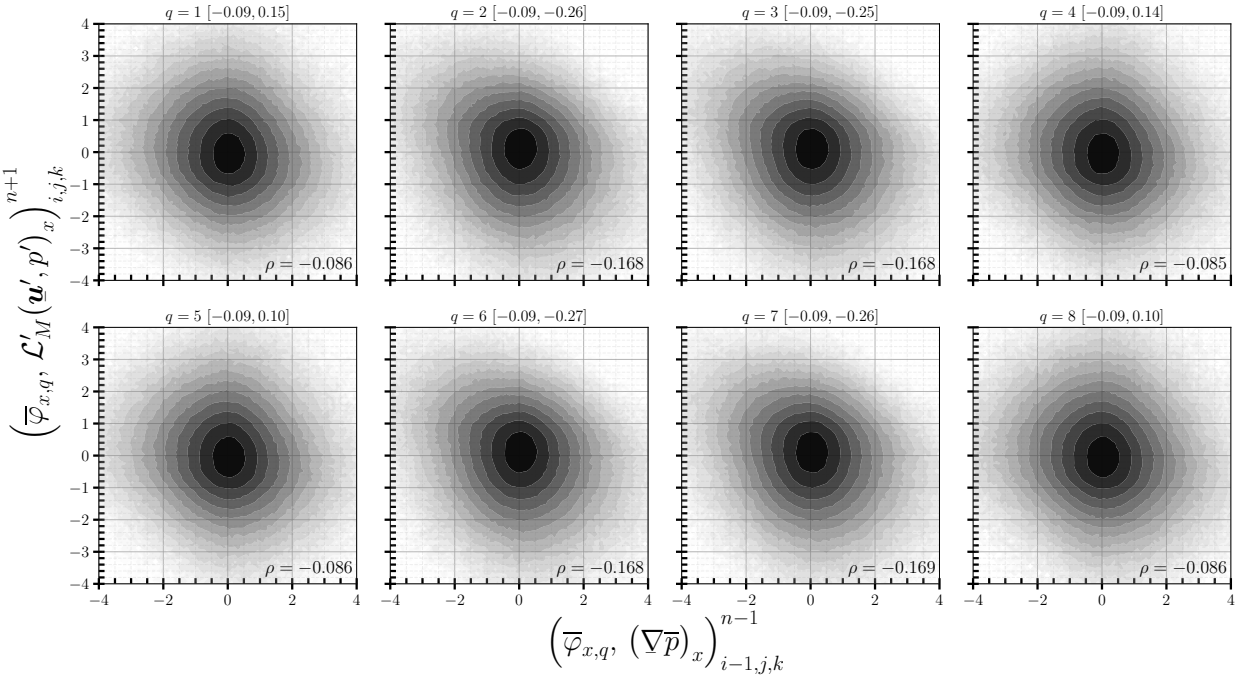


FIGURE A.236 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

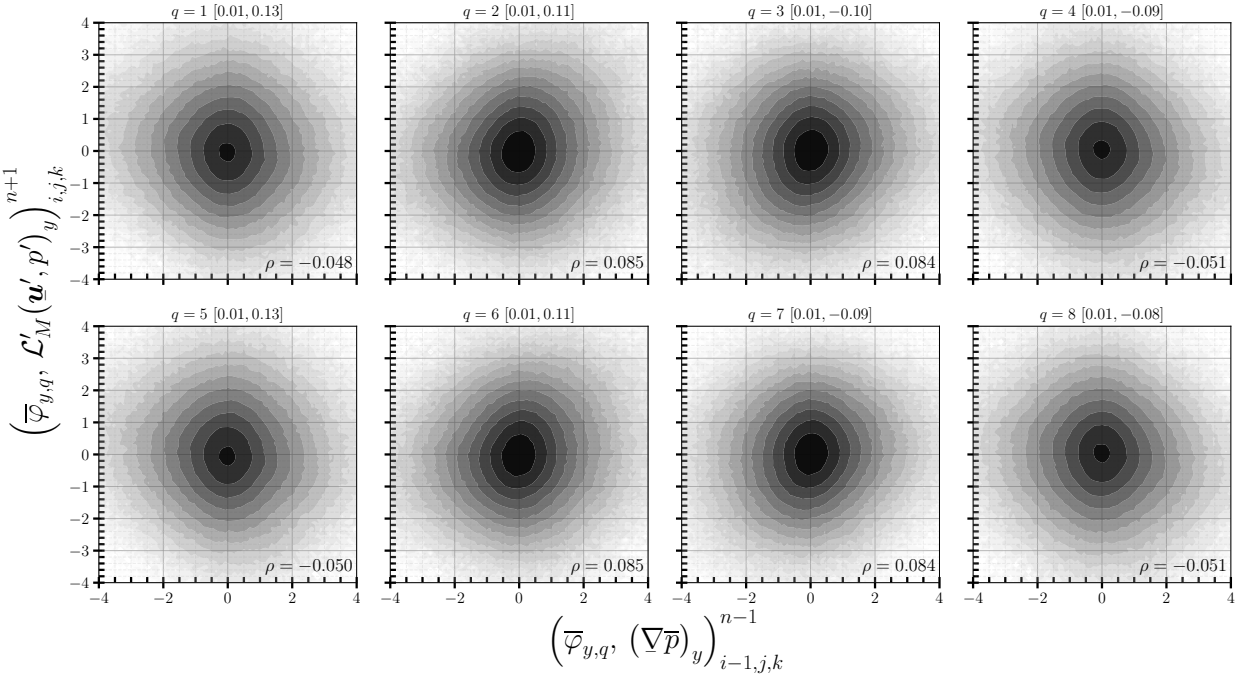


FIGURE A.237 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

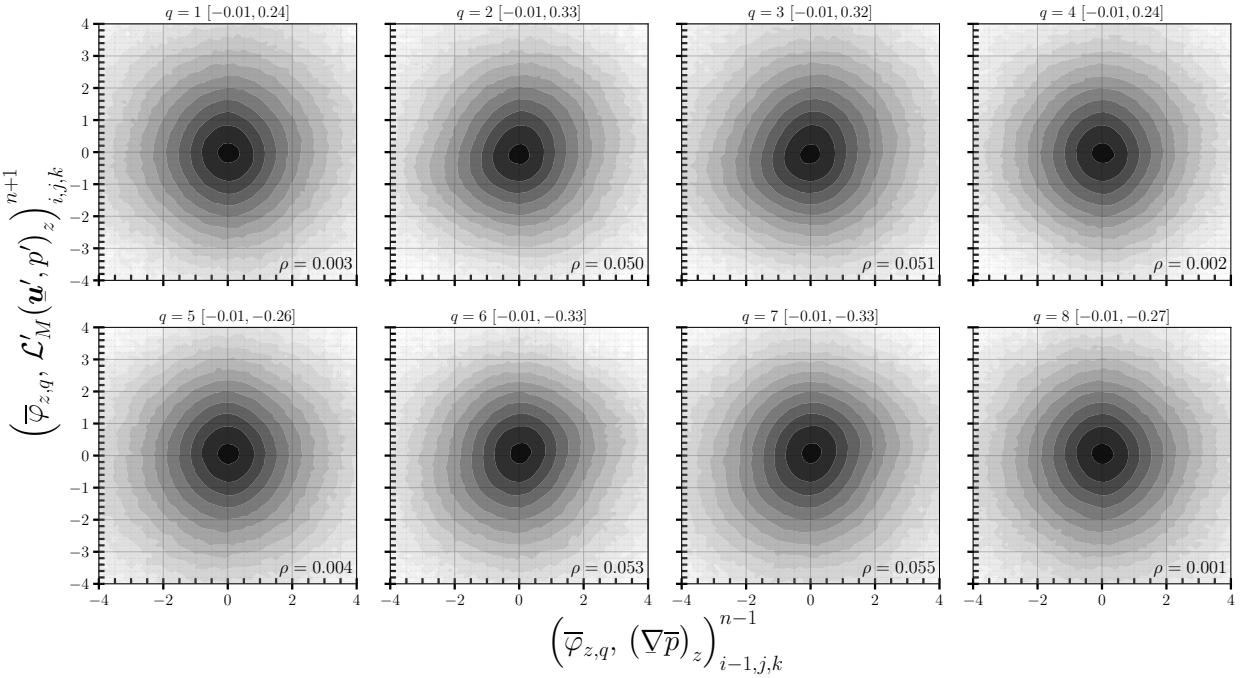


FIGURE A.238 Histograms of the t^{n-1} $(i-1, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

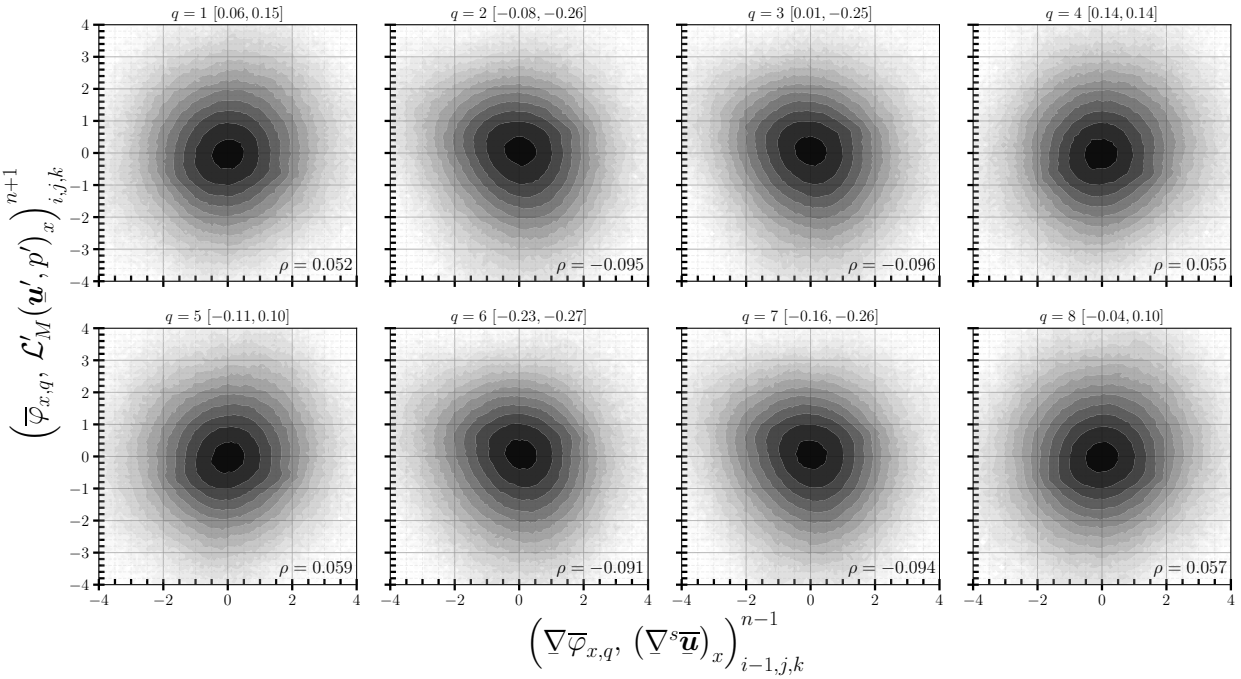


FIGURE A.239 Histograms of the t^{n-1} $(i-1, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

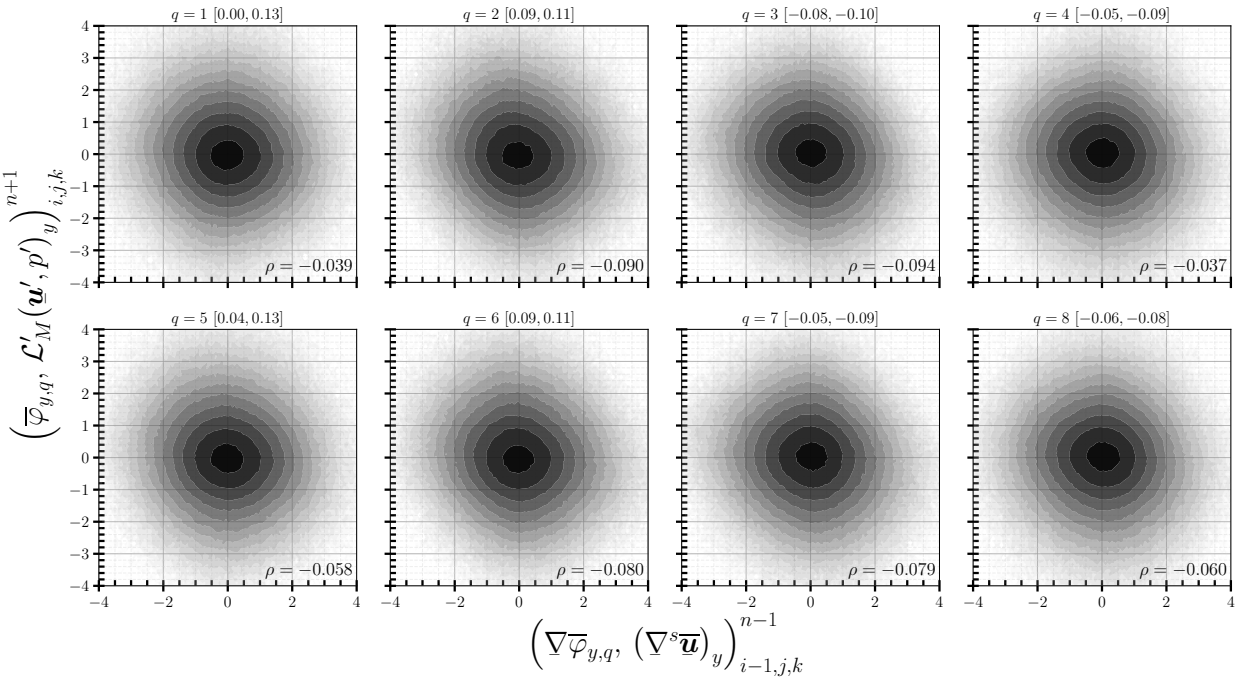


FIGURE A.240 Histograms of the t^{n-1} $(i-1, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

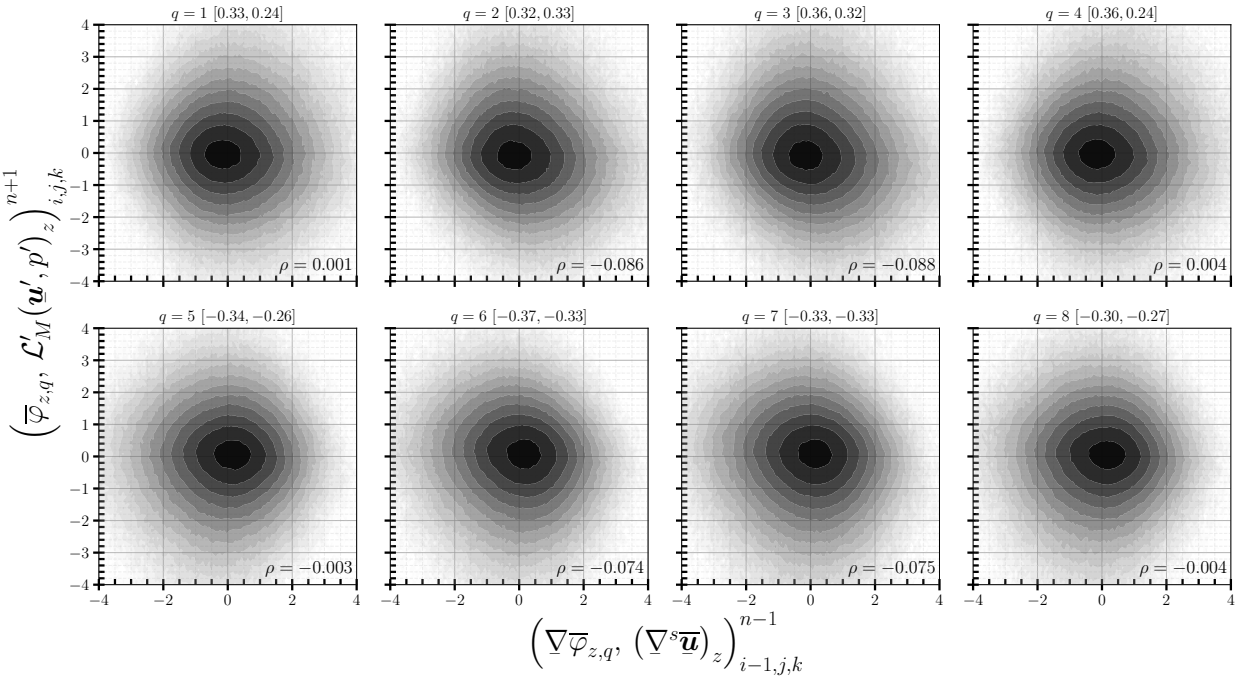


FIGURE A.241 Histograms of the t^{n-1} $(i-1, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.3 $(i-2, j, k)$ Histograms

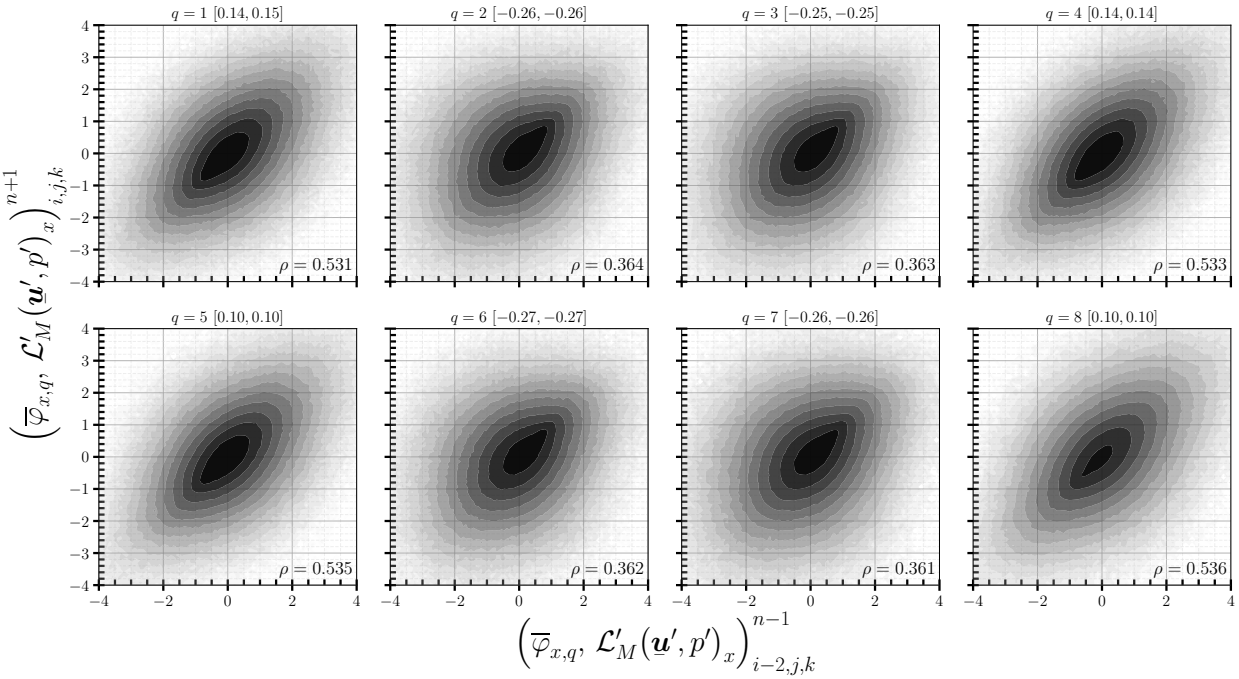


Figure A.242 Histograms of the t^{n-1} $(i-2, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

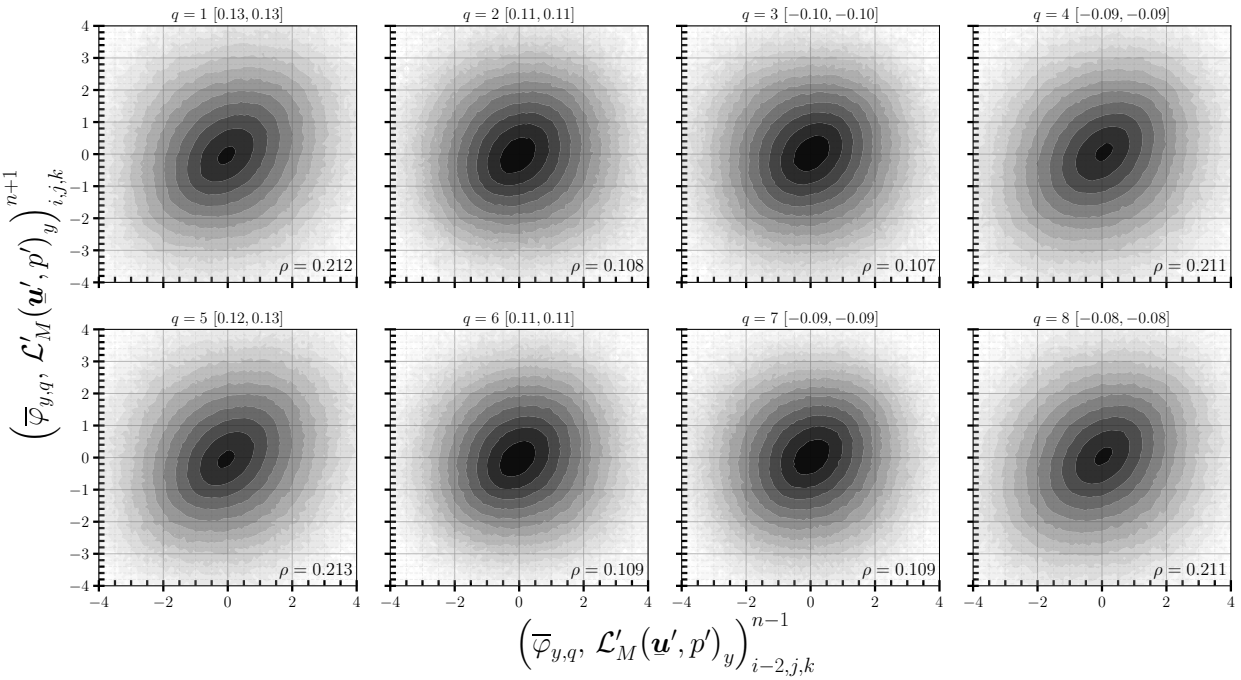


FIGURE A.243 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

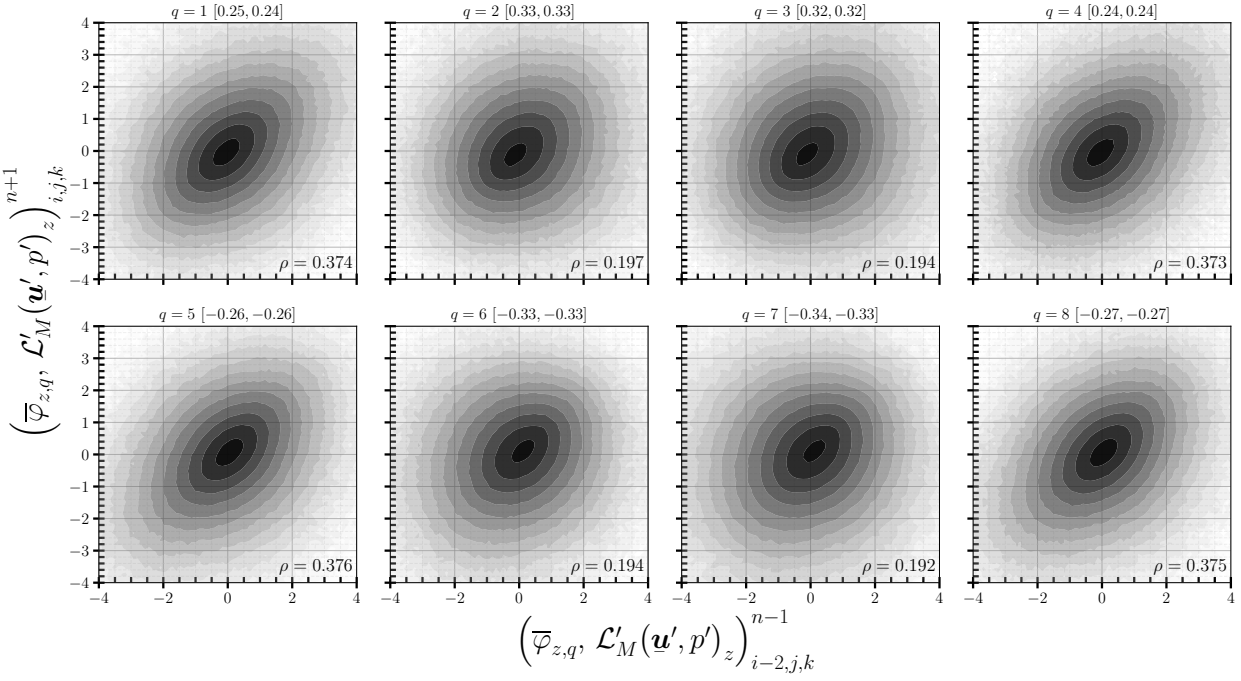


FIGURE A.244 Histograms of the t^{n-1} $(i-2, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

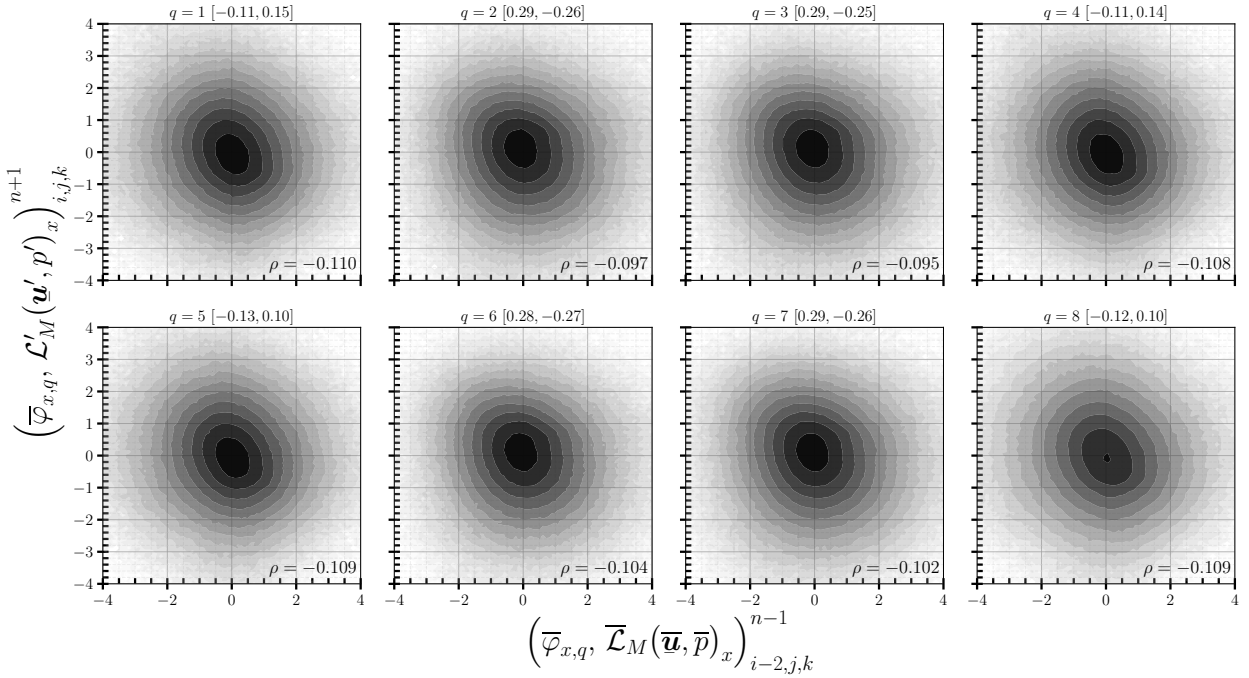


FIGURE A.245 Histograms of the t^{n-1} $(i-2, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

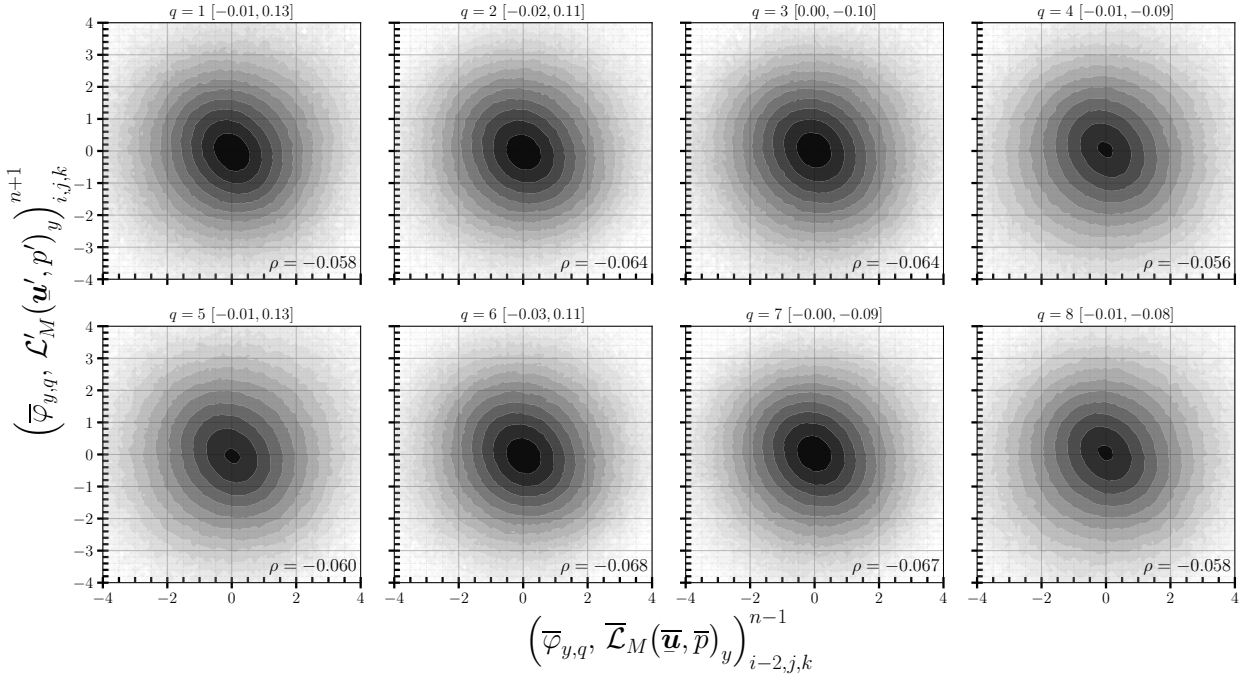


FIGURE A.246 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

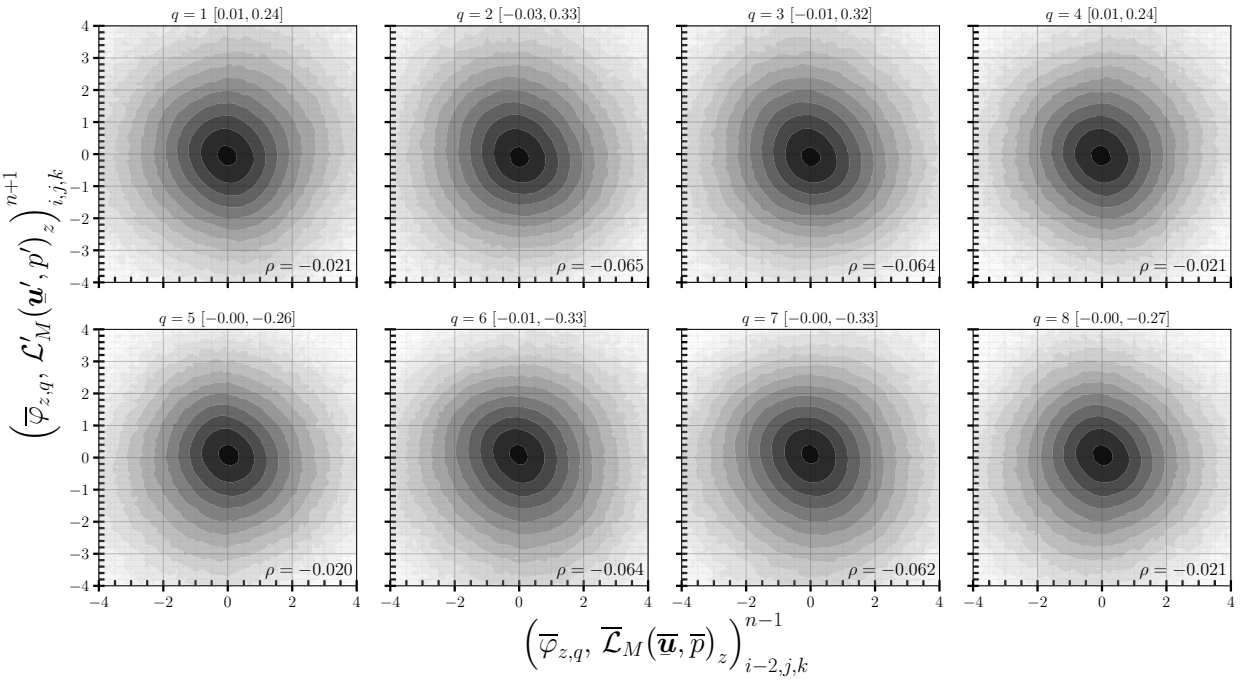


FIGURE A.247 Histograms of the t^{n-1} $(\mathrm{i}-2, \mathrm{j}, \mathrm{k})$-element z-momentum coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

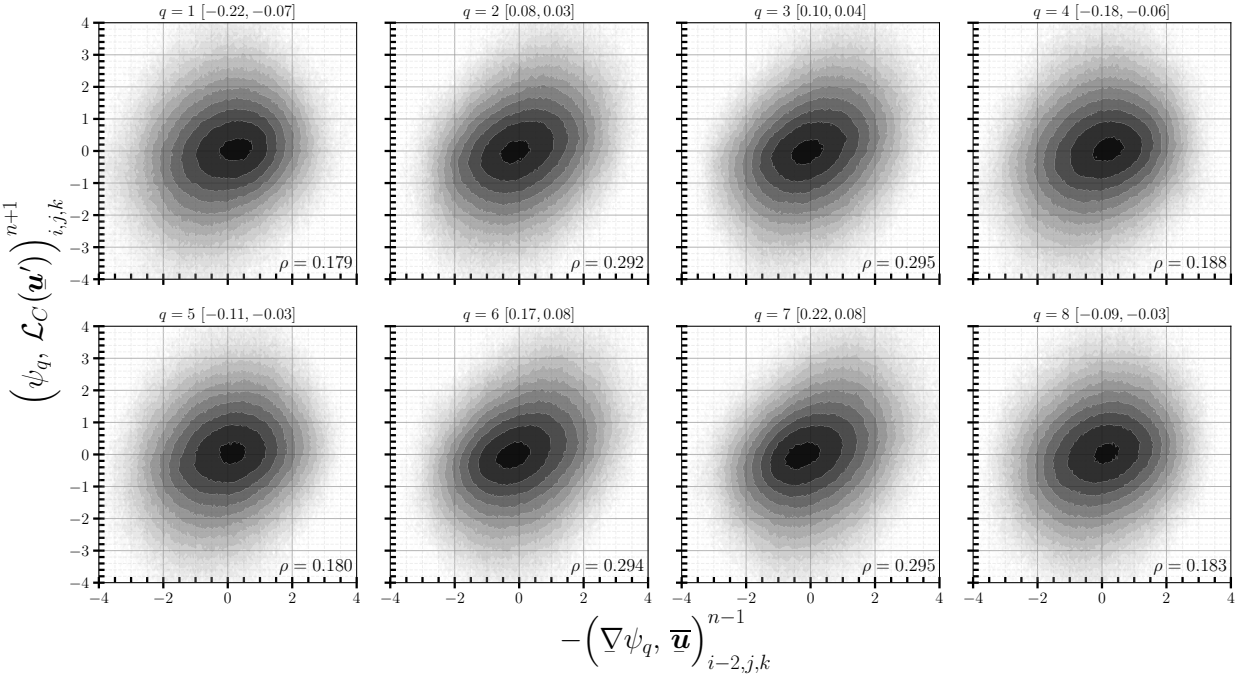


FIGURE A.248 Histograms of the t^{n-1} $(\mathrm{i}-2, \mathrm{j}, \mathrm{k})$-element continuity coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

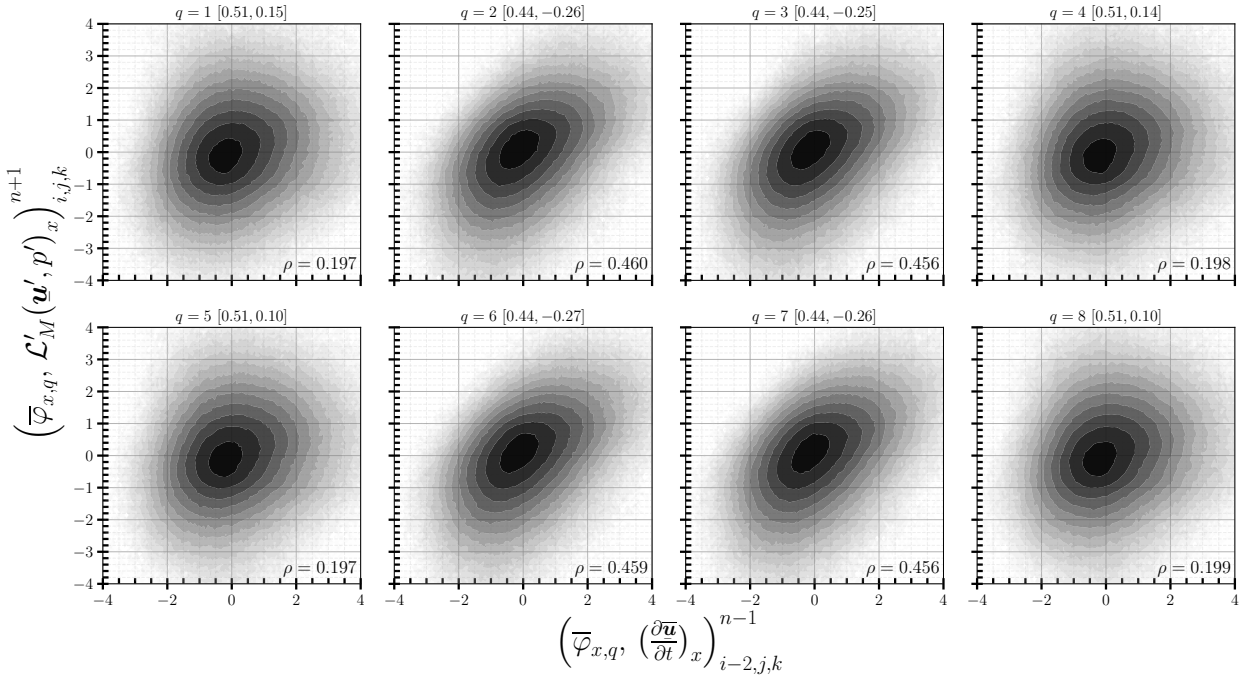


FIGURE A.249 Histograms of the t^{n-1} $(i-2, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

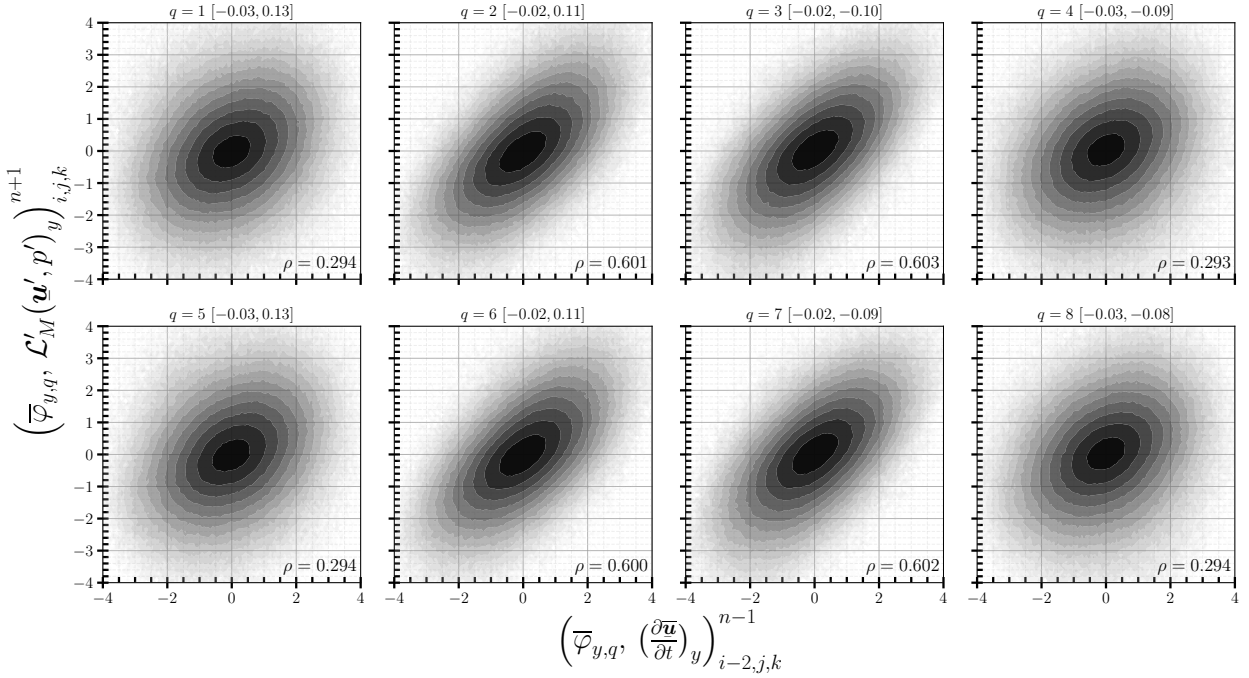


FIGURE A.250 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

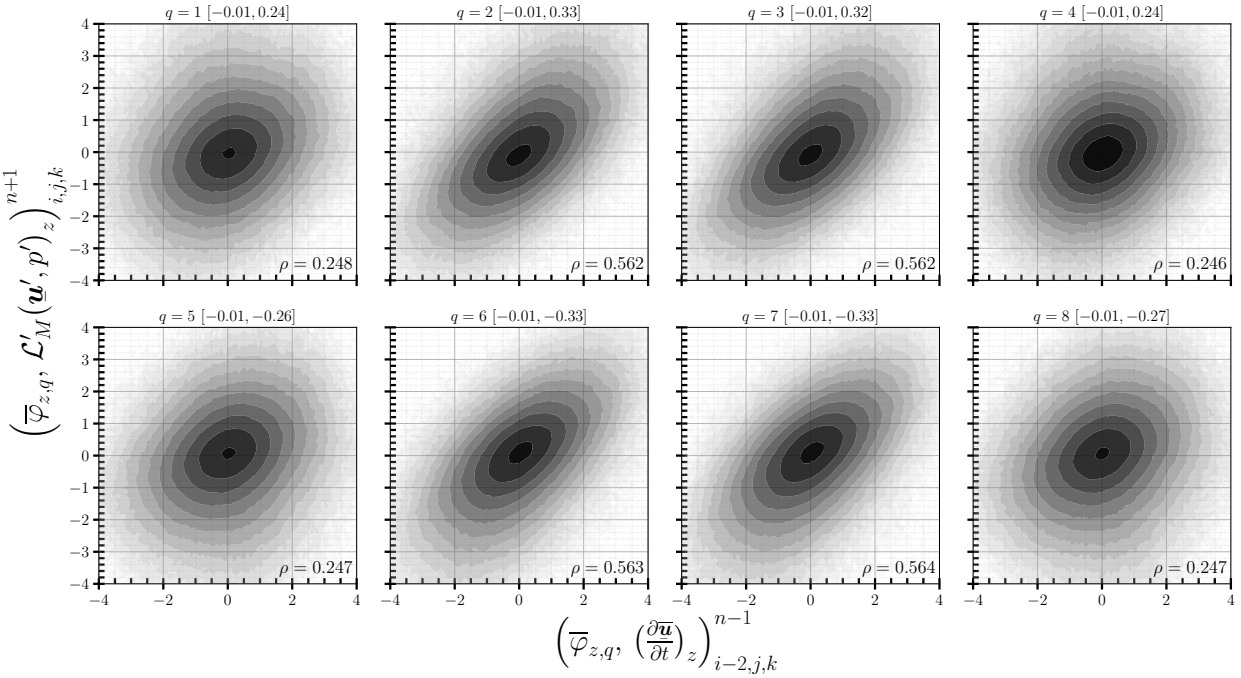


FIGURE A.251 Histograms of the t^{n-1} $(i-2, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

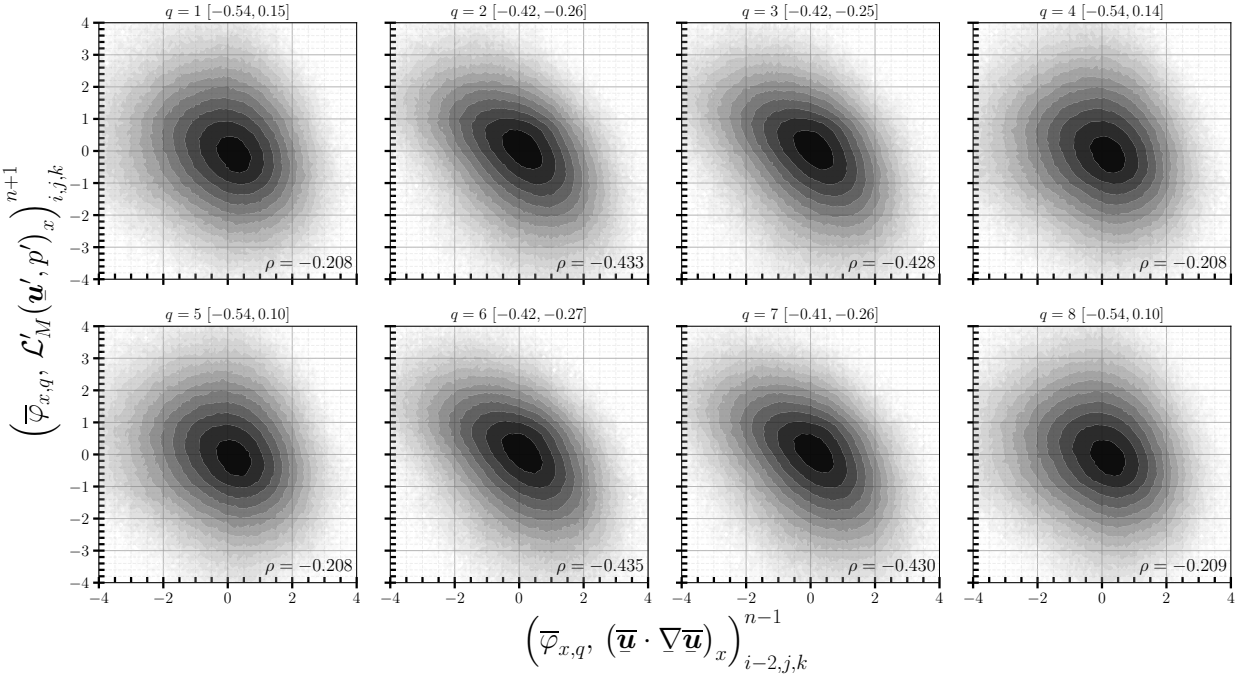


FIGURE A.252 Histograms of the t^{n-1} $(i-2, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

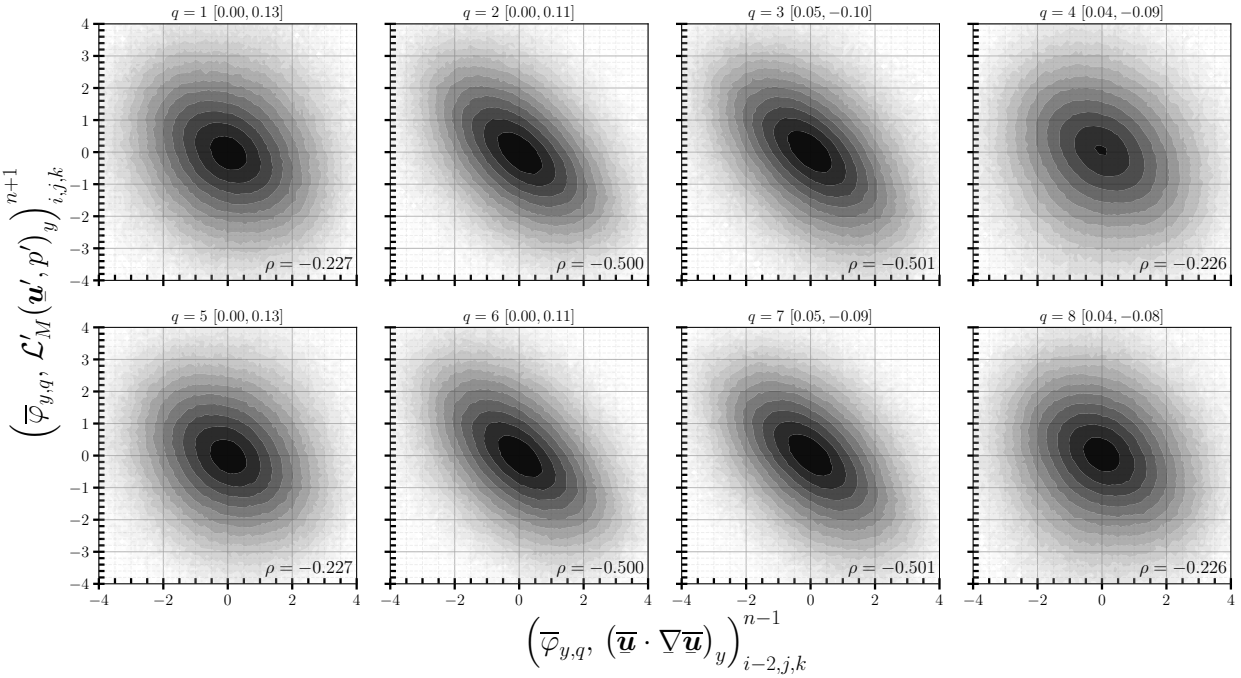


FIGURE A.253 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

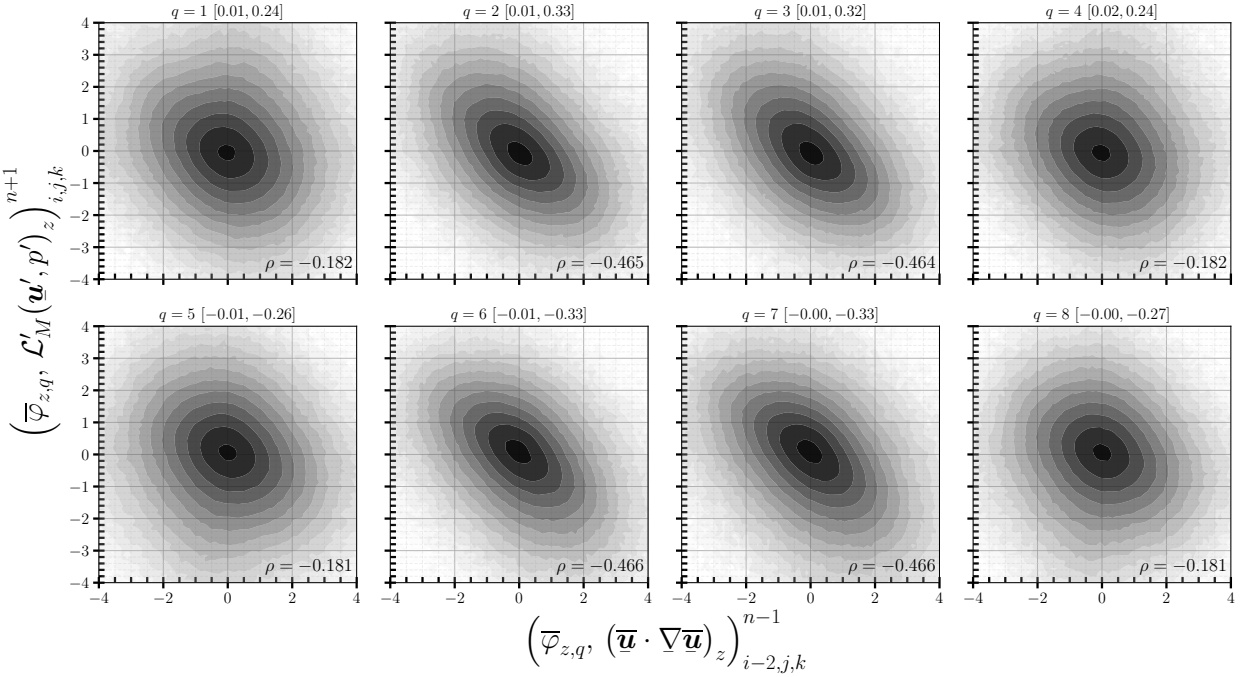


FIGURE A.254 Histograms of the t^{n-1} $(i-2, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

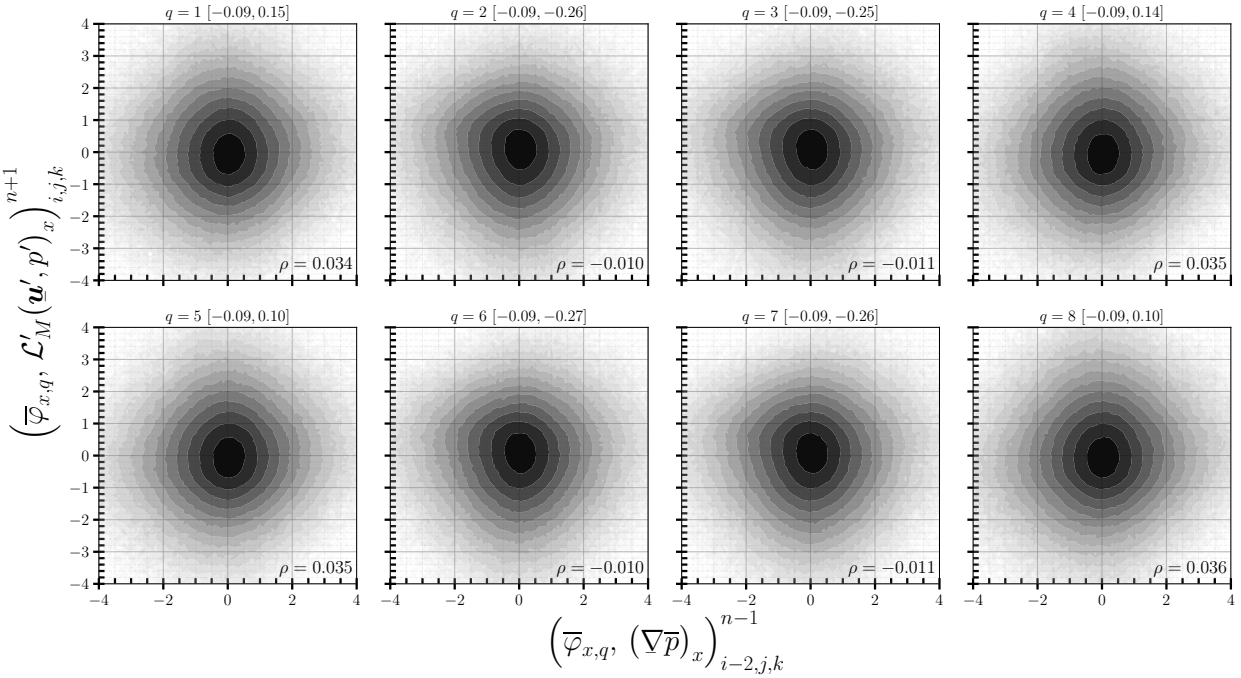


FIGURE A.255 Histograms of the t^{n-1} $(i-2, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

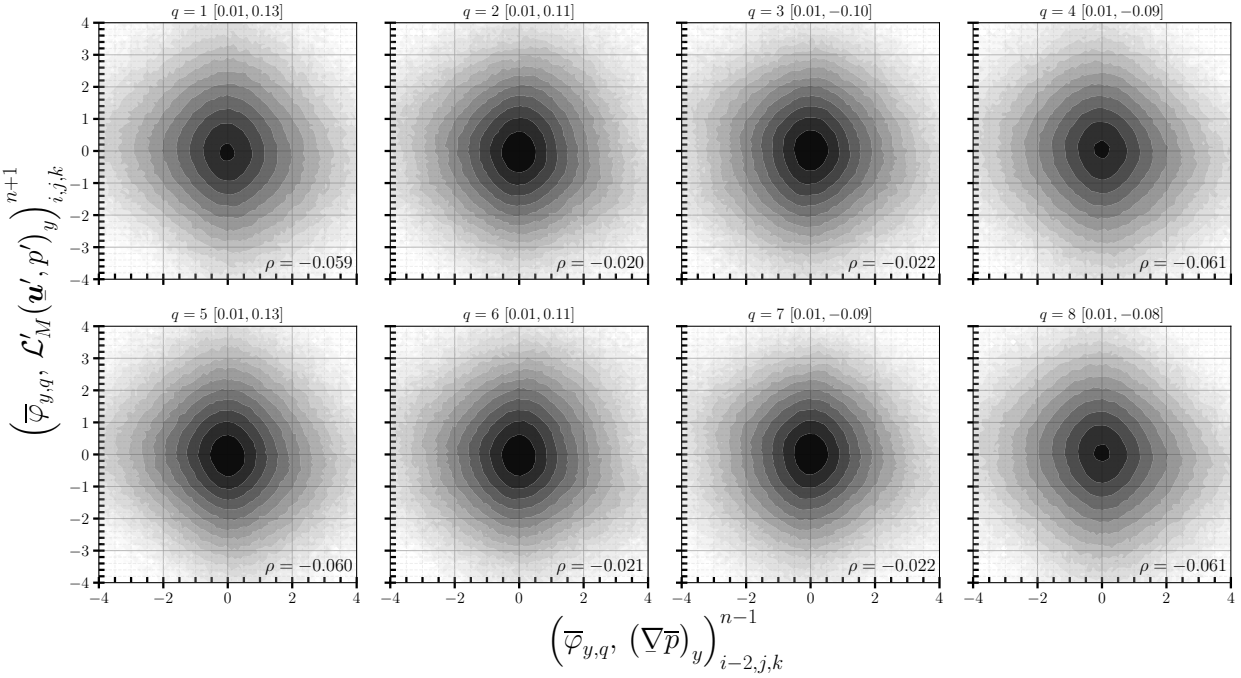


FIGURE A.256 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

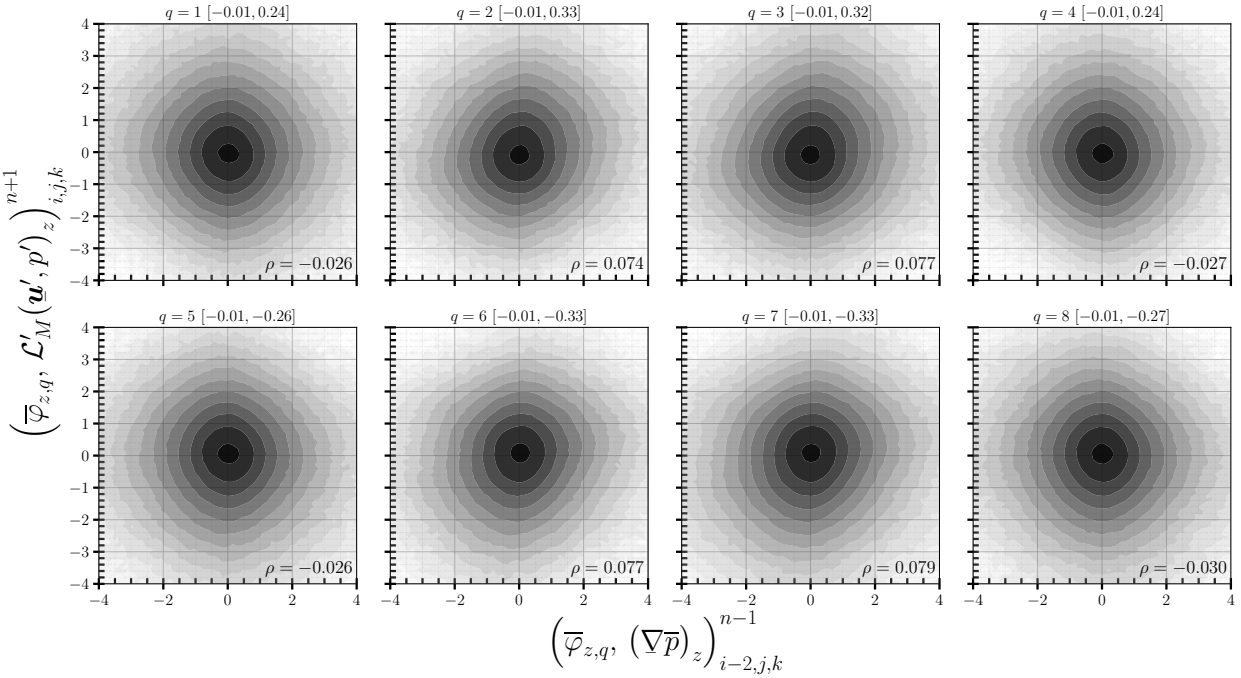


FIGURE A.257 Histograms of the t^{n-1} $(i-2, j, k)$-element *z*-momentum pressure gradient term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

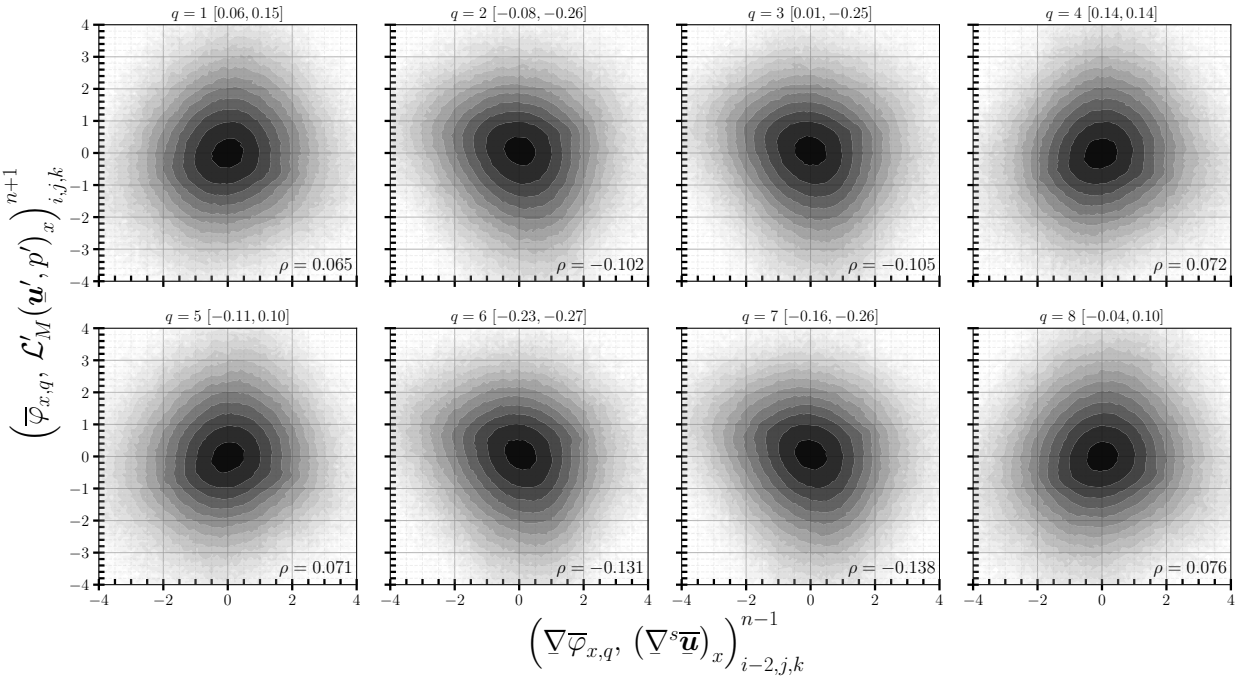


FIGURE A.258 Histograms of the t^{n-1} $(i-2, j, k)$-element *x*-momentum diffusion term against the t^{n+1} (i, j, k)-element *x*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

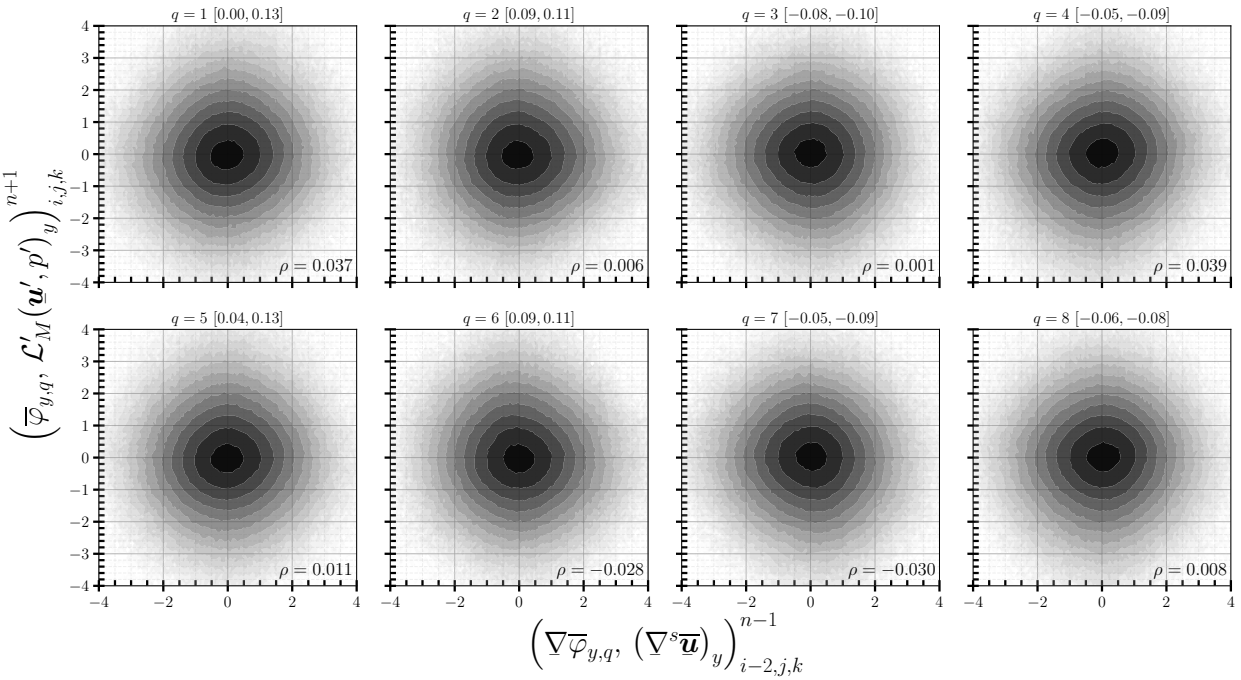


FIGURE A.259 Histograms of the t^{n-1} $(i-2, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

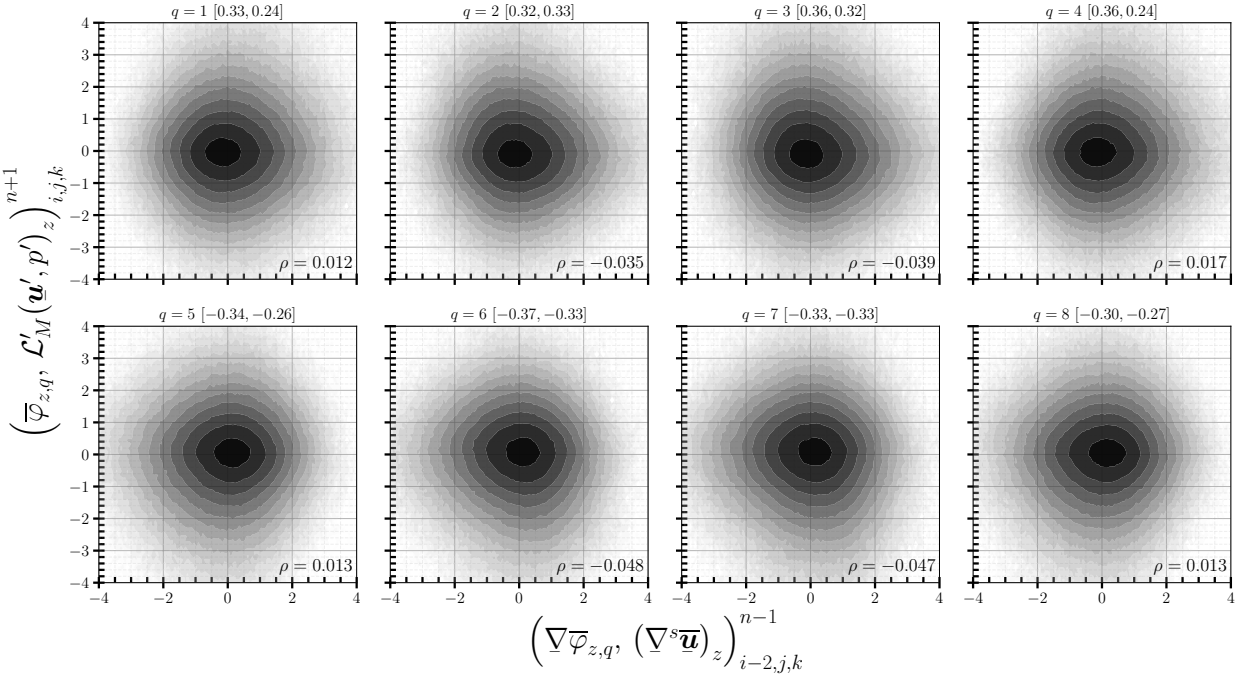


FIGURE A.260 Histograms of the t^{n-1} $(i-2, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.4 $(i-3, j, k)$ Histograms

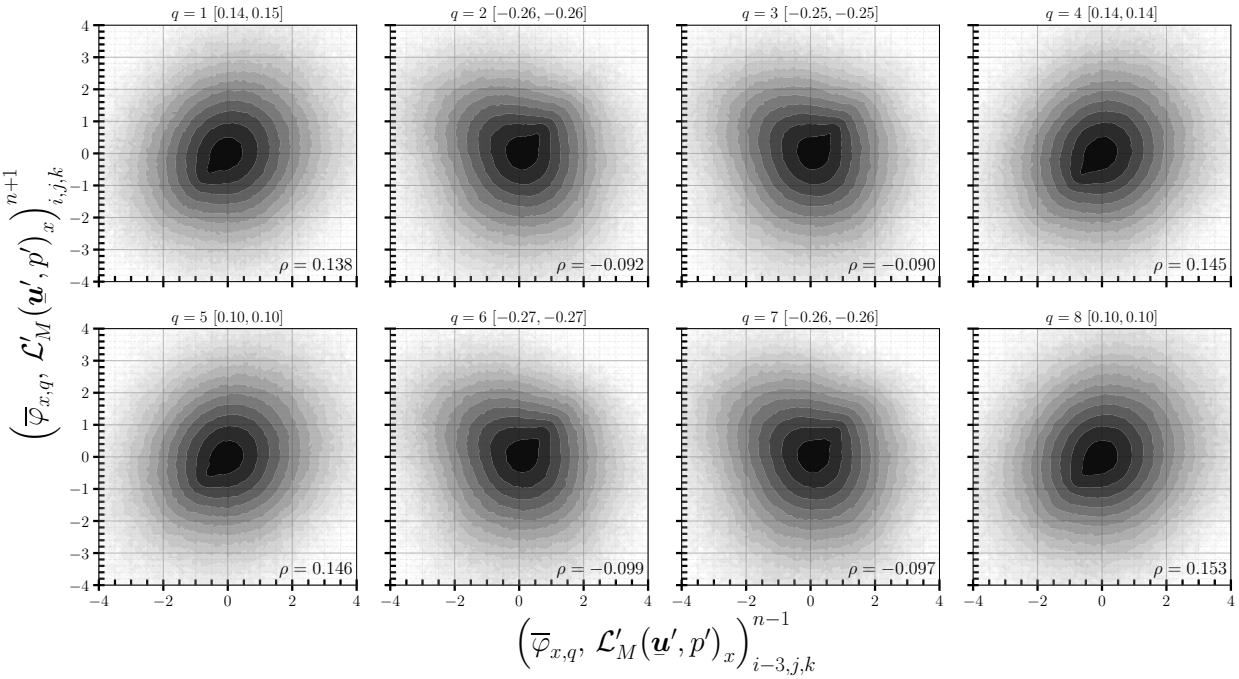


FIGURE A.261 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

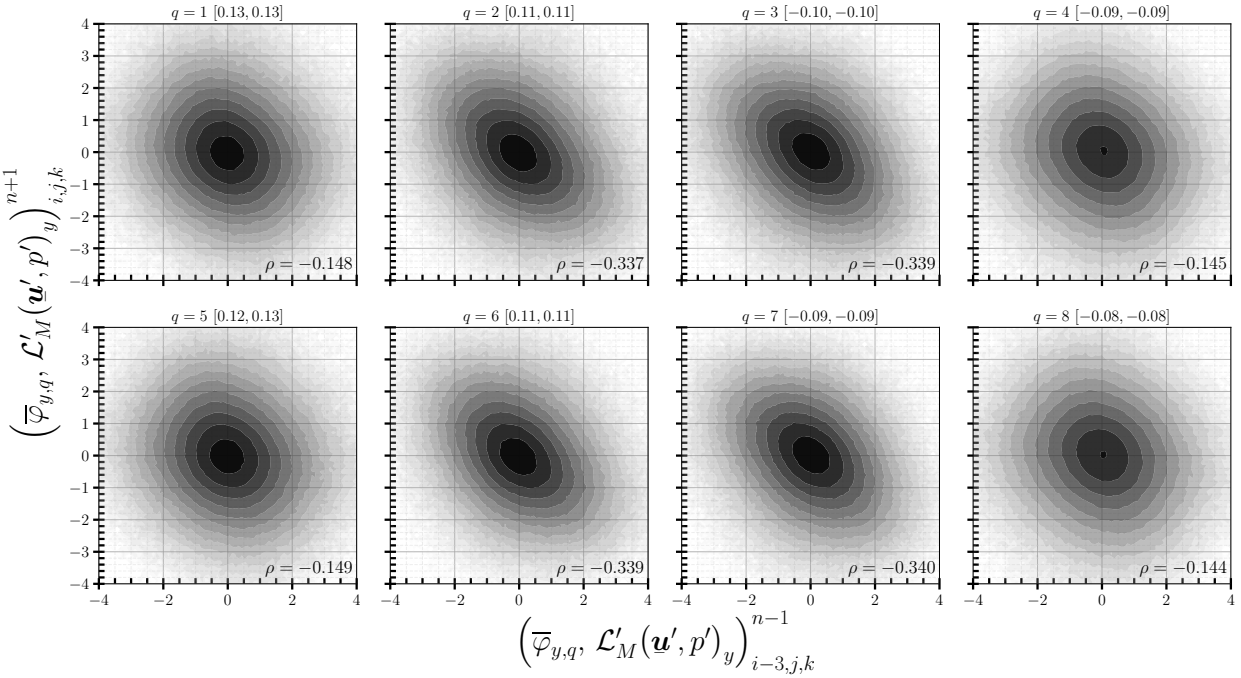


FIGURE A.262 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

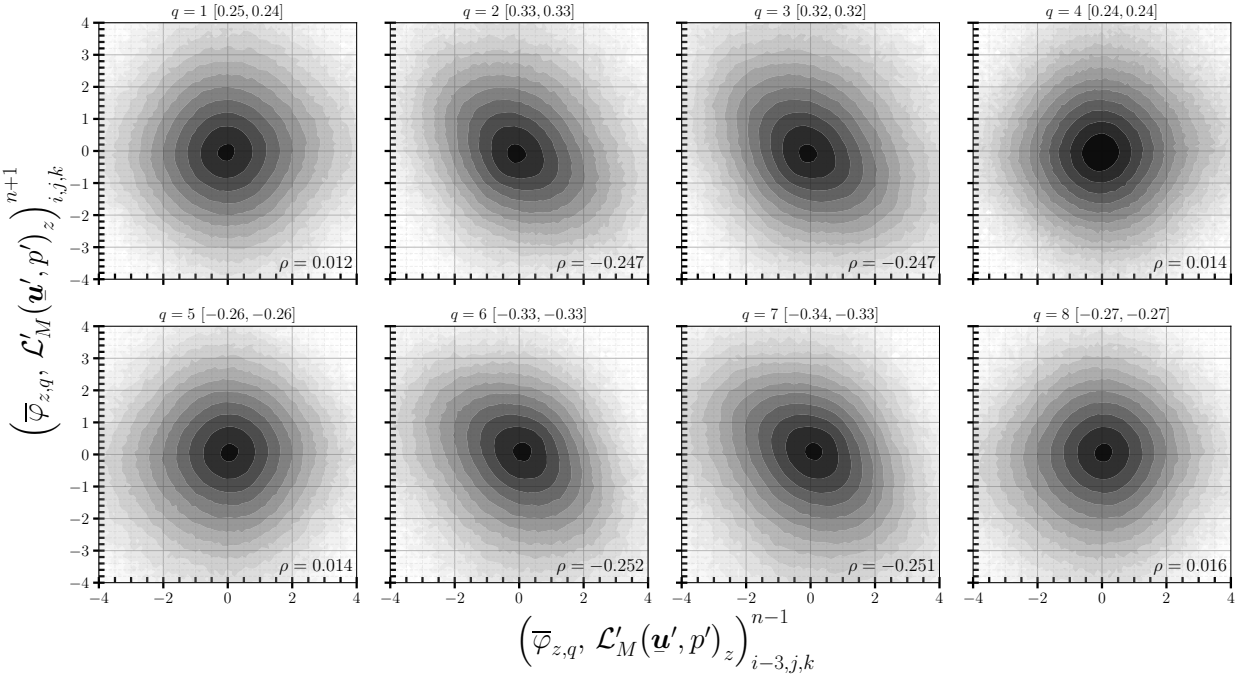


FIGURE A.263 Histograms of the t^{n-1} $(i-3, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

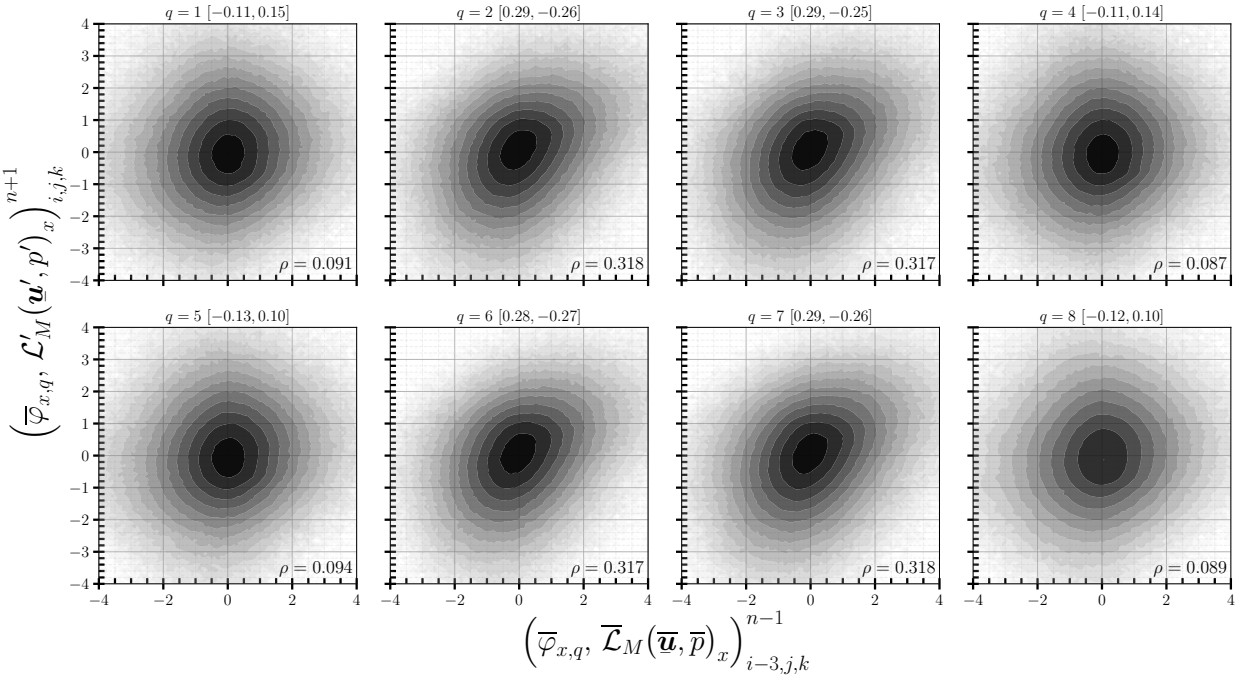


FIGURE A.264 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

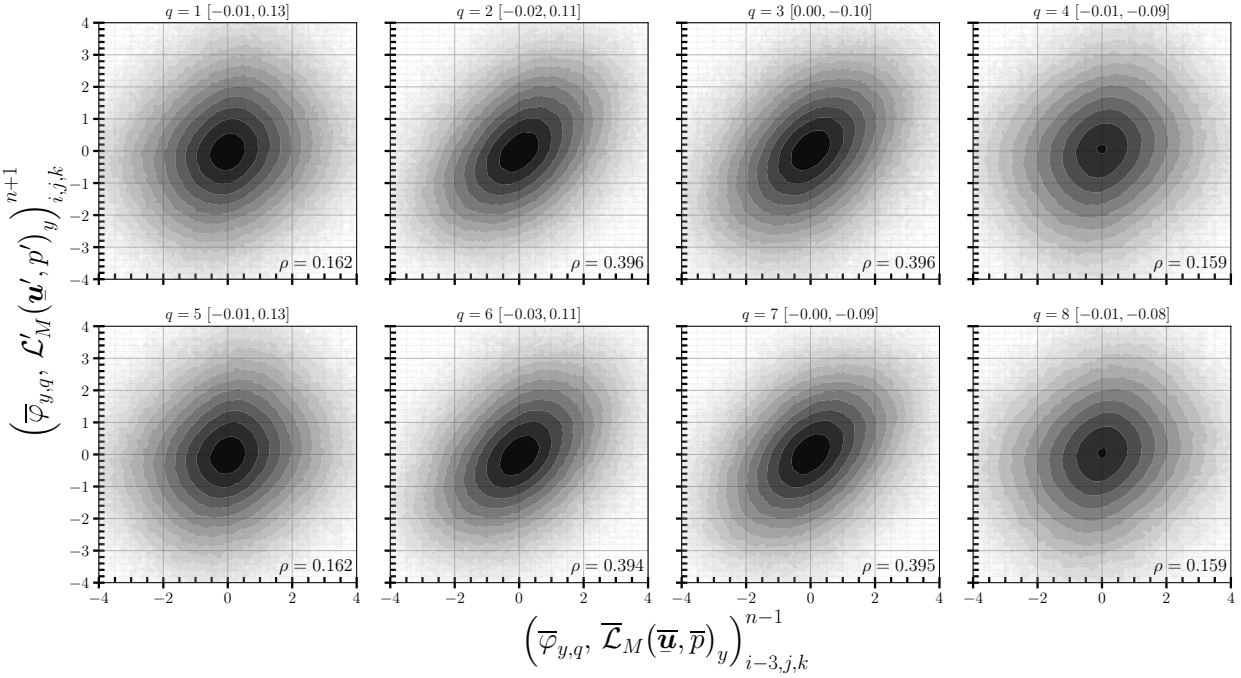


FIGURE A.265 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

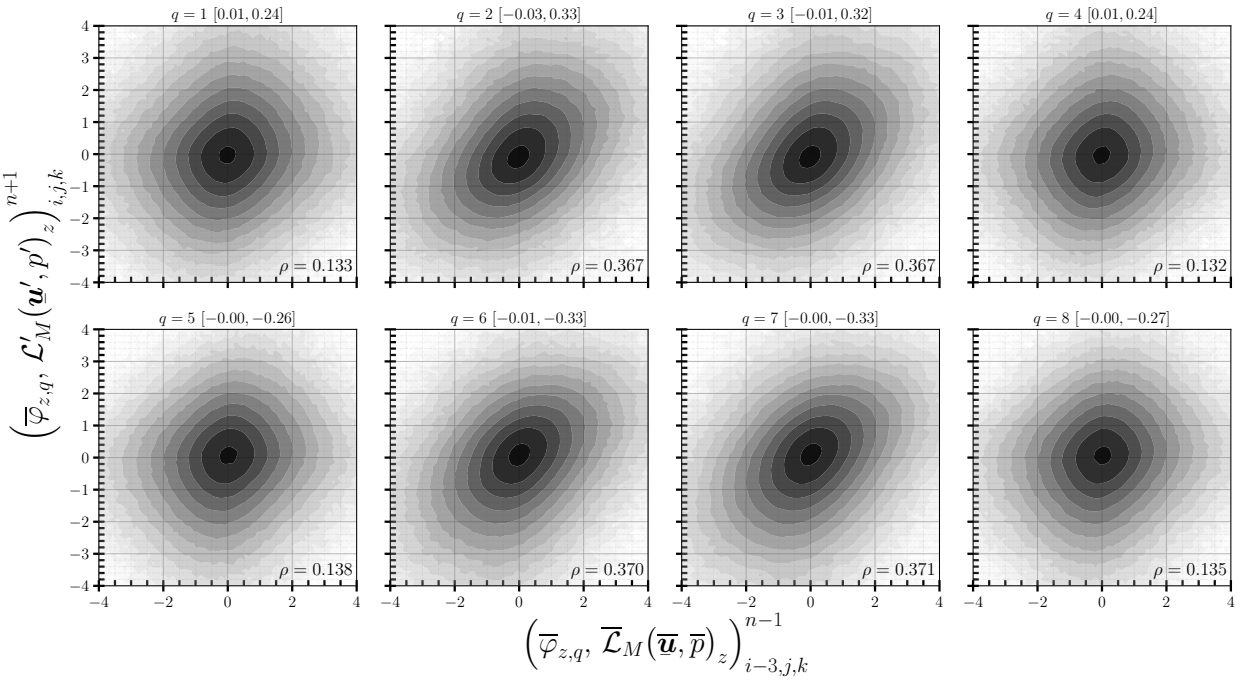


FIGURE A.266 Histograms of the t^{n-1} $(\mathrm{i}-3, \mathrm{j}, \mathrm{k})$-element z-momentum coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

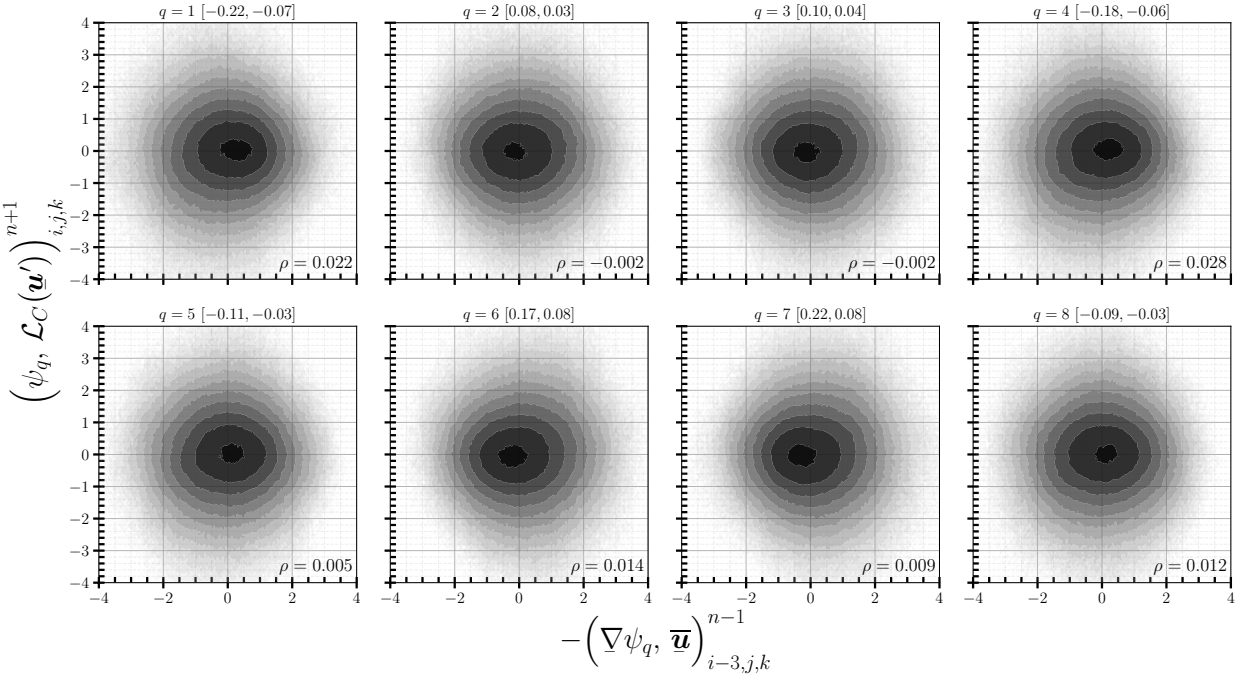


FIGURE A.267 Histograms of the t^{n-1} $(\mathrm{i}-3, \mathrm{j}, \mathrm{k})$-element continuity coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

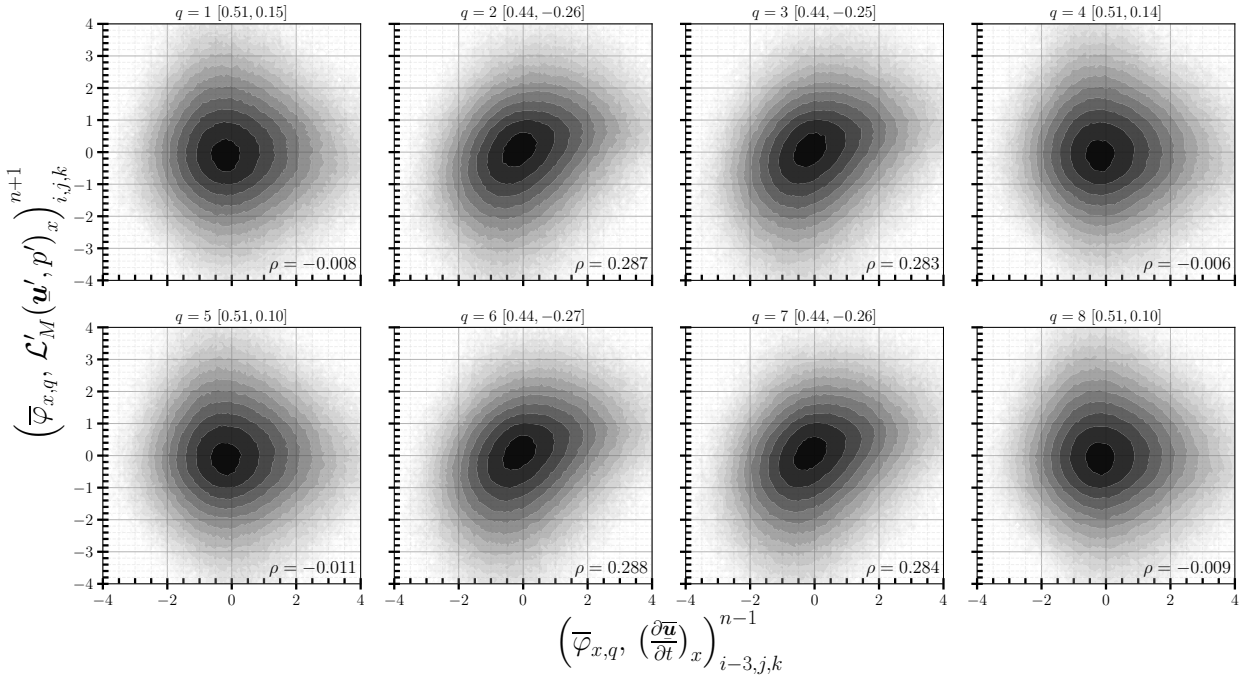


FIGURE A.268 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

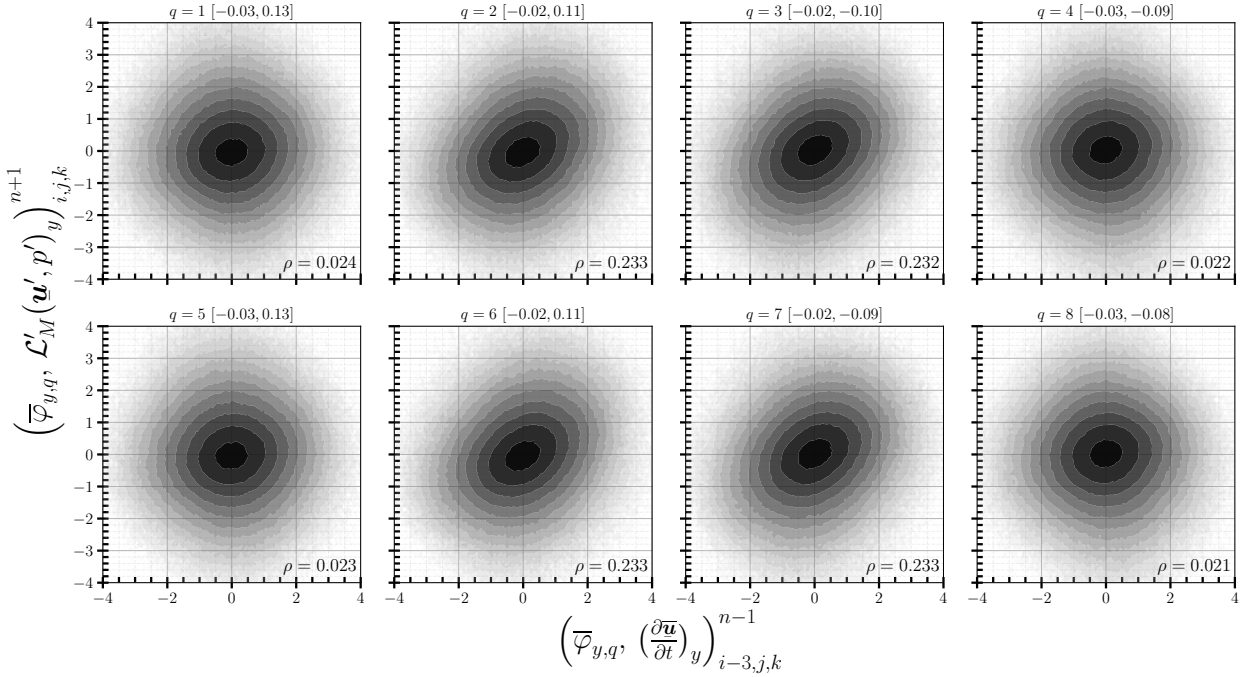


FIGURE A.269 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

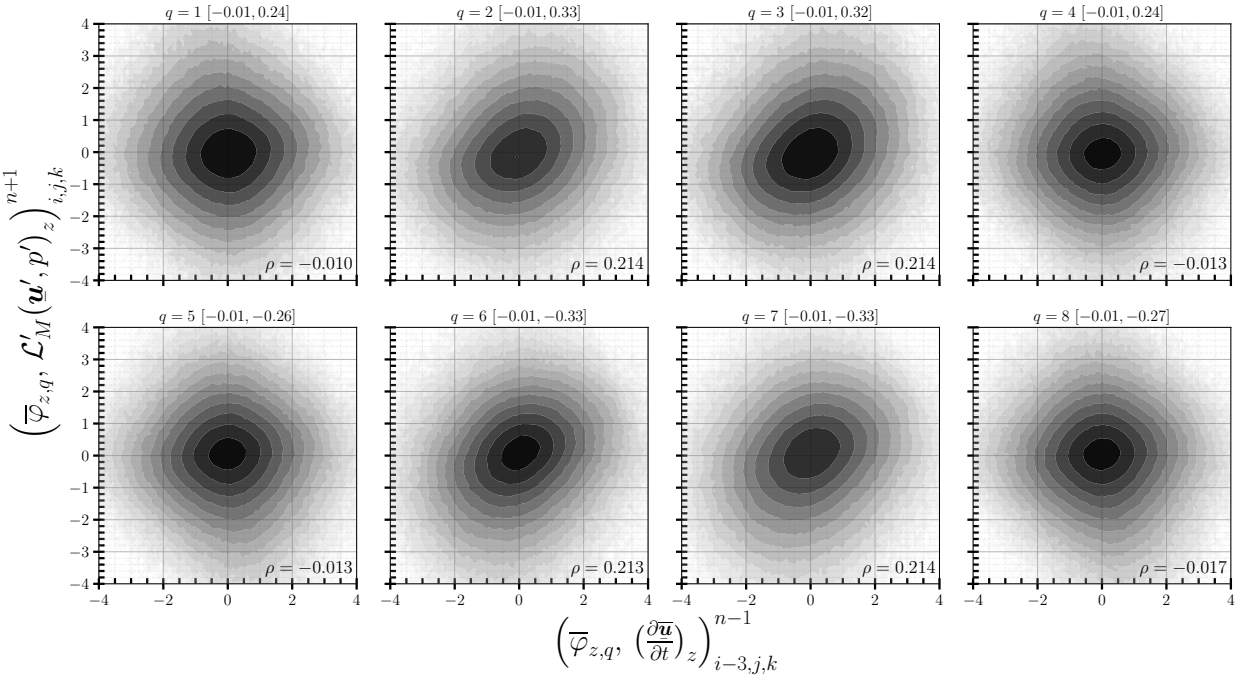


FIGURE A.270 Histograms of the t^{n-1} $(i-3, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

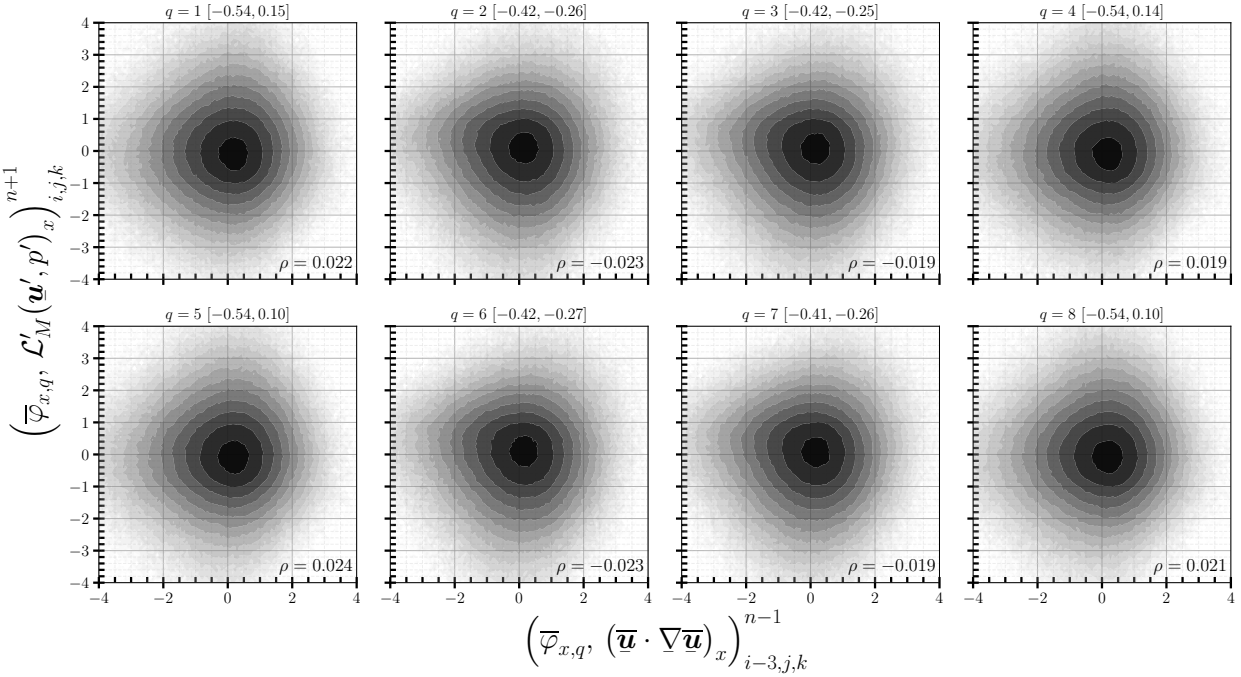


FIGURE A.271 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

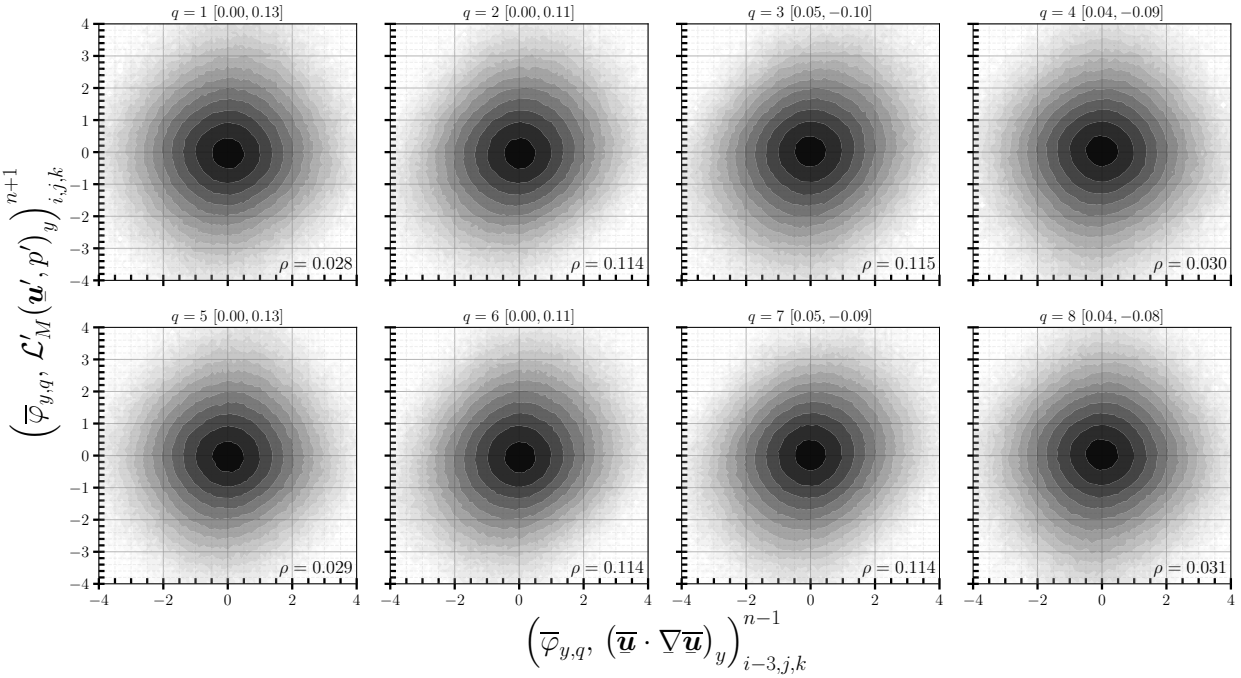


FIGURE A.272 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

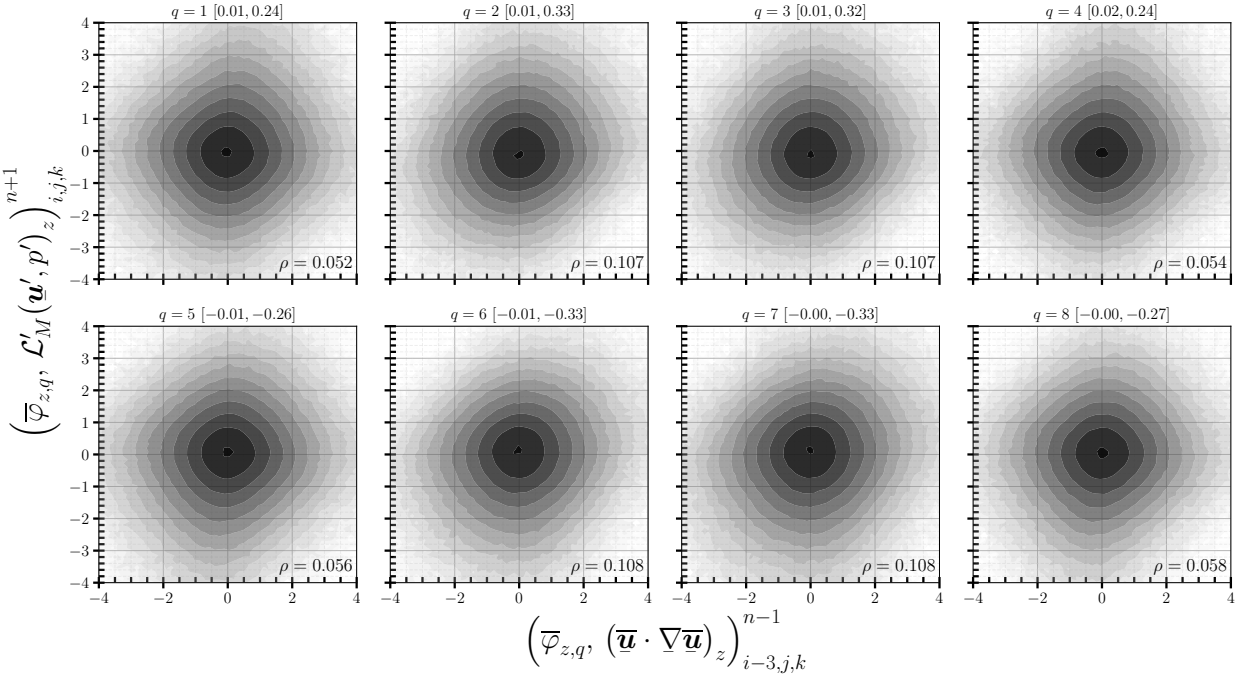


FIGURE A.273 Histograms of the t^{n-1} $(i-3, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

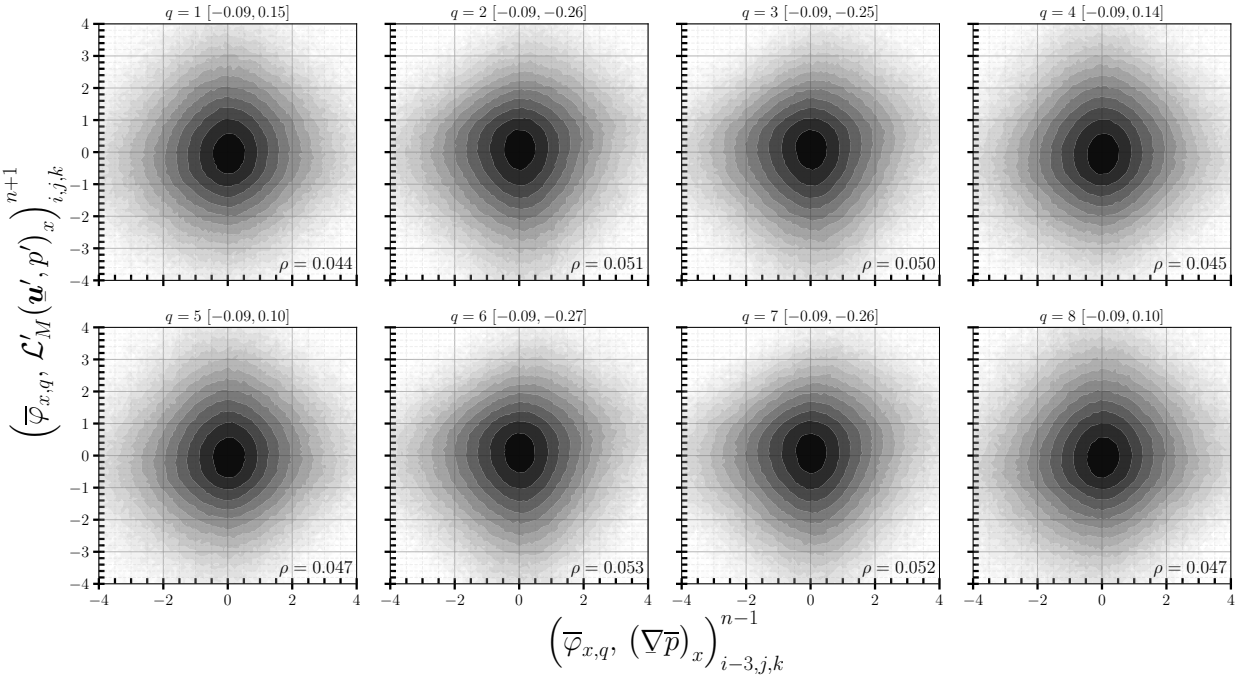


FIGURE A.274 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

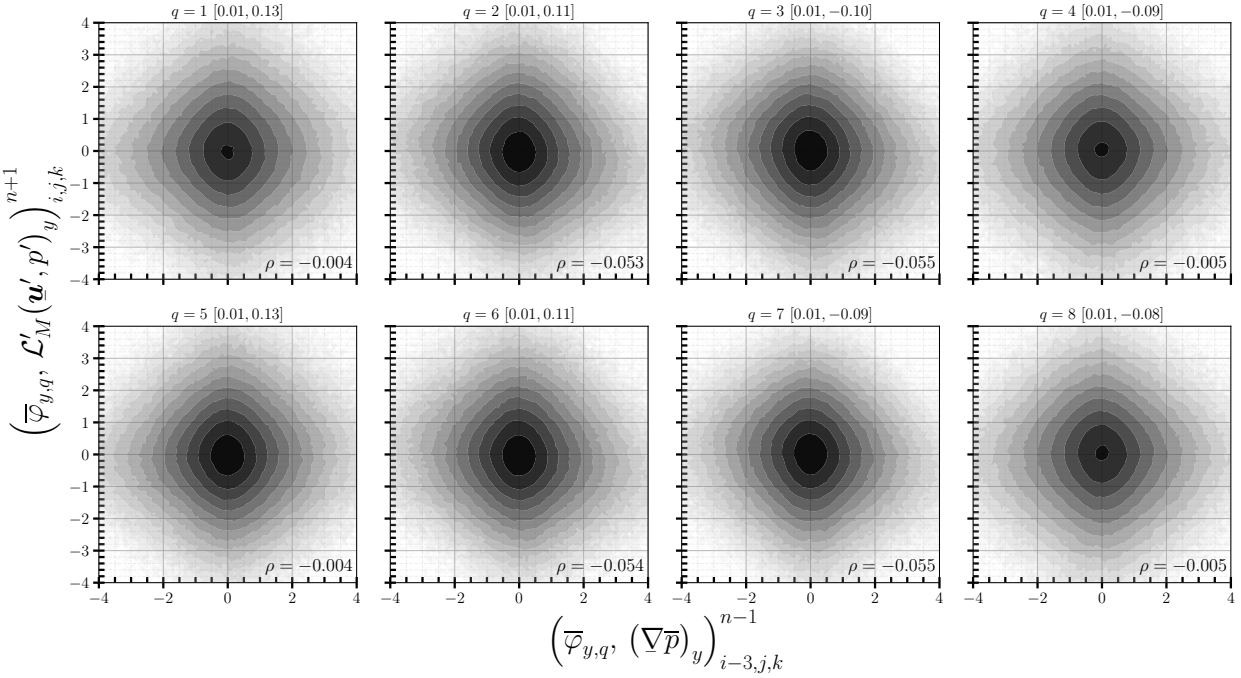


FIGURE A.275 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

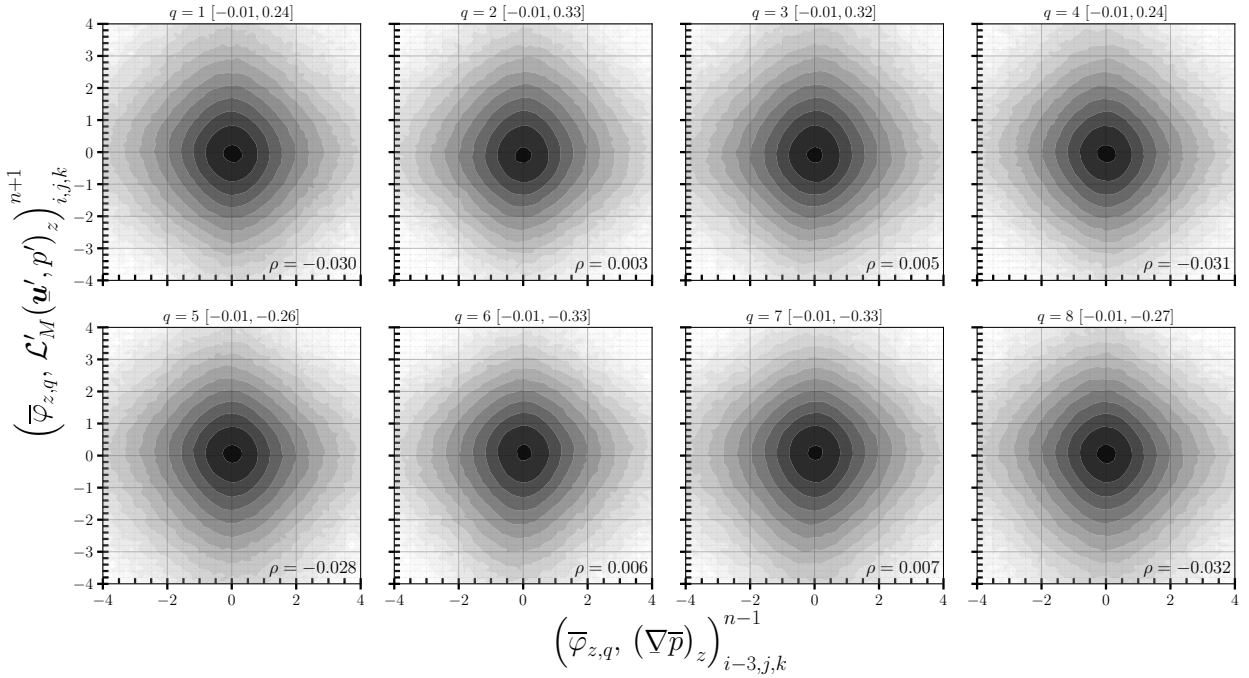


FIGURE A.276 Histograms of the t^{n-1} $(i-3, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

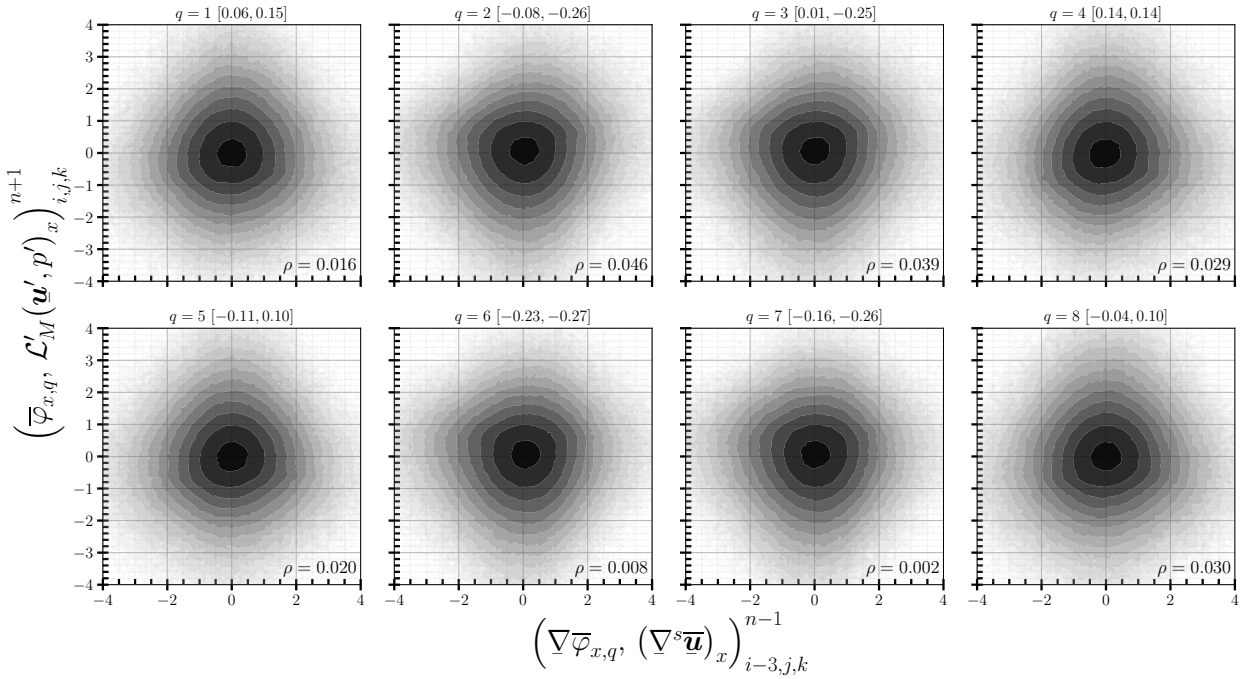


FIGURE A.277 Histograms of the t^{n-1} $(i-3, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

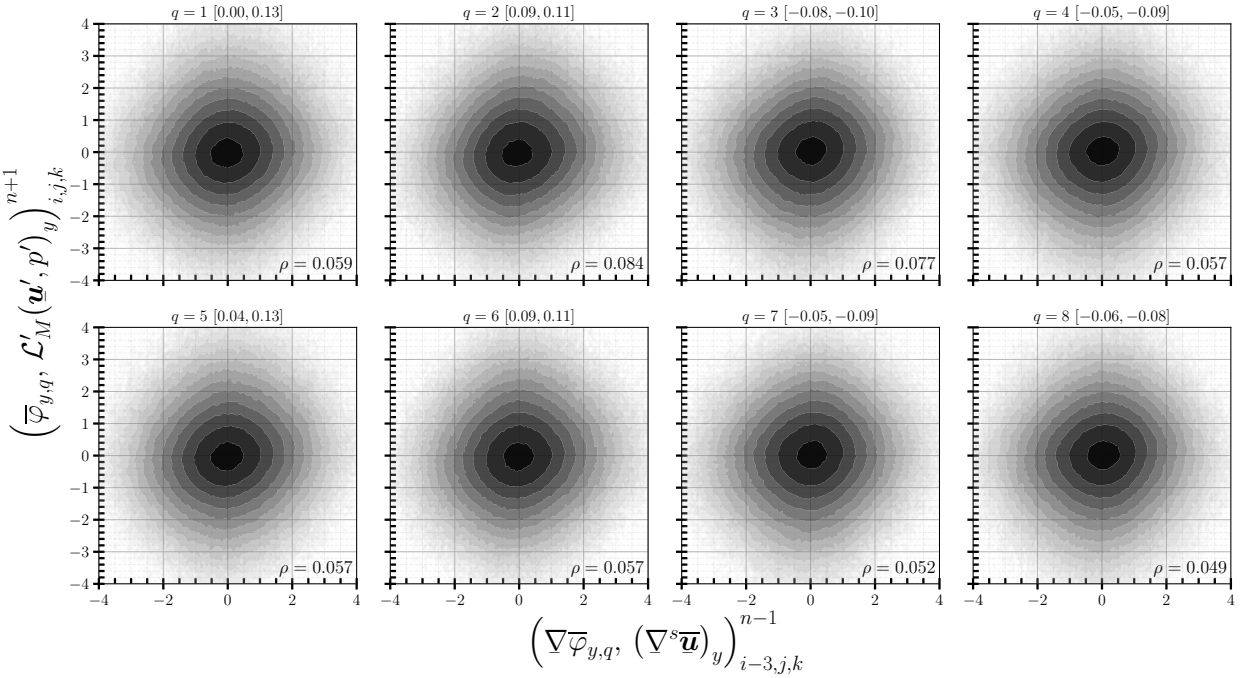


FIGURE A.278 Histograms of the t^{n-1} $(i-3, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

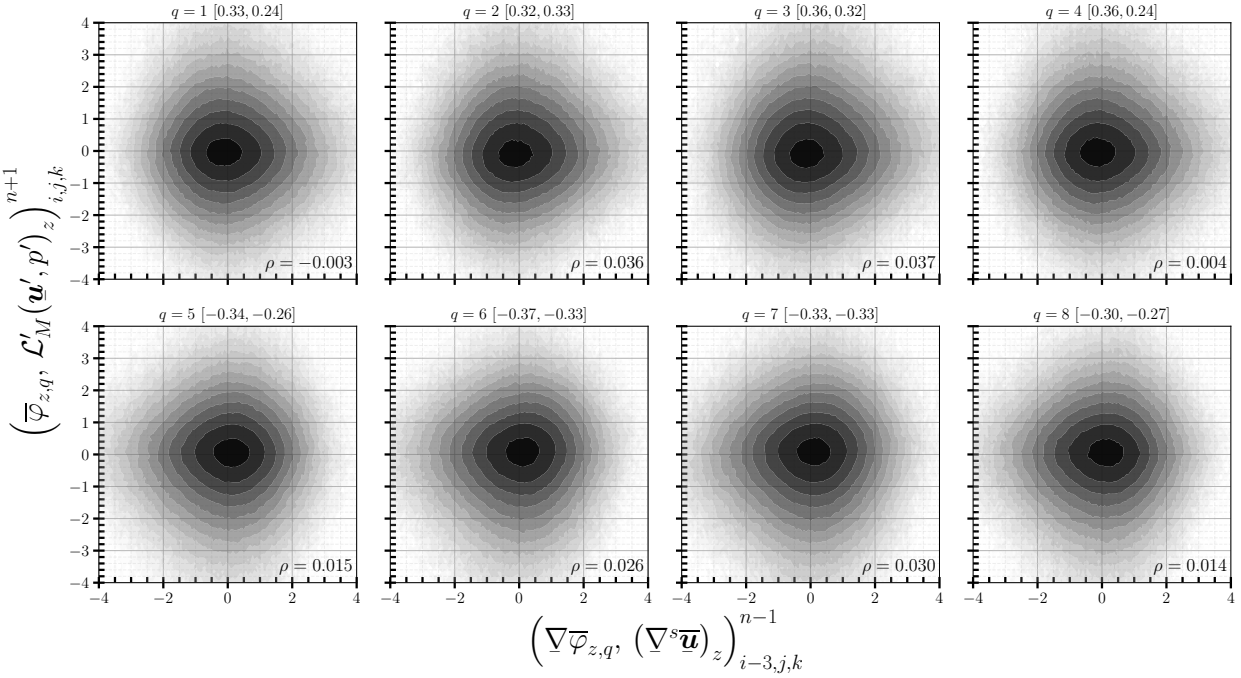


FIGURE A.279 Histograms of the t^{n-1} $(i-3, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.5 $(i-4, j, k)$ Histograms

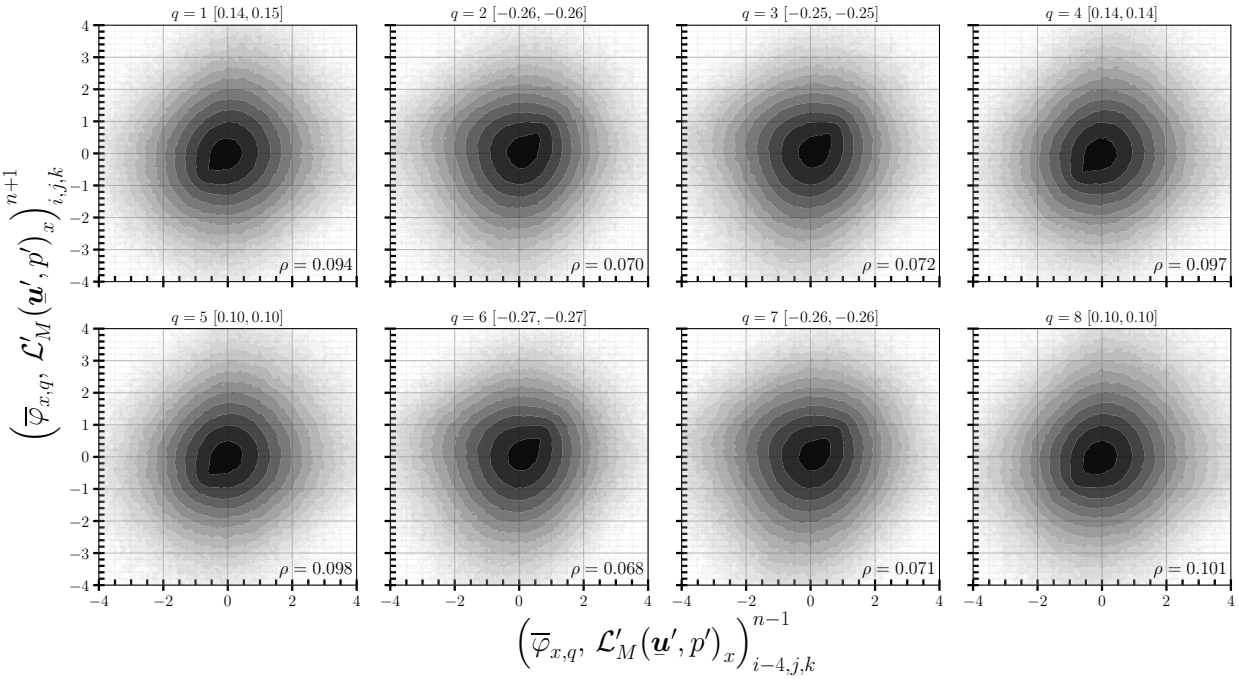


FIGURE A.280 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

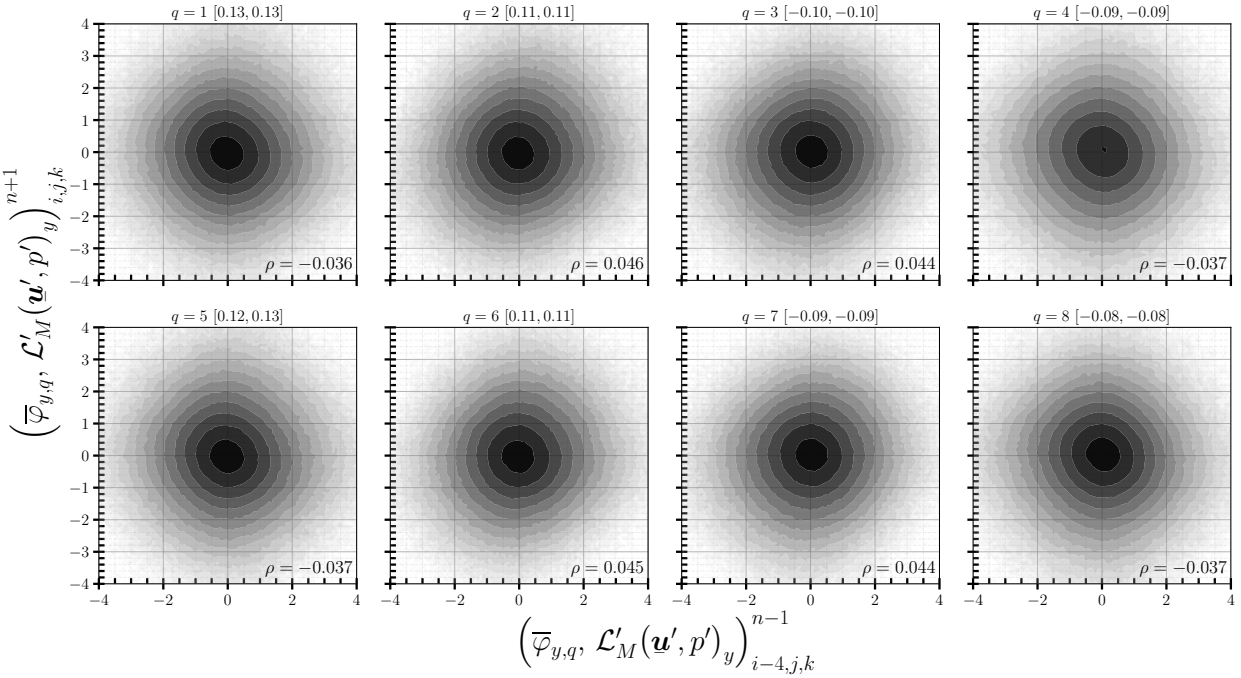


FIGURE A.281 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

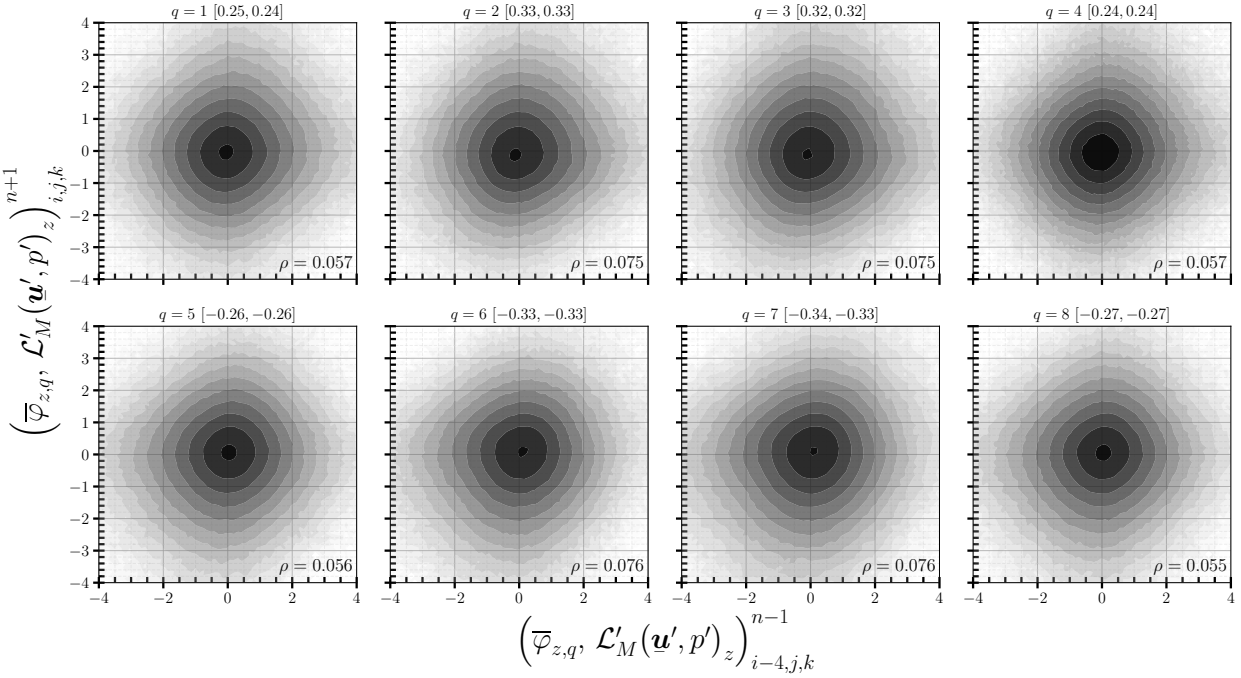


FIGURE A.282 Histograms of the t^{n-1} $(i-4, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

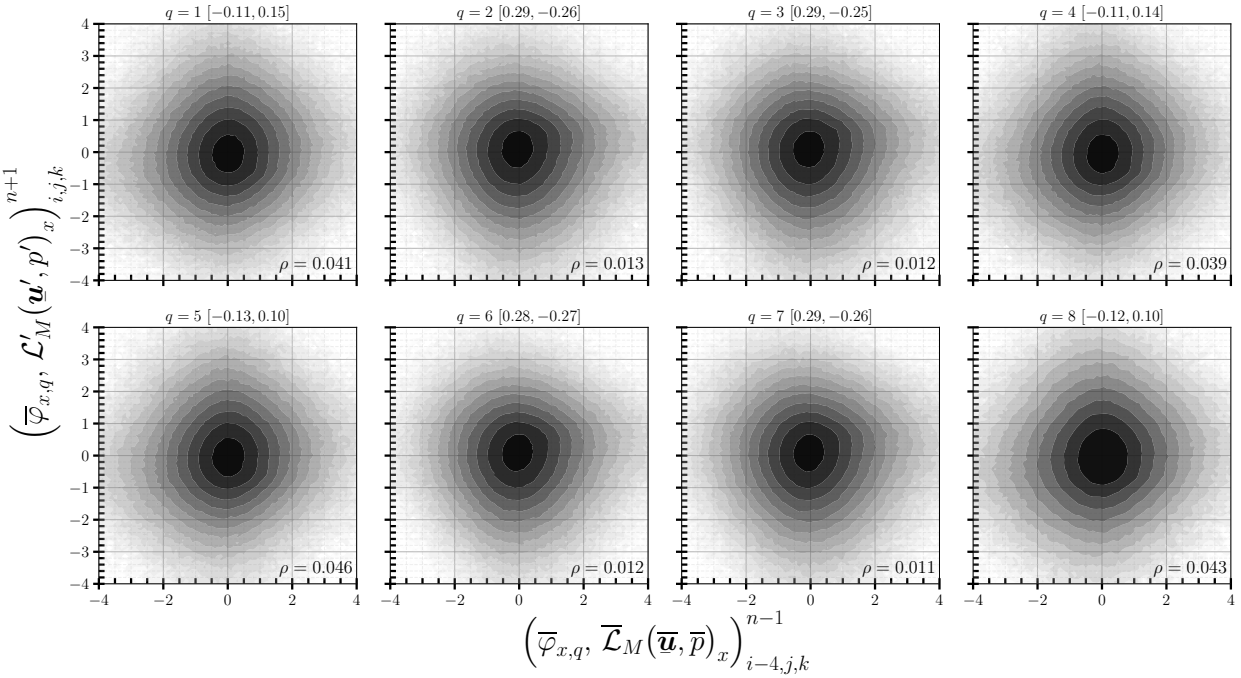


FIGURE A.283 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

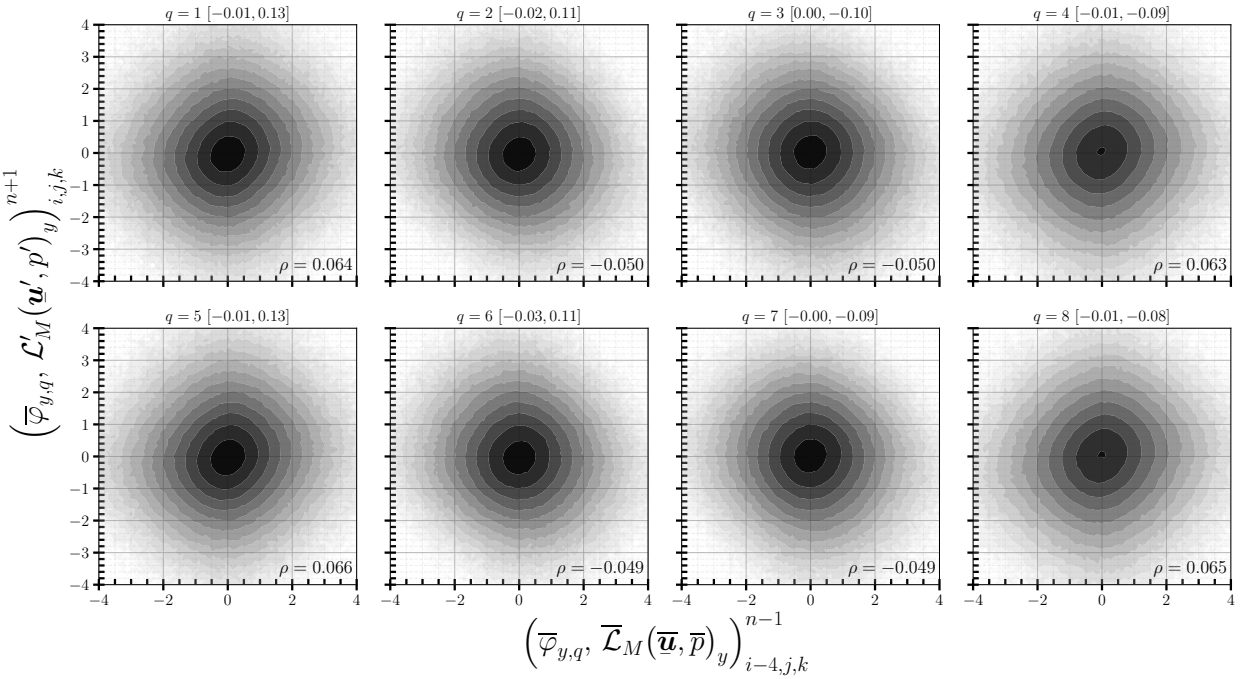


FIGURE A.284 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

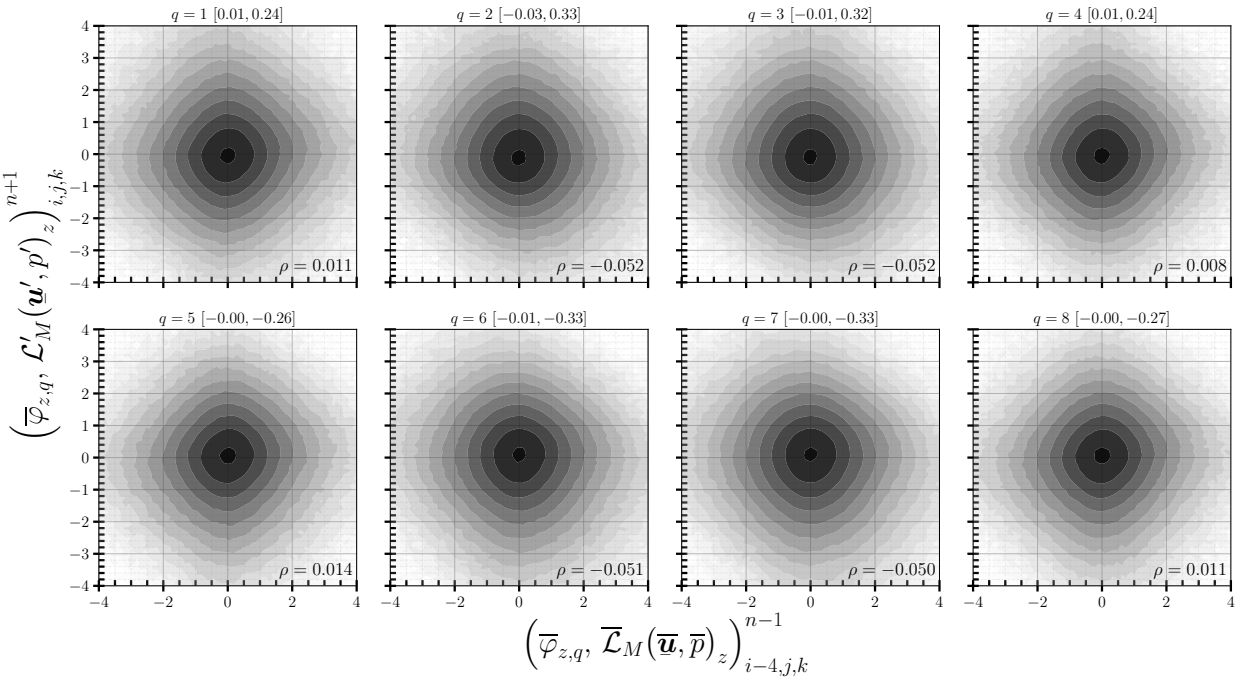


FIGURE A.285 Histograms of the t^{n-1} $(\mathrm{i}-4, \mathrm{j}, \mathrm{k})$-element z-momentum coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

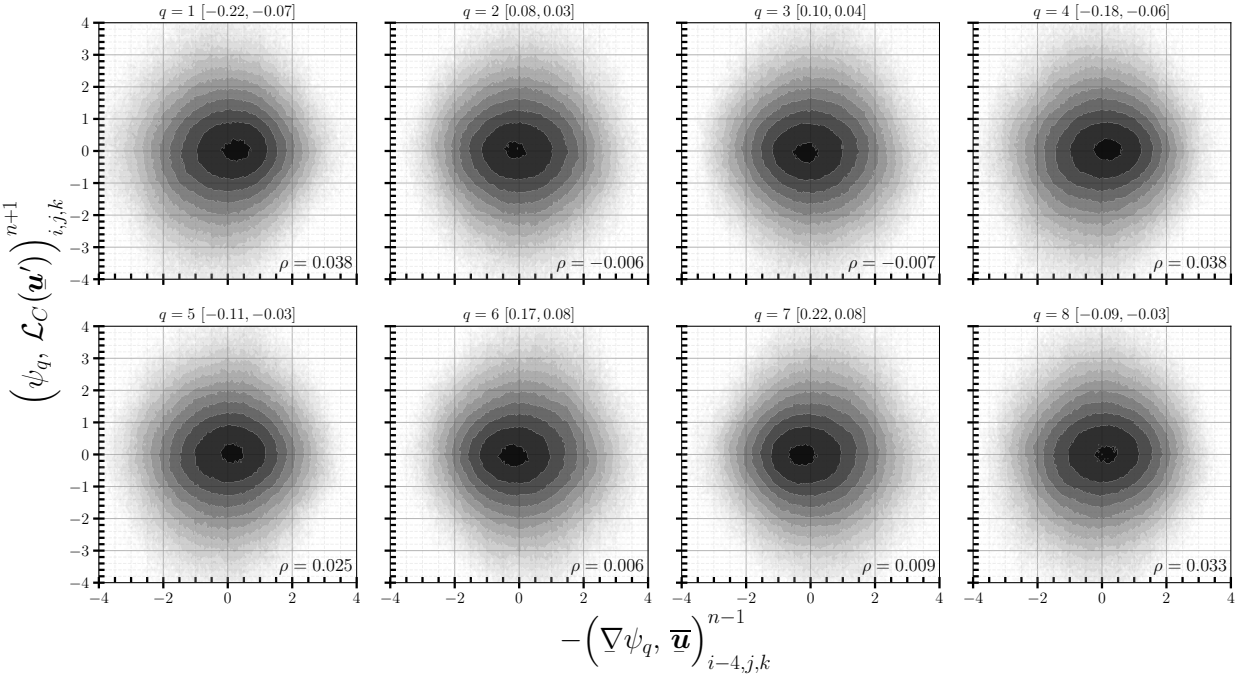


FIGURE A.286 Histograms of the t^{n-1} $(\mathrm{i}-4, \mathrm{j}, \mathrm{k})$-element continuity coarse-scale sum against the t^{n+1} $(\mathrm{i}, \mathrm{j}, \mathrm{k})$-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

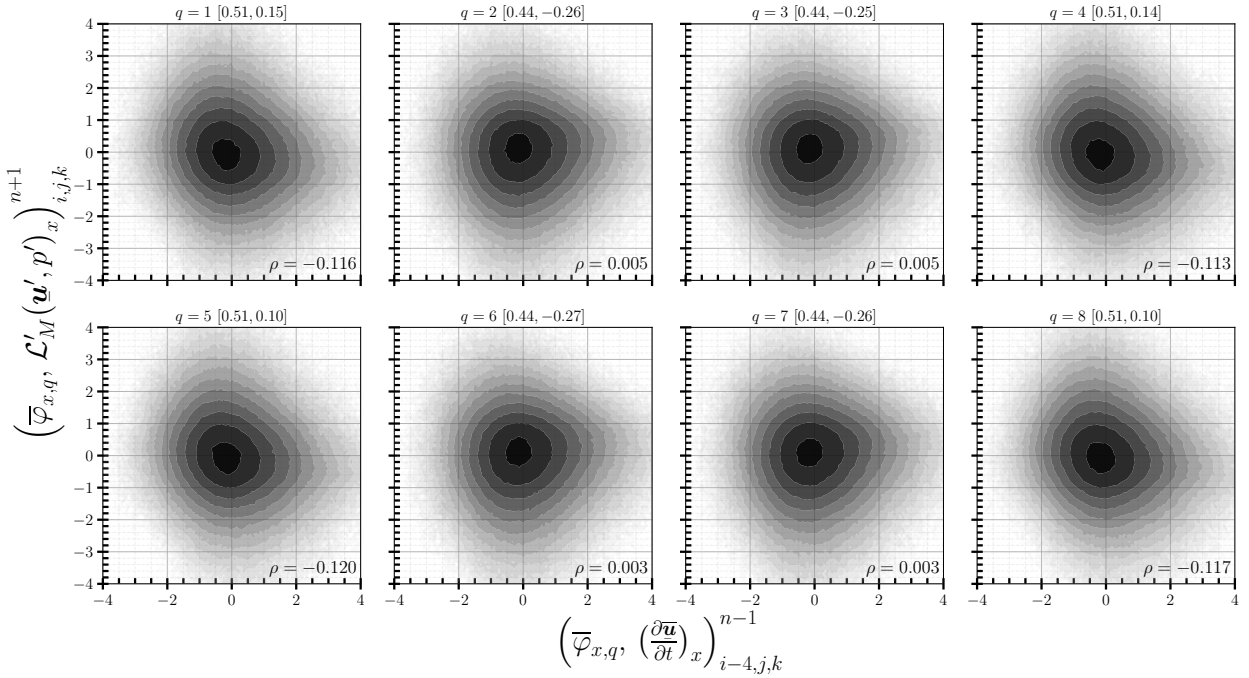


FIGURE A.287 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

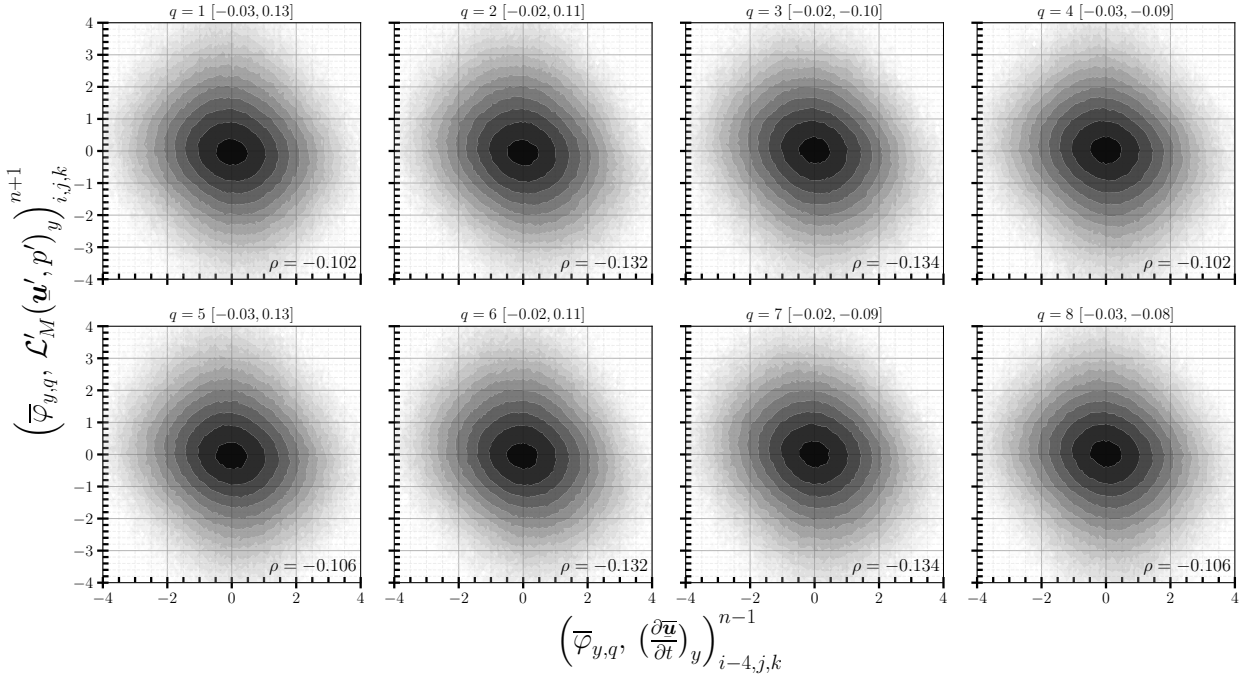


FIGURE A.288 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

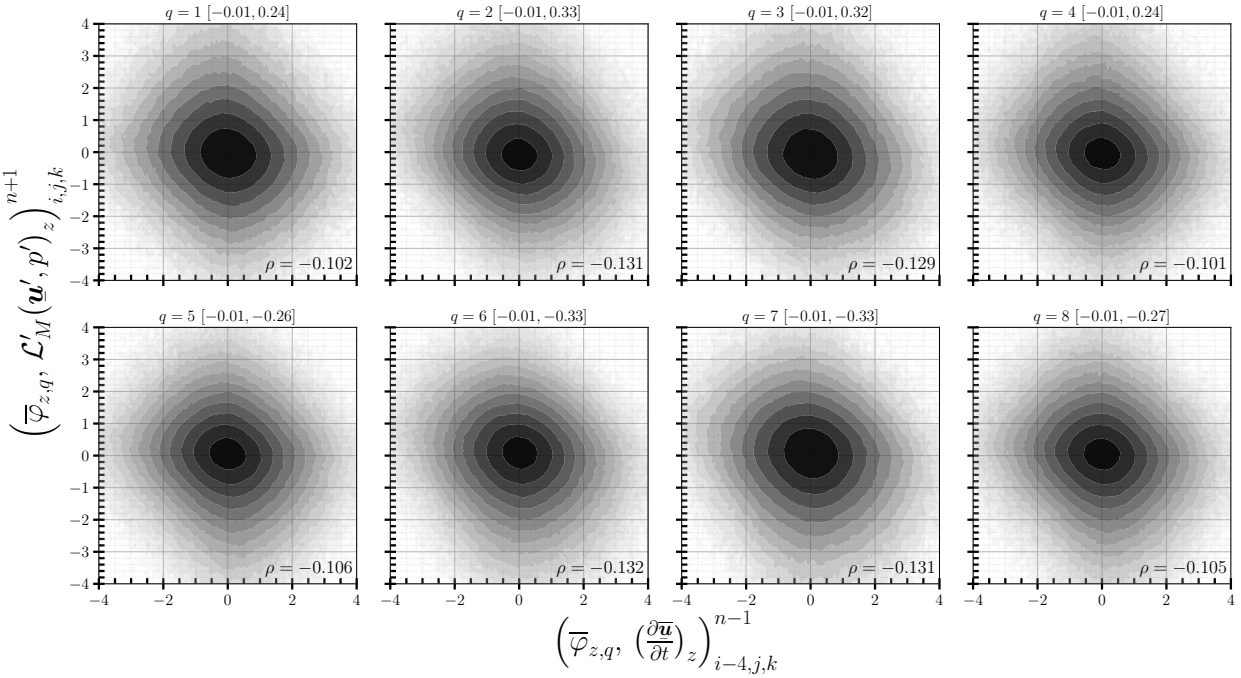


FIGURE A.289 Histograms of the t^{n-1} $(i-4, j, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

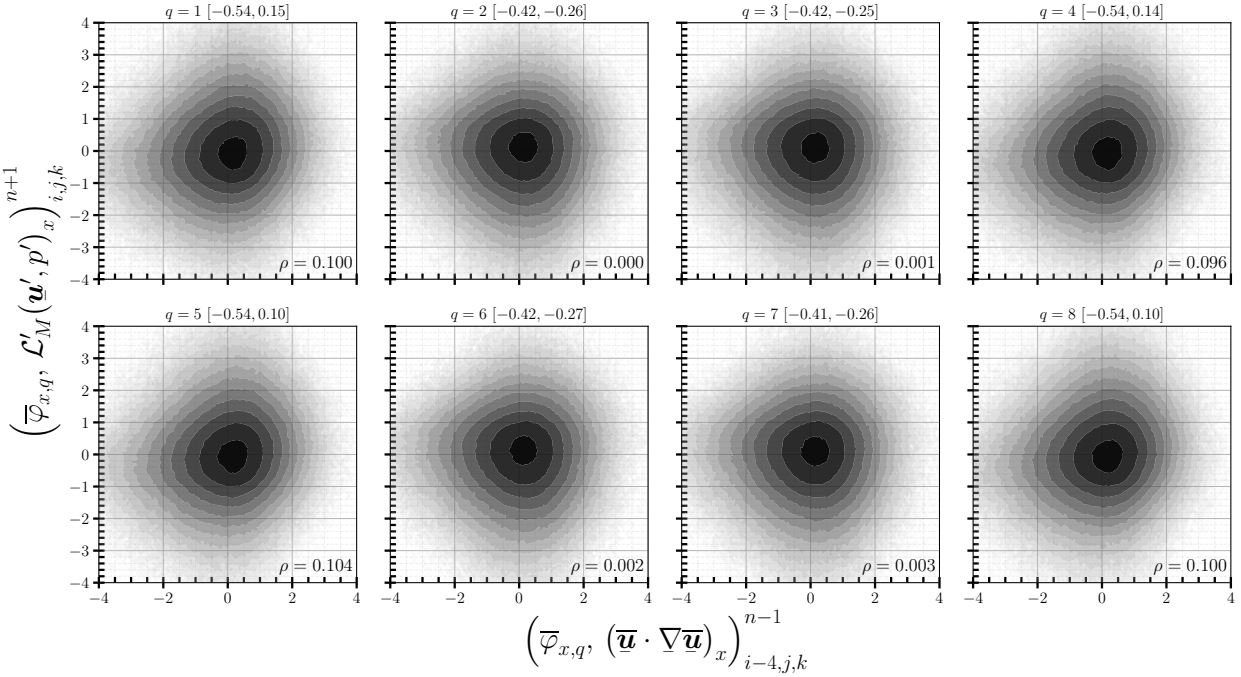


FIGURE A.290 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

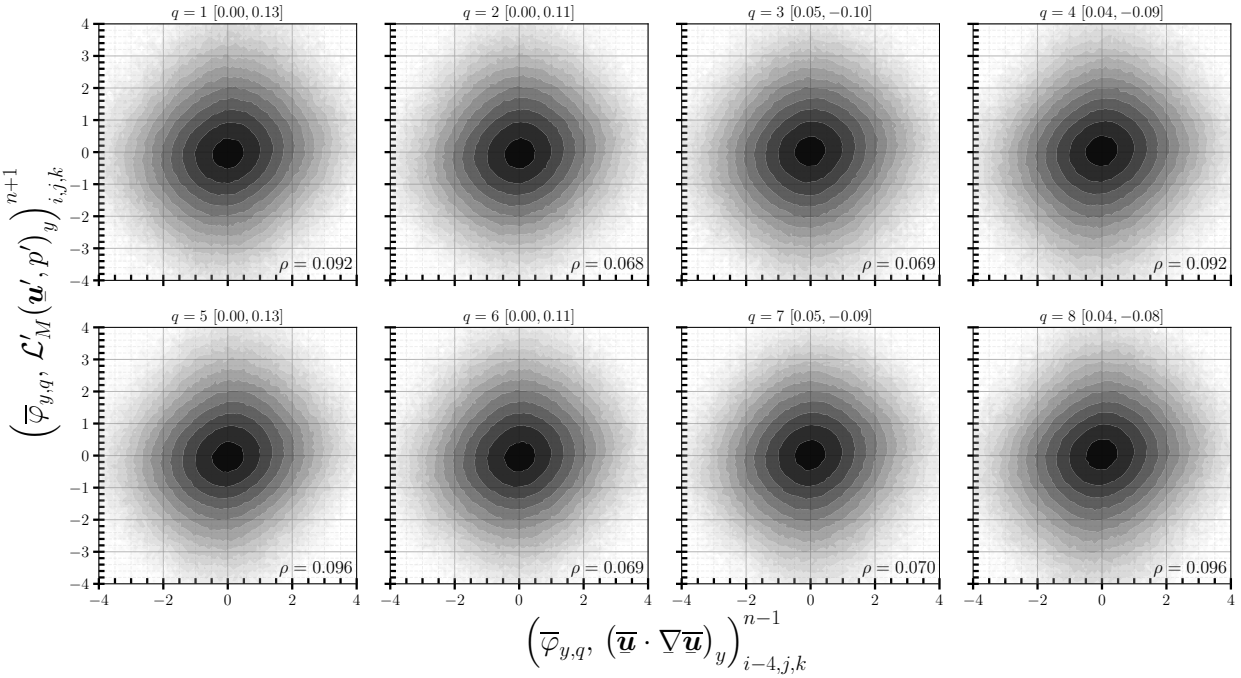


FIGURE A.291 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

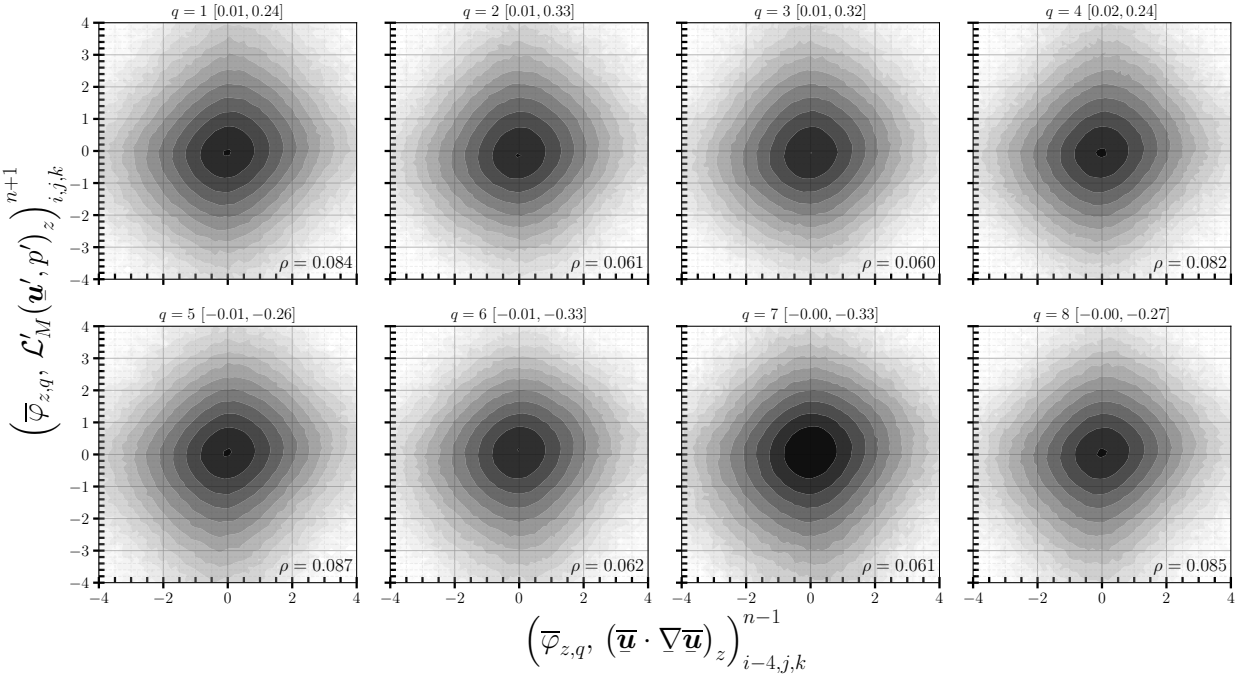


FIGURE A.292 Histograms of the t^{n-1} $(i-4, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

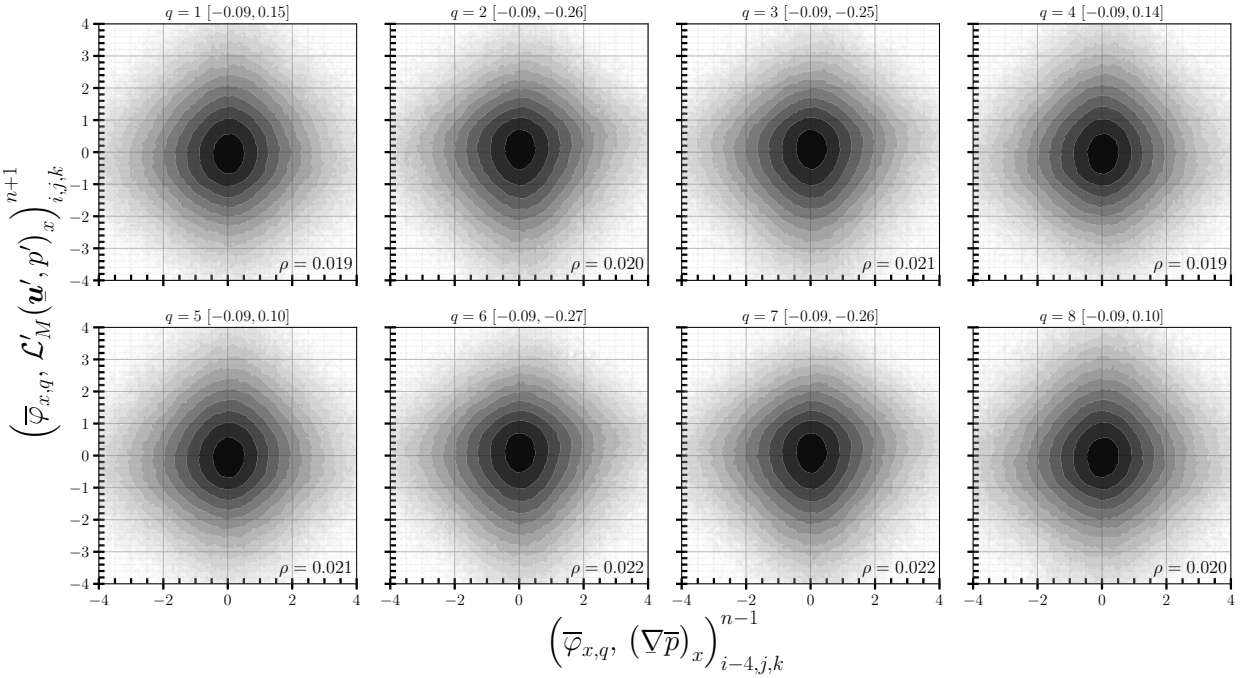


FIGURE A.293 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

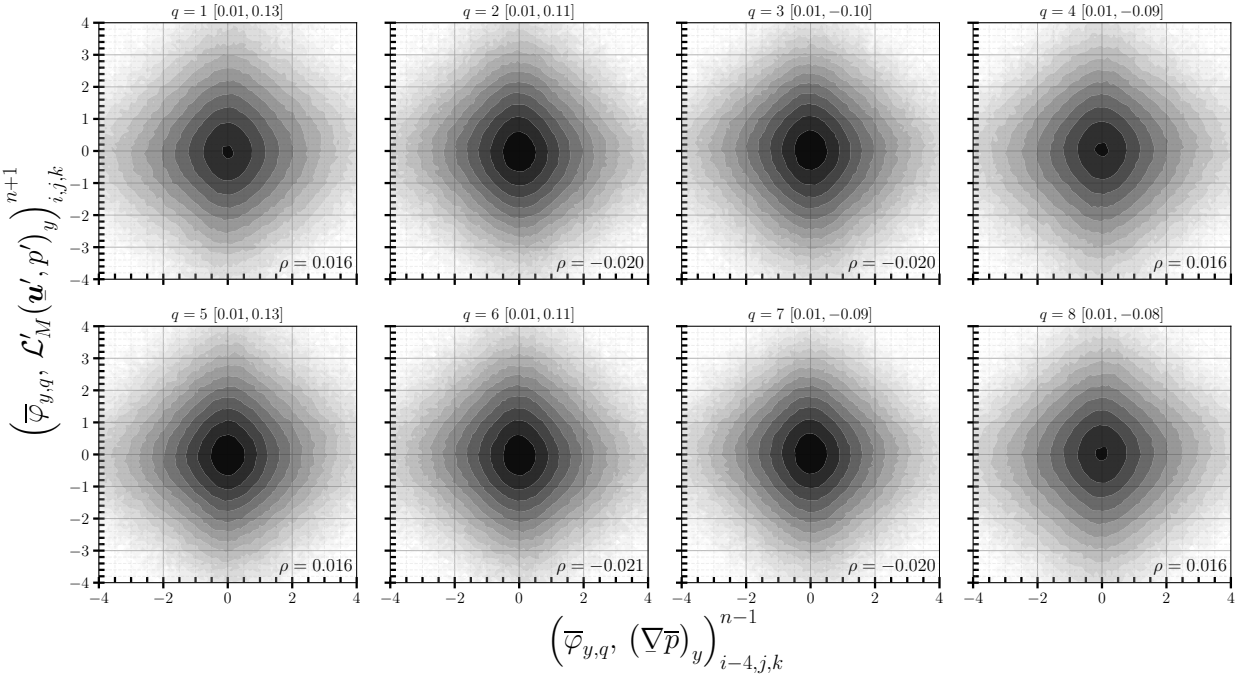


FIGURE A.294 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

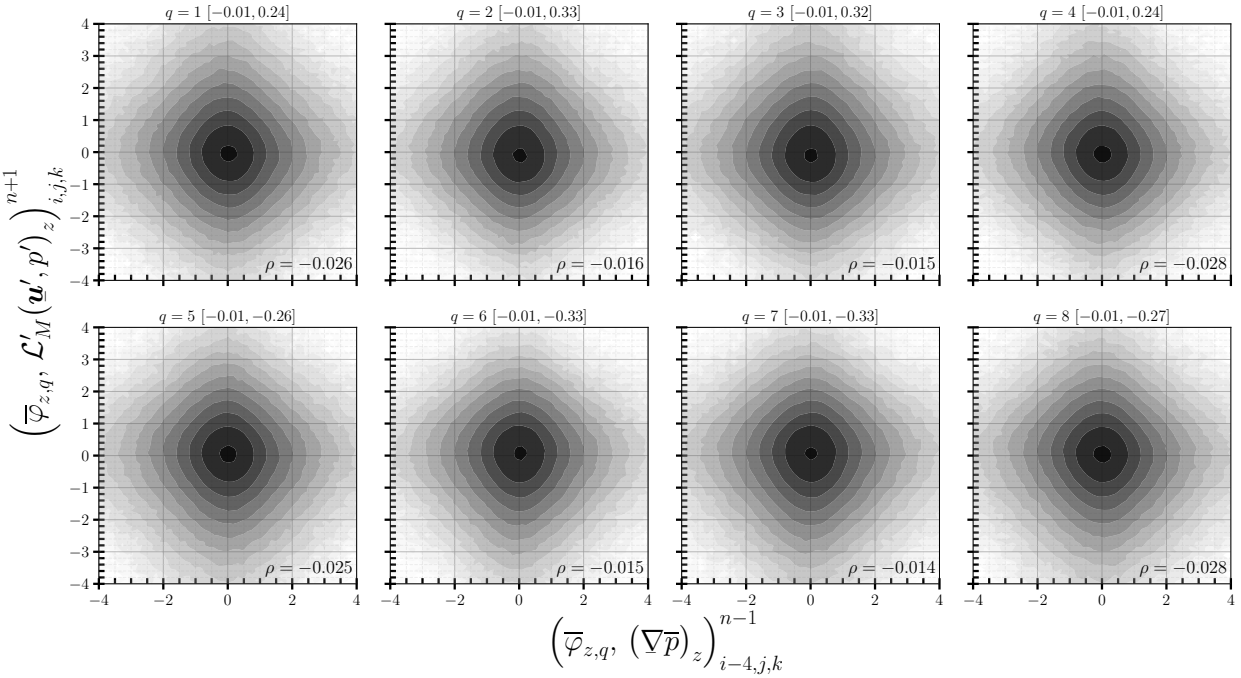


FIGURE A.295 Histograms of the t^{n-1} $(i-4, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

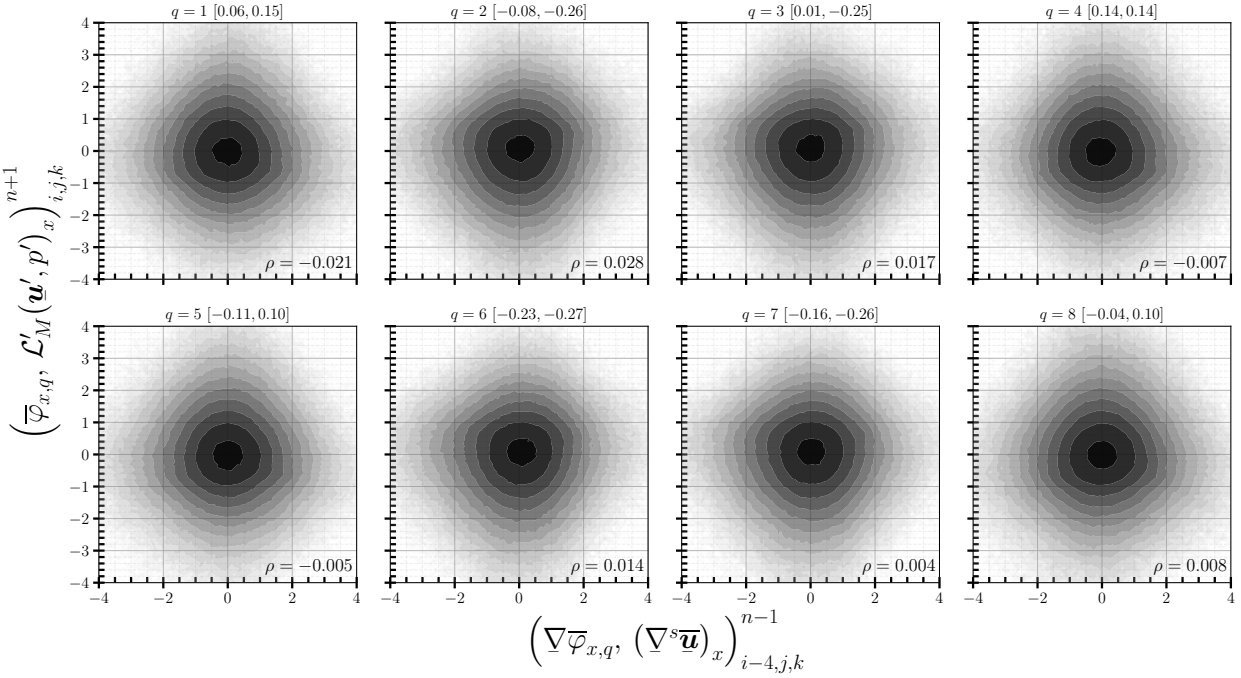


FIGURE A.296 Histograms of the t^{n-1} $(i-4, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

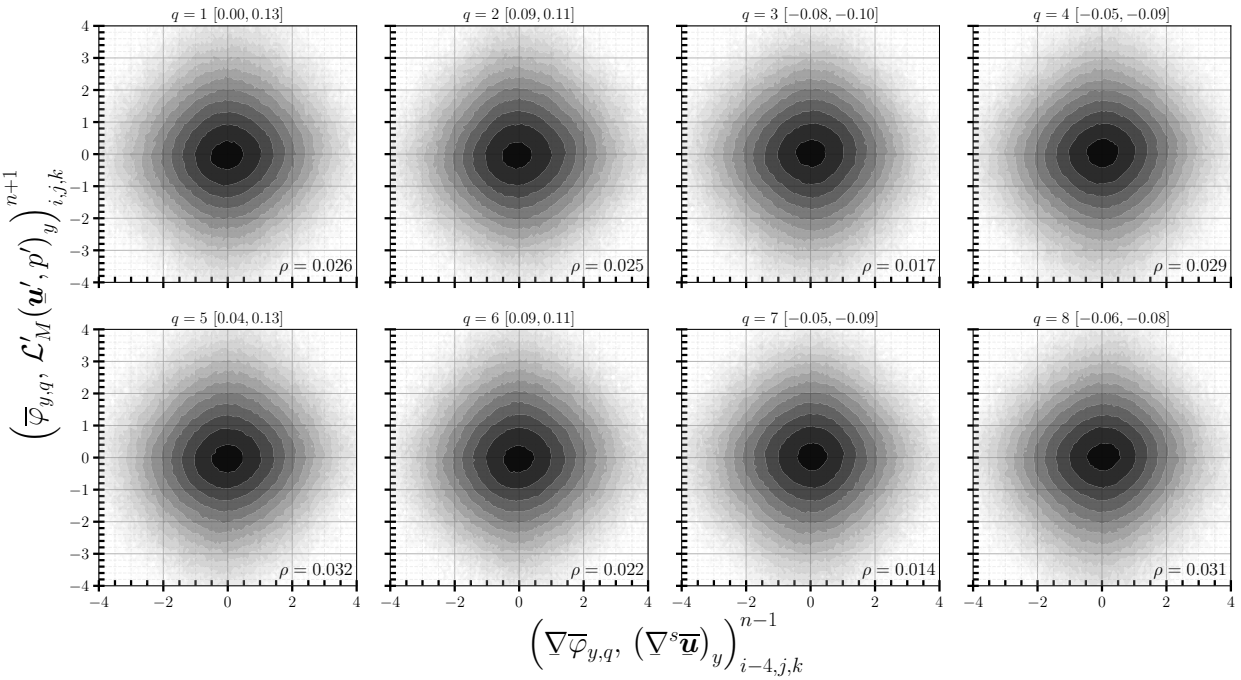


FIGURE A.297 Histograms of the t^{n-1} $(i-4, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

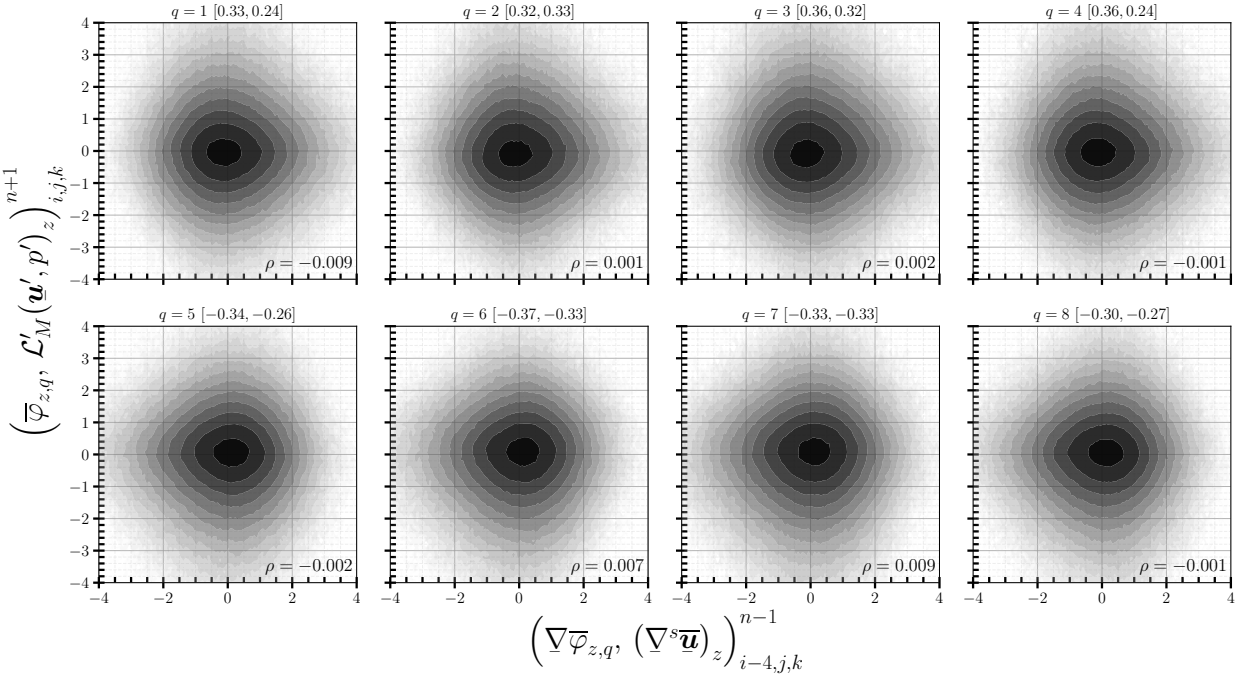


FIGURE A.298 Histograms of the t^{n-1} $(i-4, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.6 $(i+1, j, k)$ Histograms

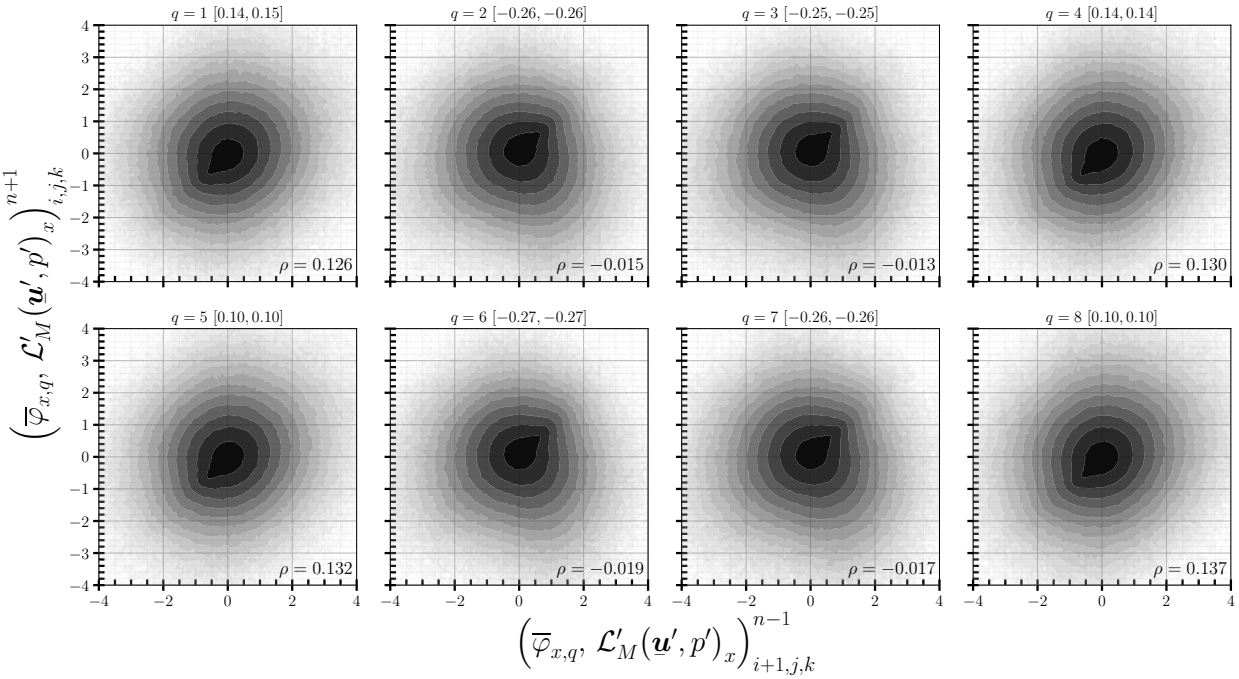


Figure A.299 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

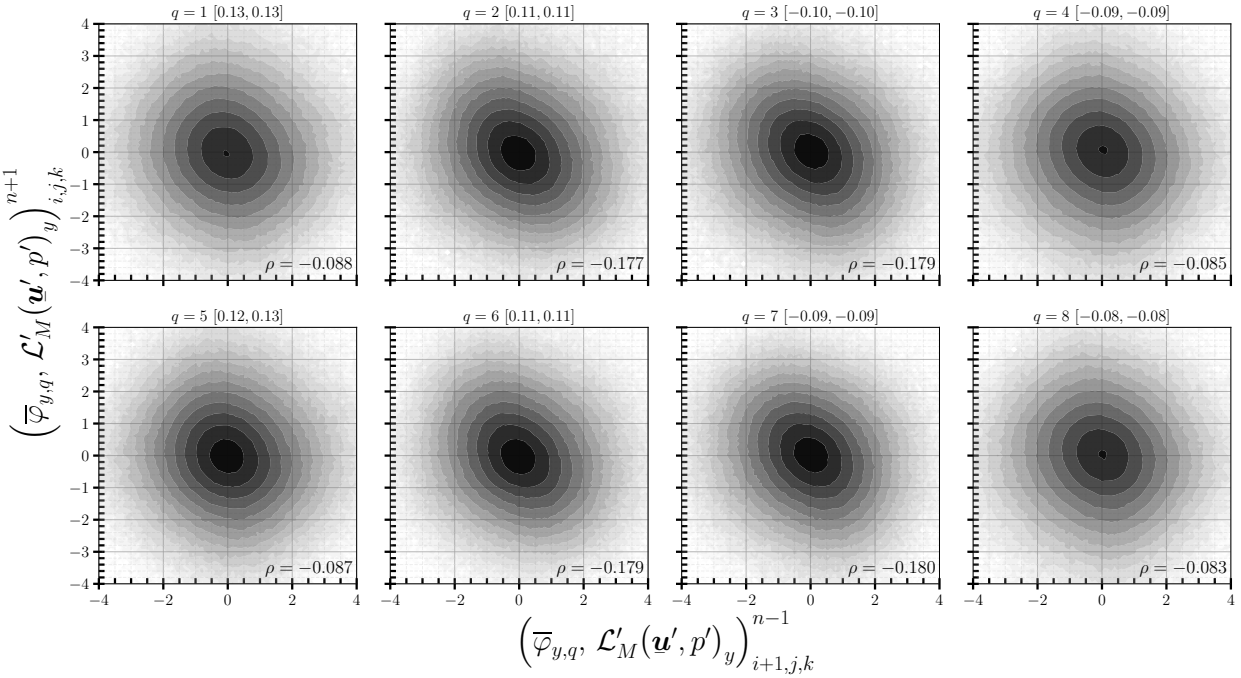


FIGURE A.300 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

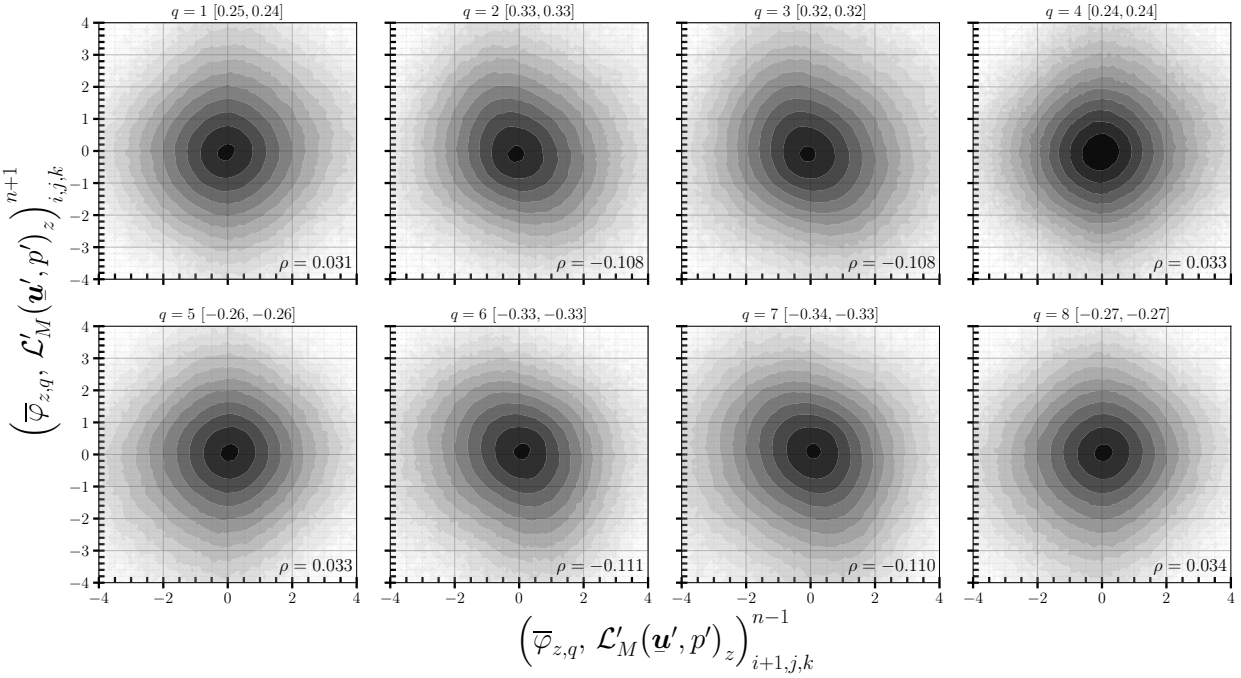


FIGURE A.301 Histograms of the t^{n-1} $(i+1, j, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

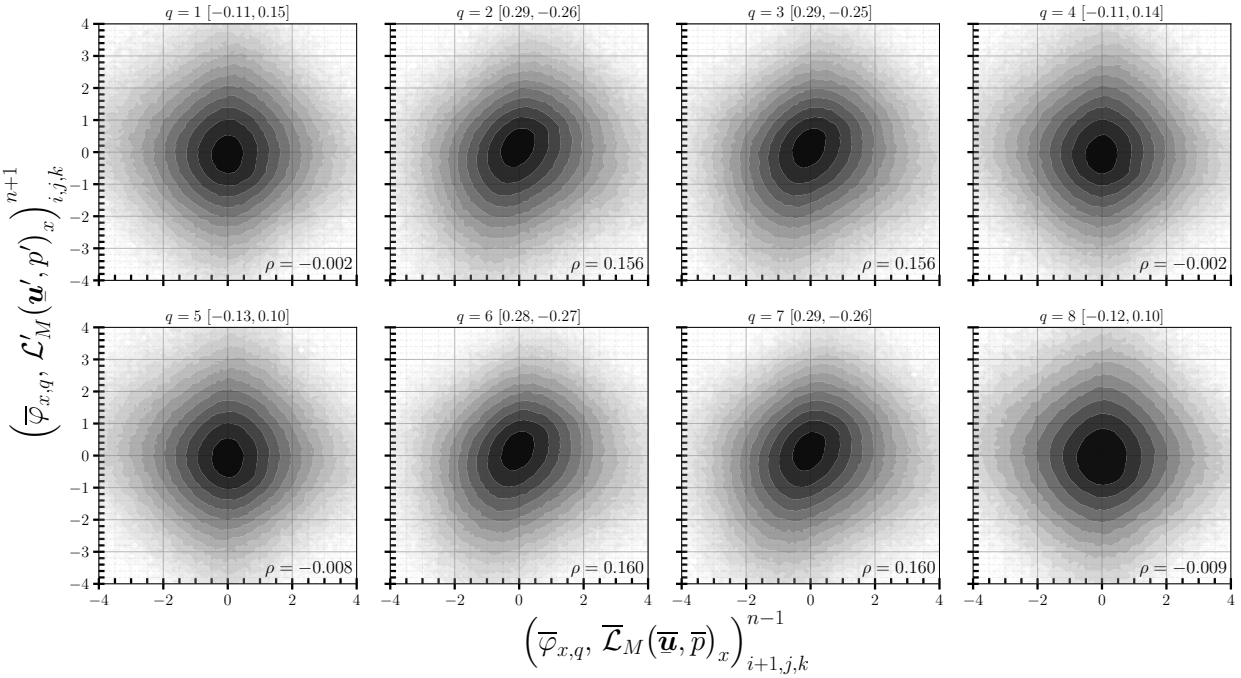


FIGURE A.302 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

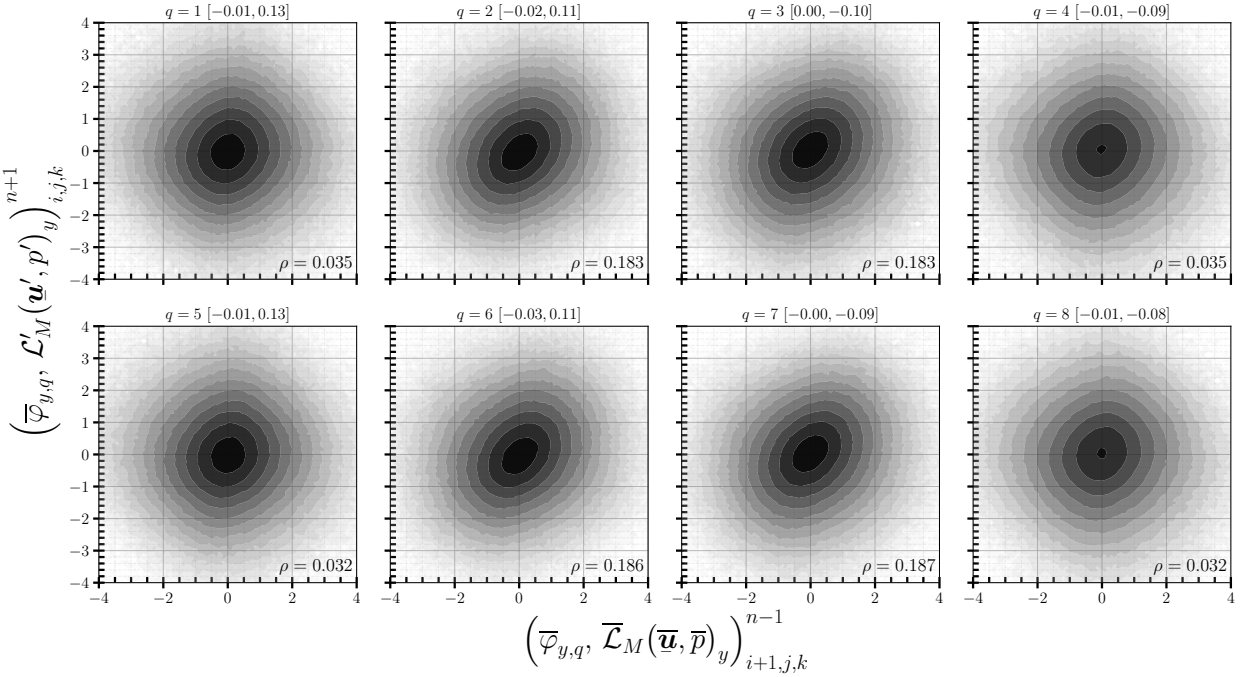


FIGURE A.303 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

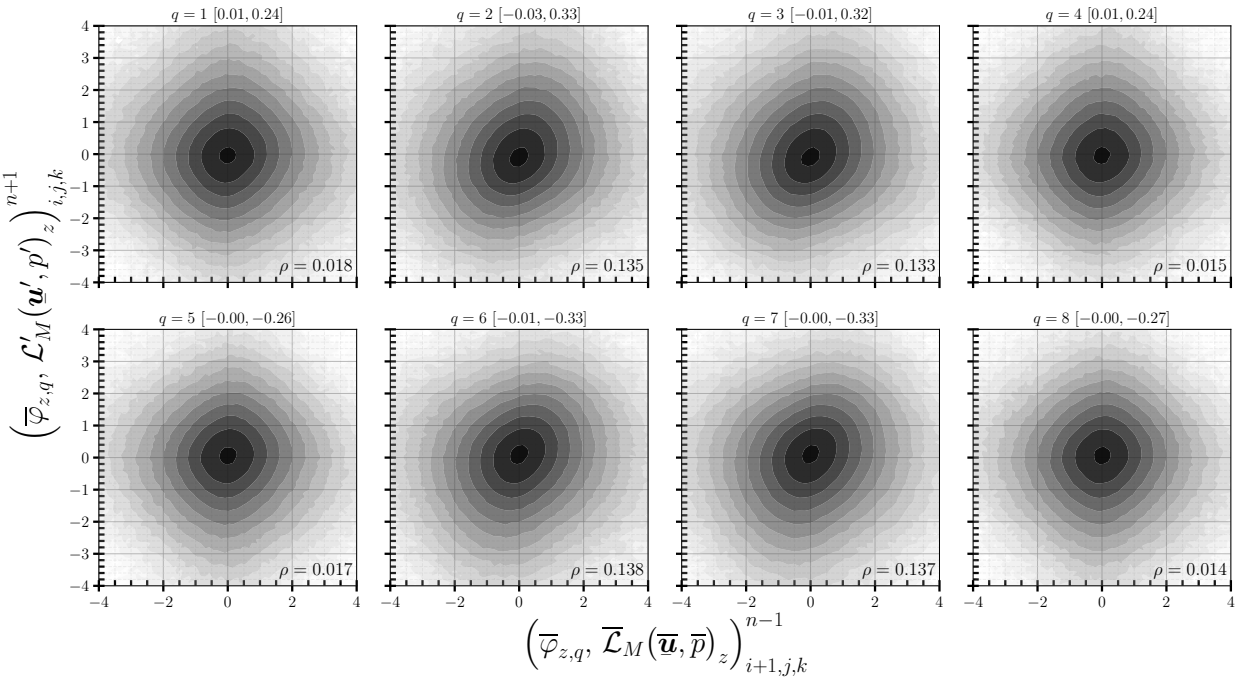


FIGURE A.304 Histograms of the t^{n-1} $(i+1, j, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

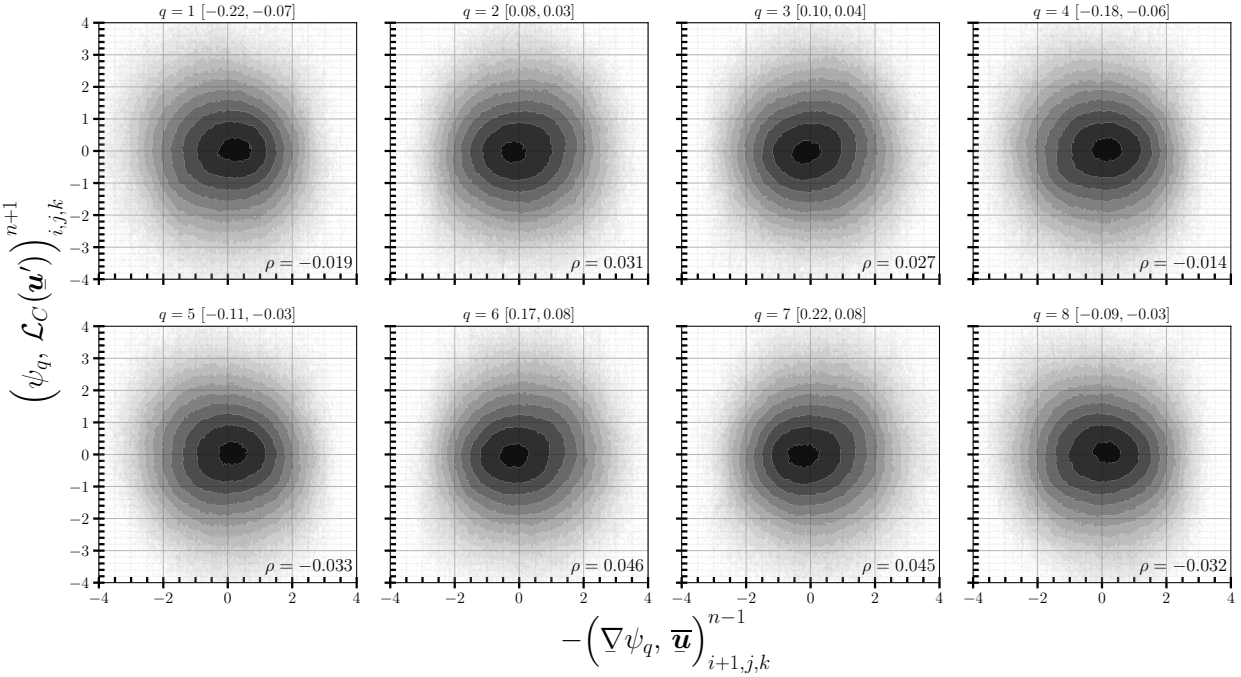


FIGURE A.305 Histograms of the t^{n-1} $(i+1, j, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

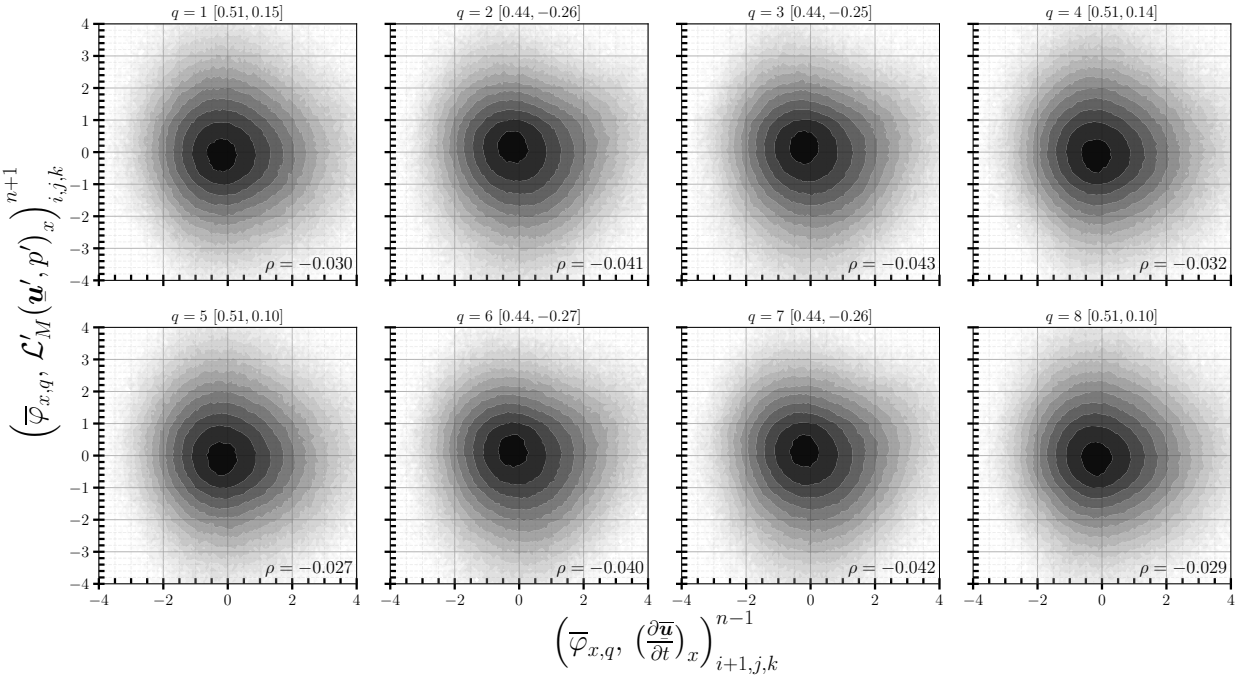


FIGURE A.306 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

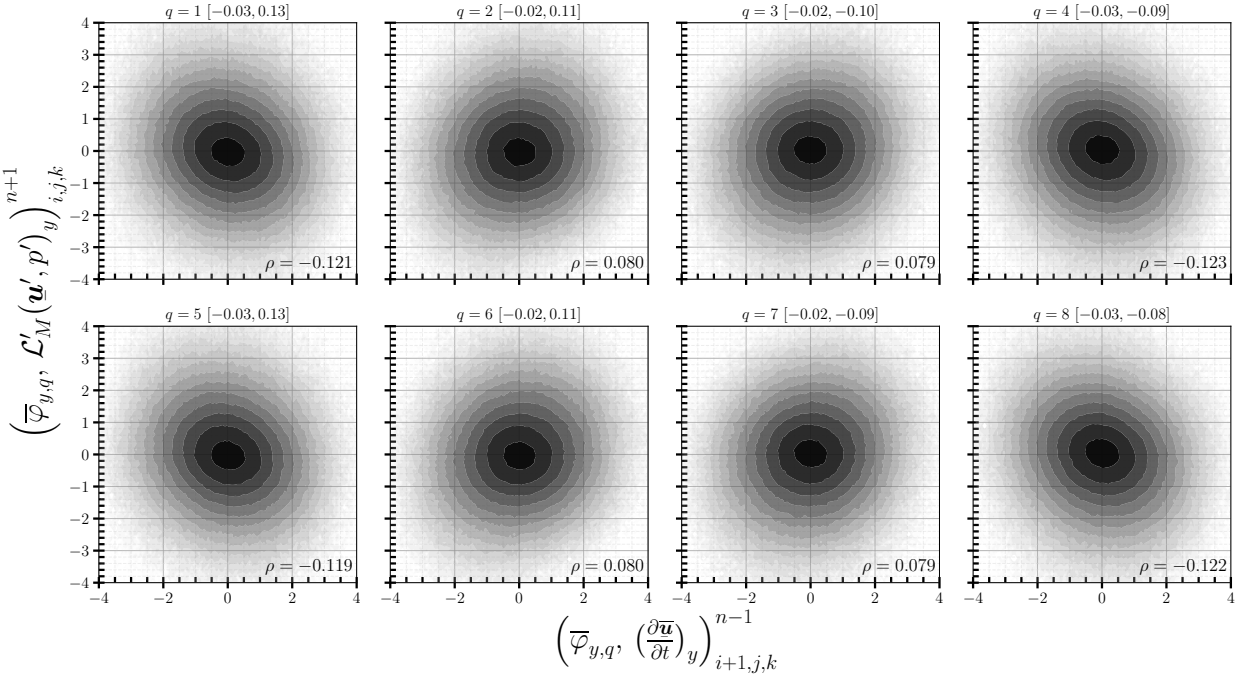


FIGURE A.307 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

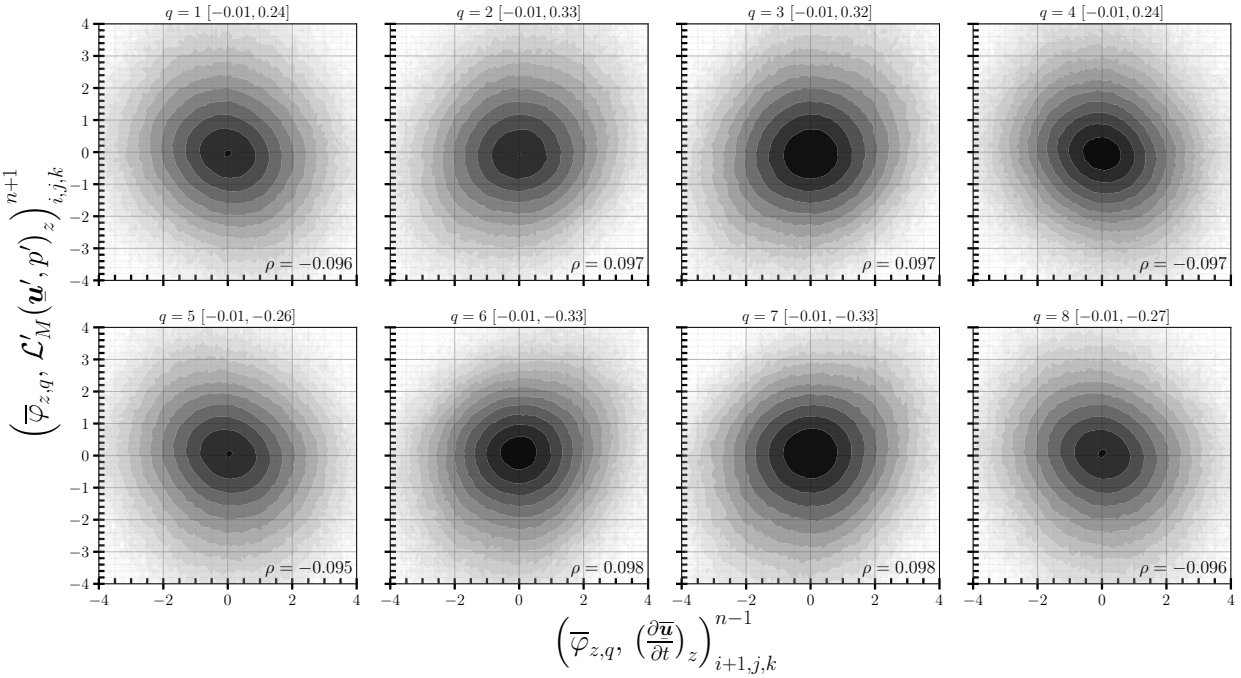


FIGURE A.308 Histograms of the t^{n-1} $(i+1, j, k)$-element *z*-momentum time derivative term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

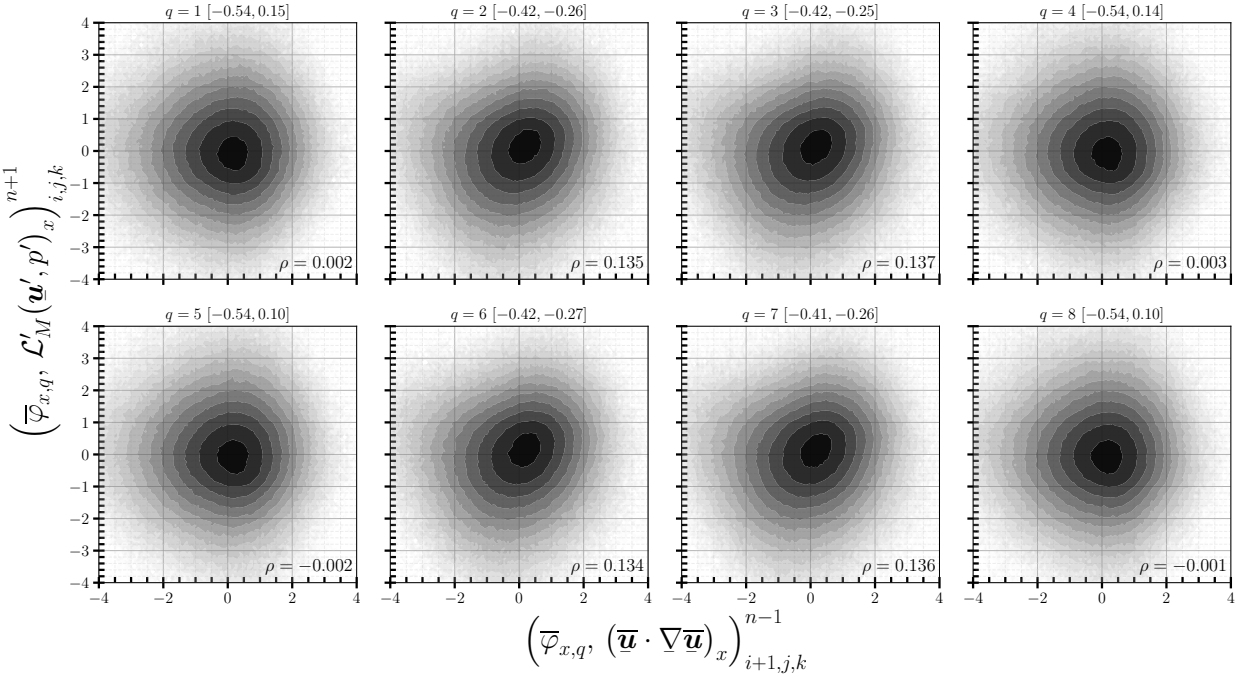


FIGURE A.309 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

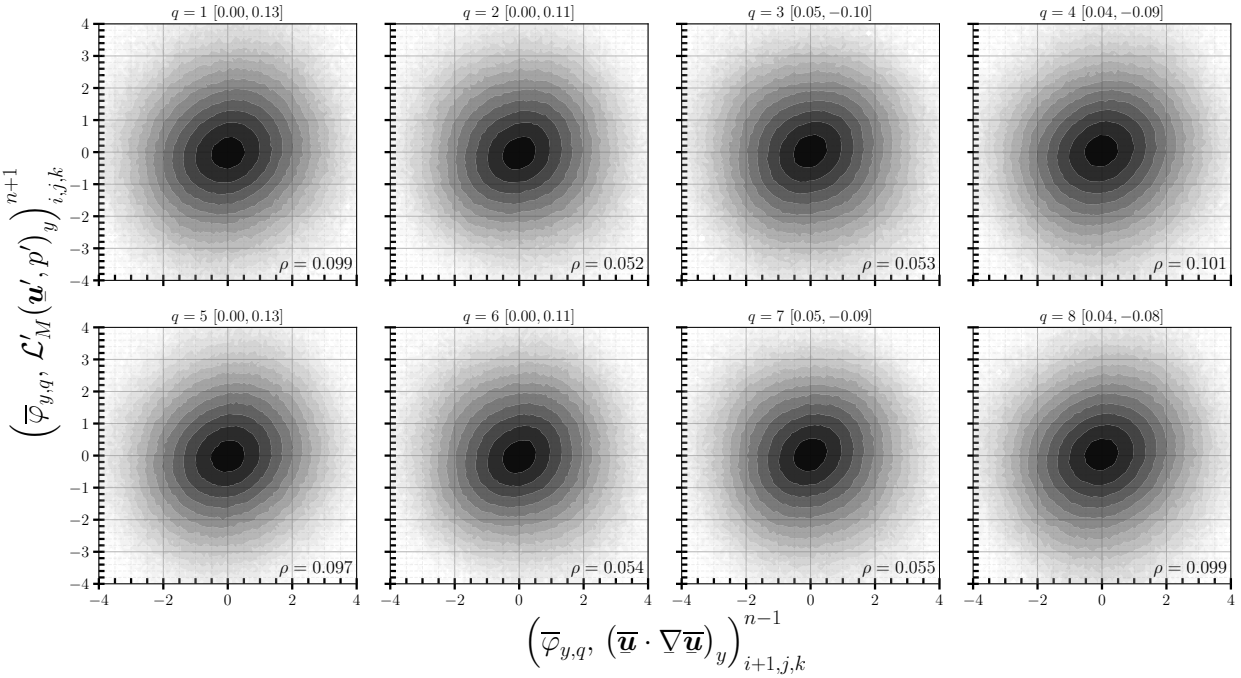


FIGURE A.310 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

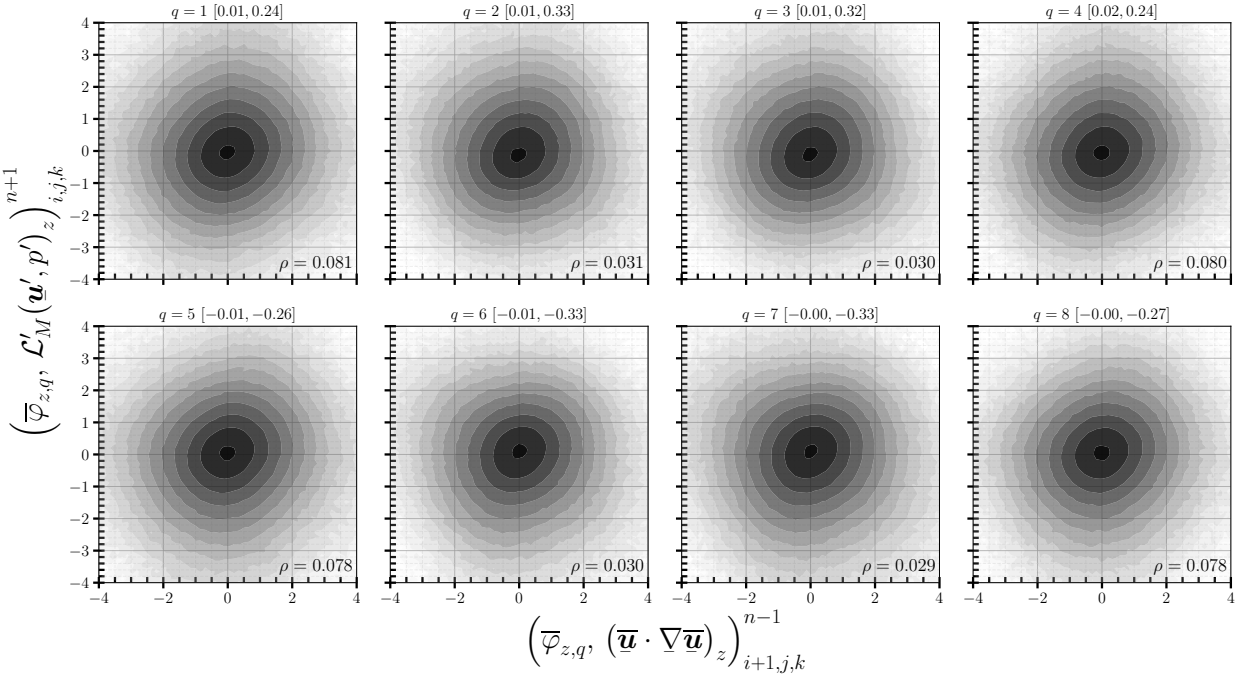


FIGURE A.311 Histograms of the t^{n-1} $(i+1, j, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

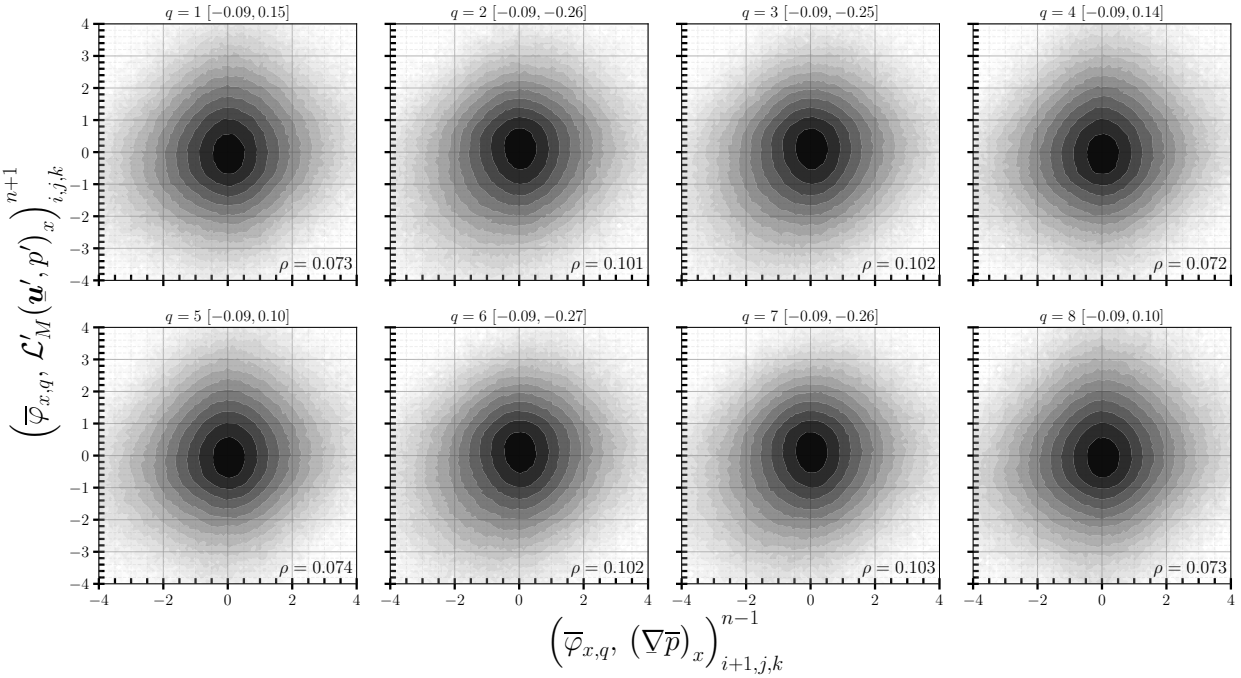


FIGURE A.312 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

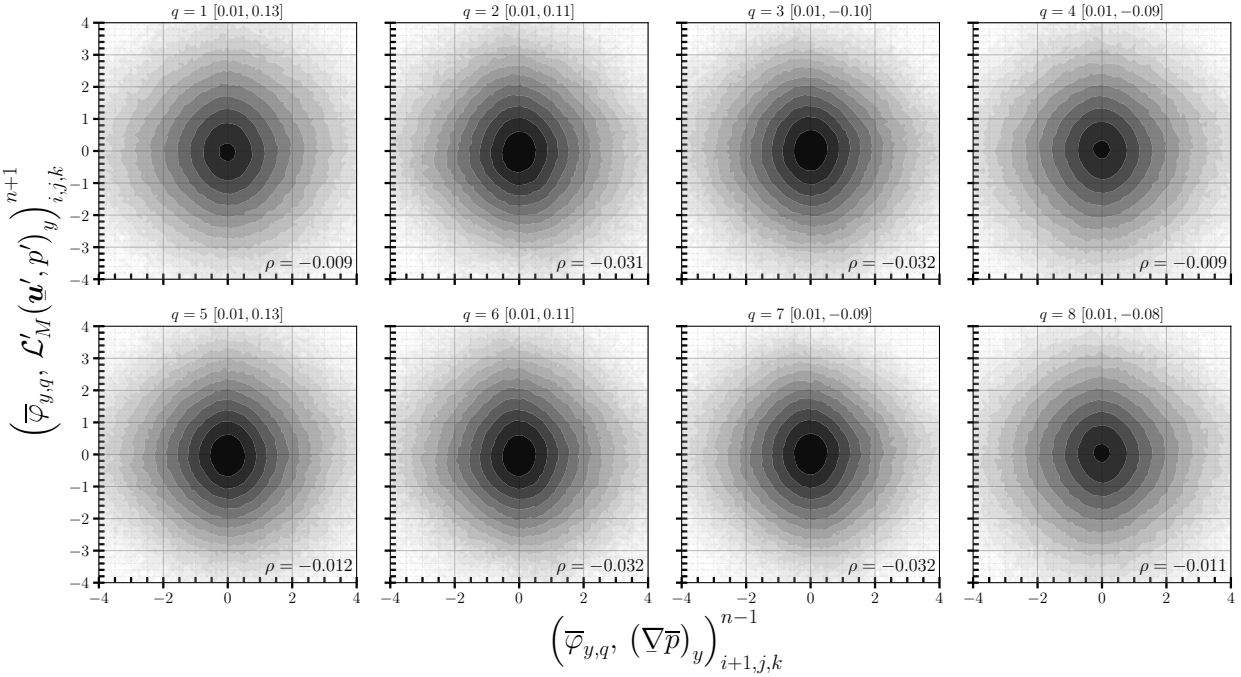


FIGURE A.313 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

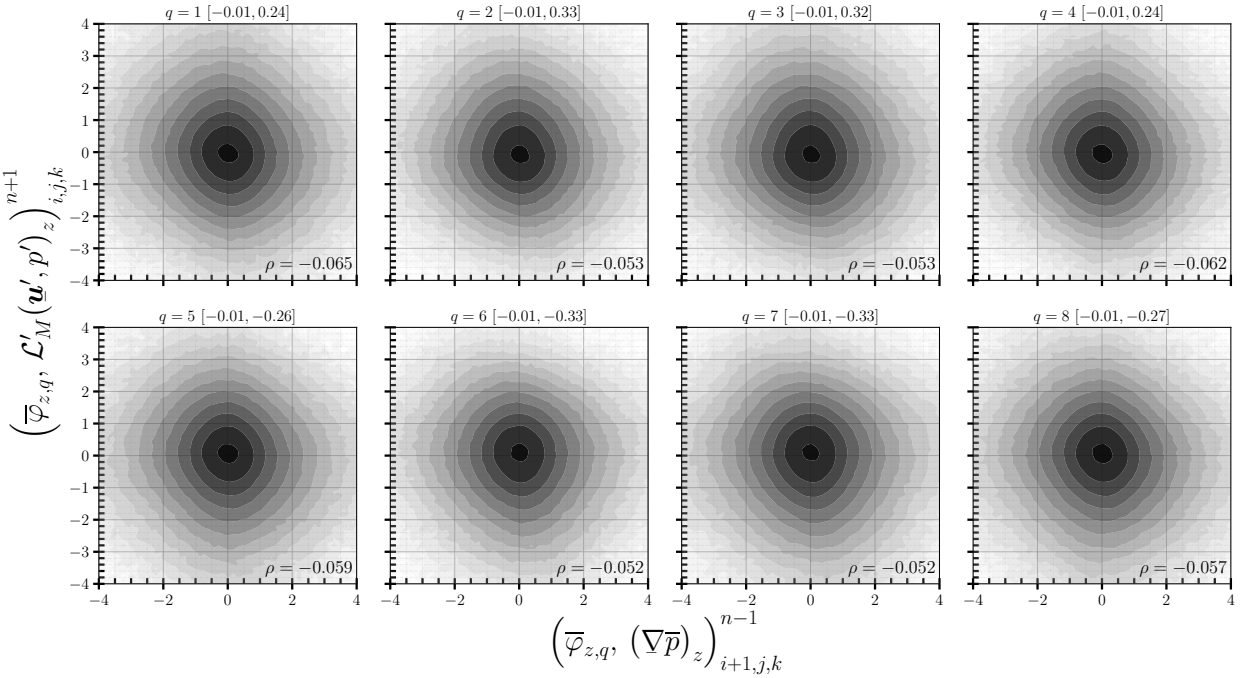


FIGURE A.314 Histograms of the t^{n-1} $(i+1, j, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

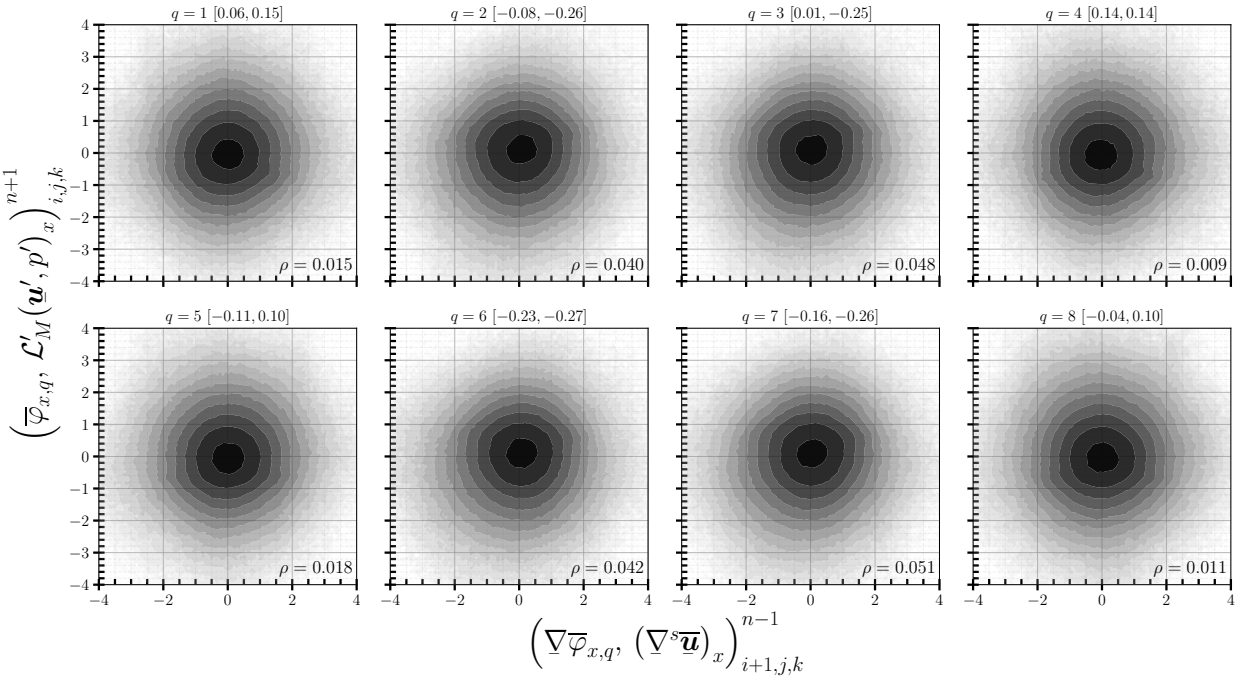


FIGURE A.315 Histograms of the t^{n-1} $(i+1, j, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

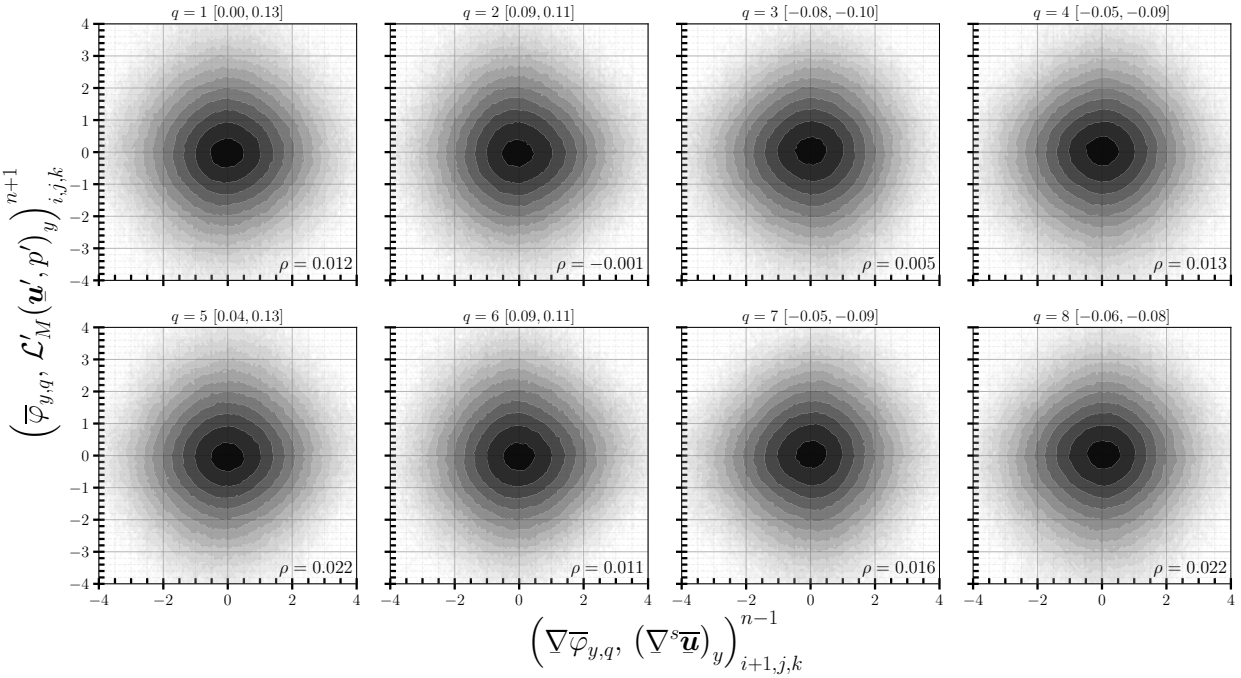


FIGURE A.316 Histograms of the t^{n-1} $(i+1, j, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

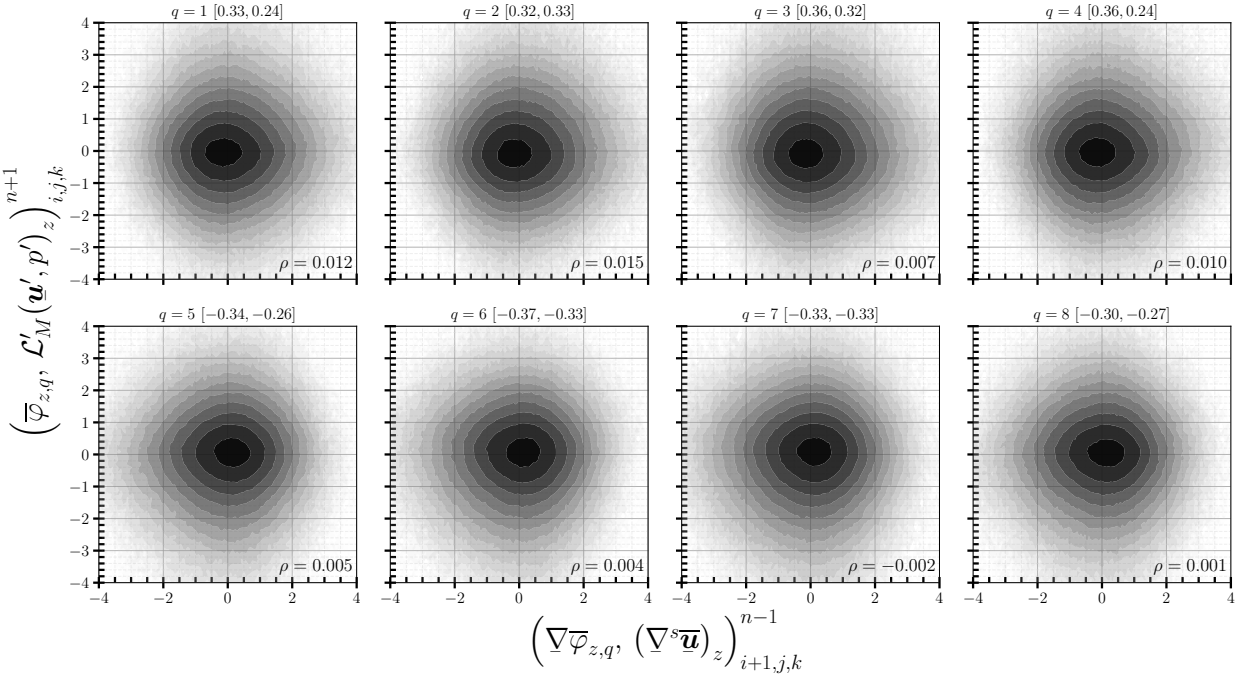


FIGURE A.317 Histograms of the t^{n-1} $(i+1, j, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.7 $(i, j-1, k)$ Histograms

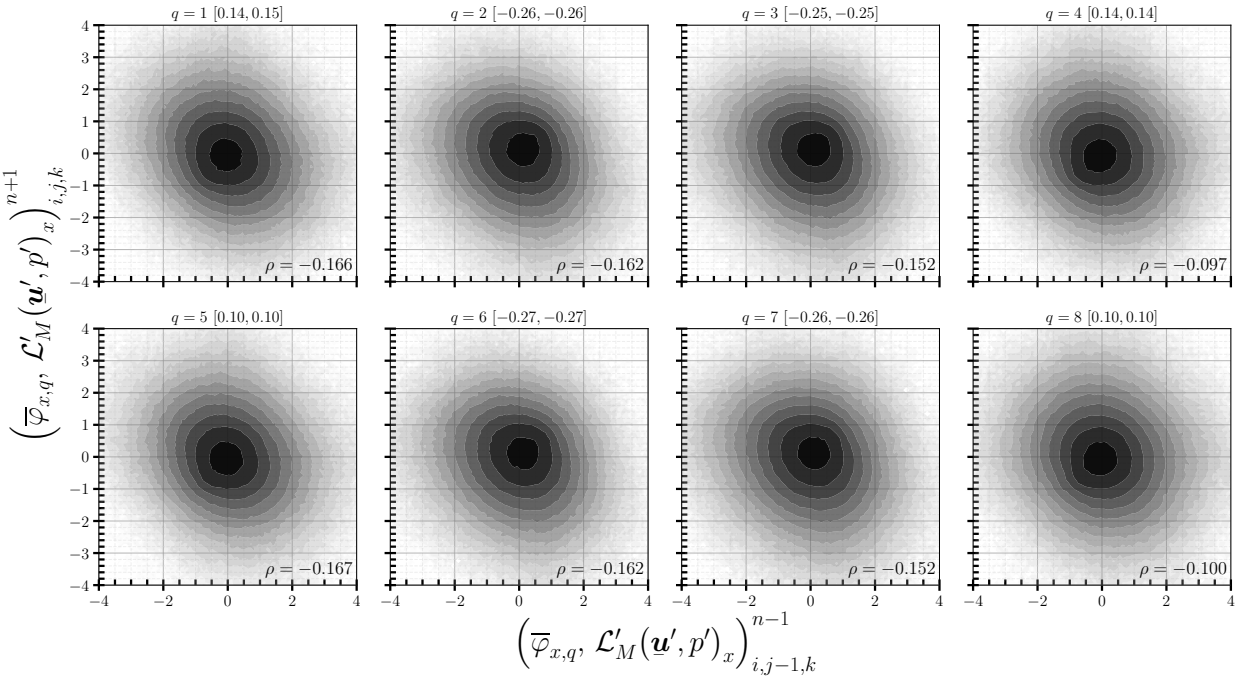


Figure A.318 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

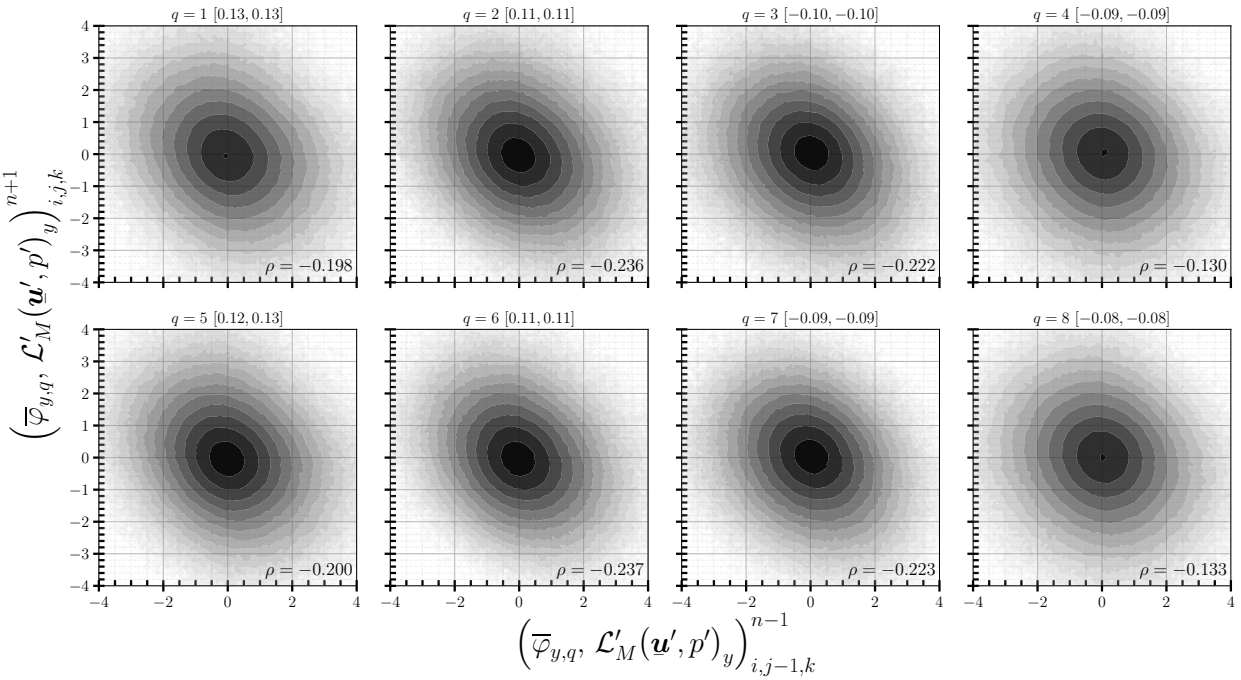


FIGURE A.319 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

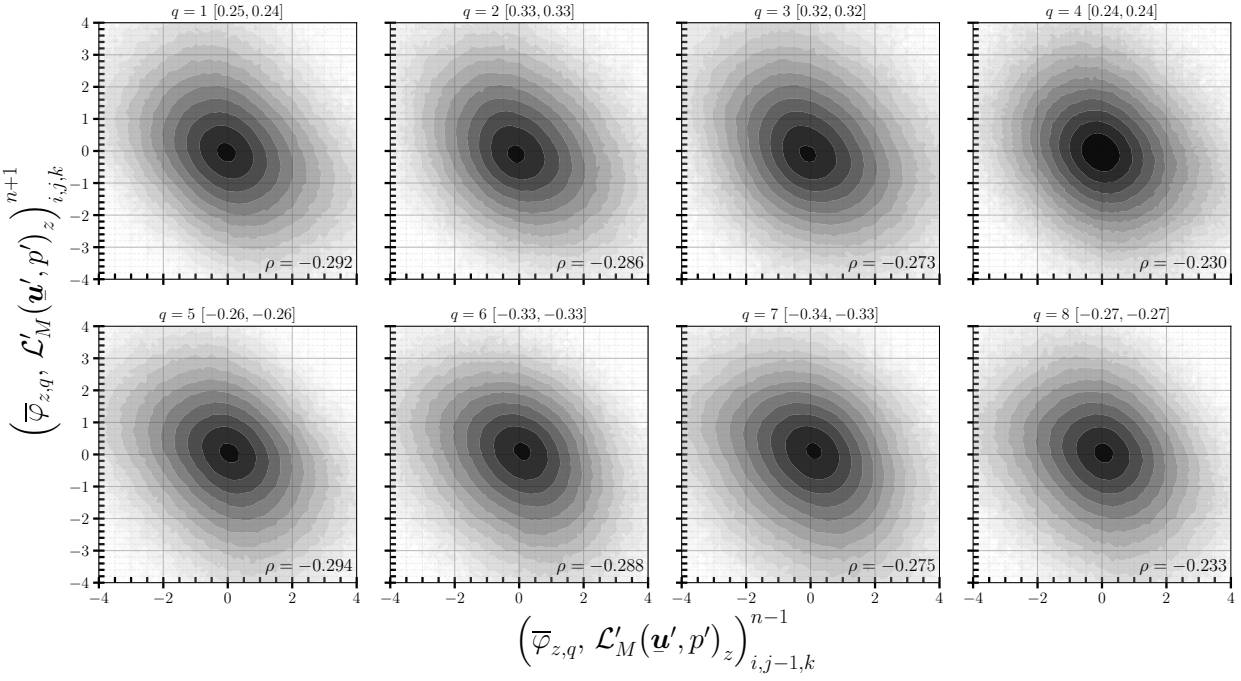


FIGURE A.320 Histograms of the t^{n-1} $(i, j-1, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

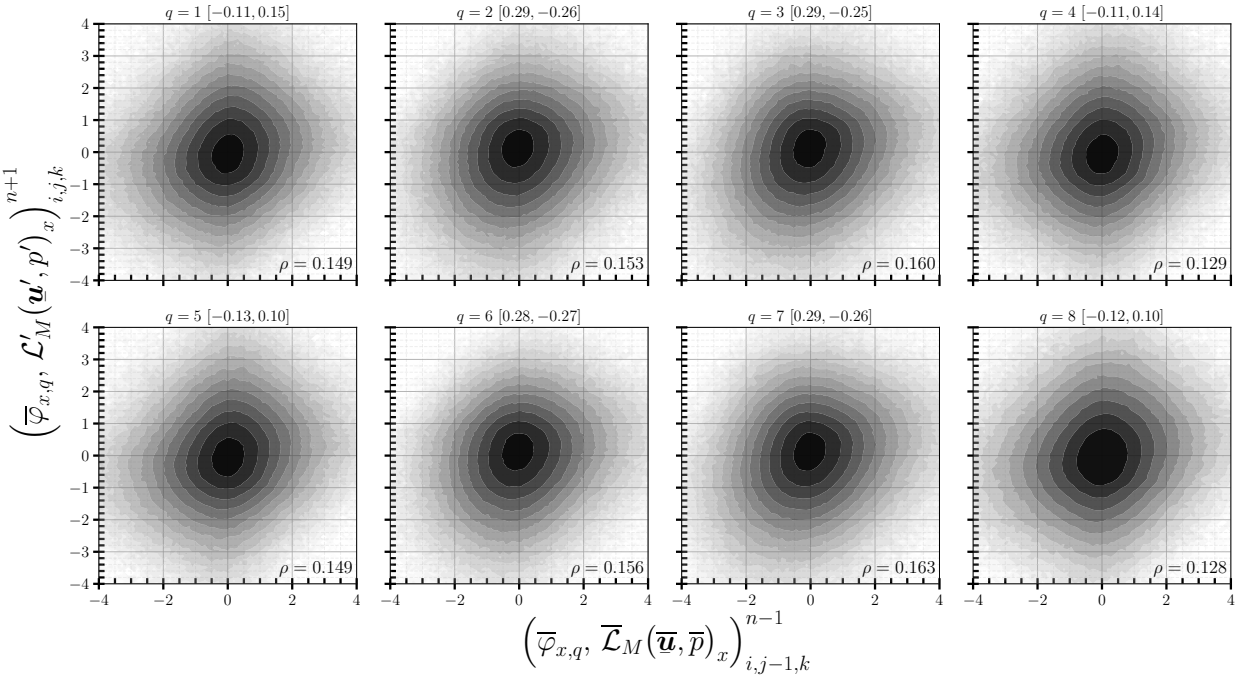


FIGURE A.321 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

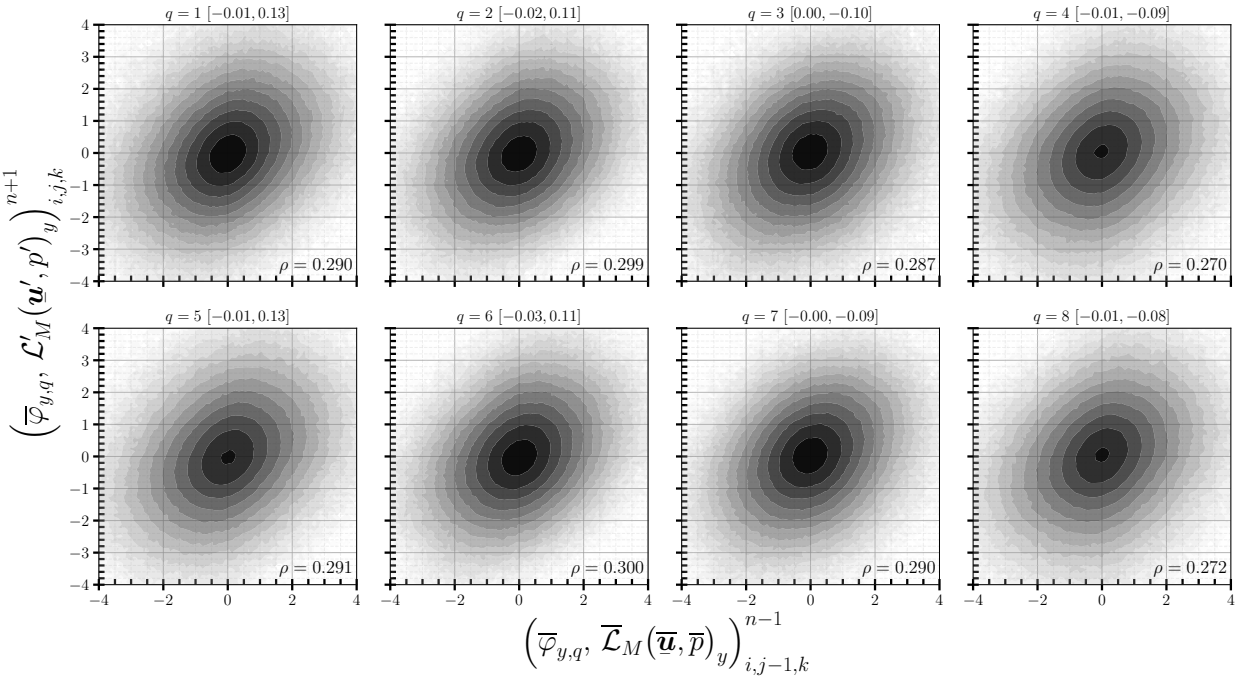


FIGURE A.322 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

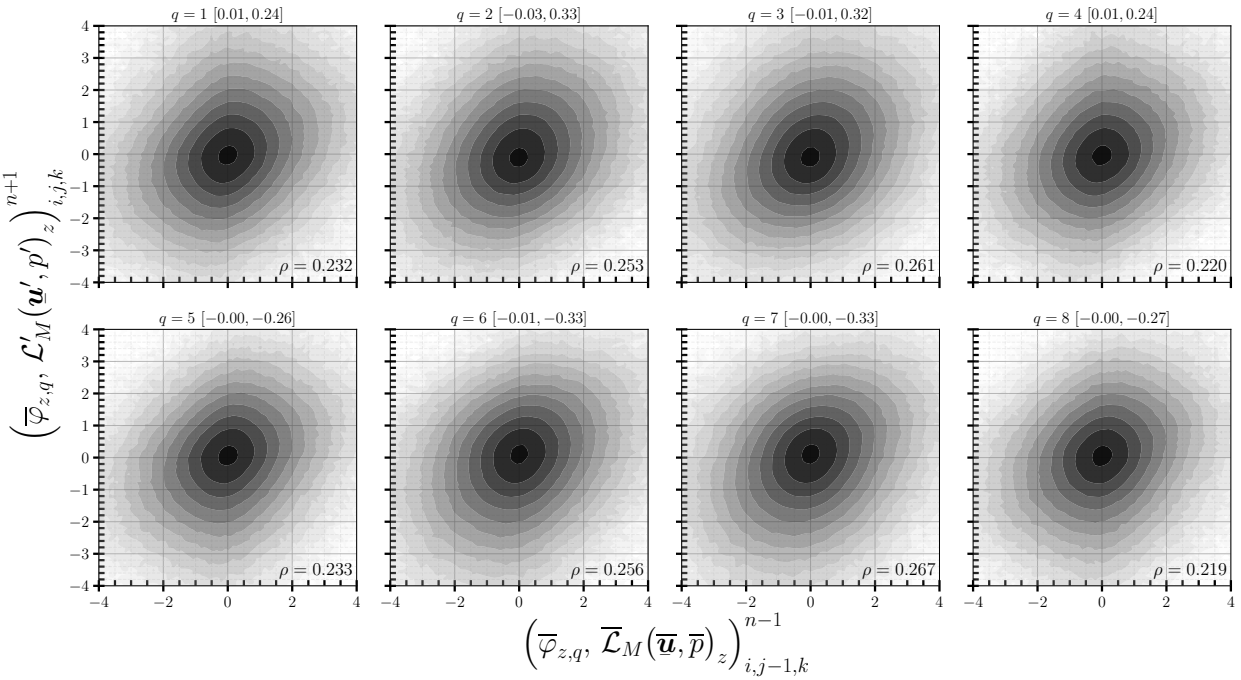


FIGURE A.323 Histograms of the t^{n-1} $(i, j-1, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

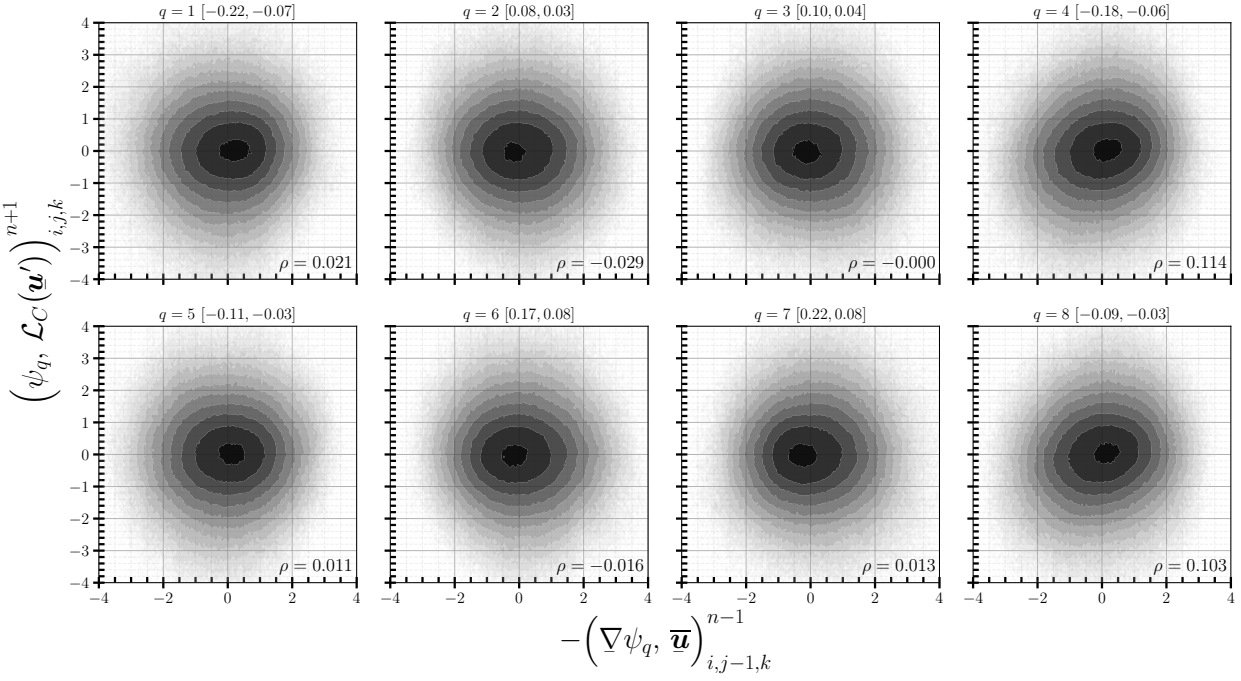


FIGURE A.324 Histograms of the t^{n-1} $(i, j-1, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

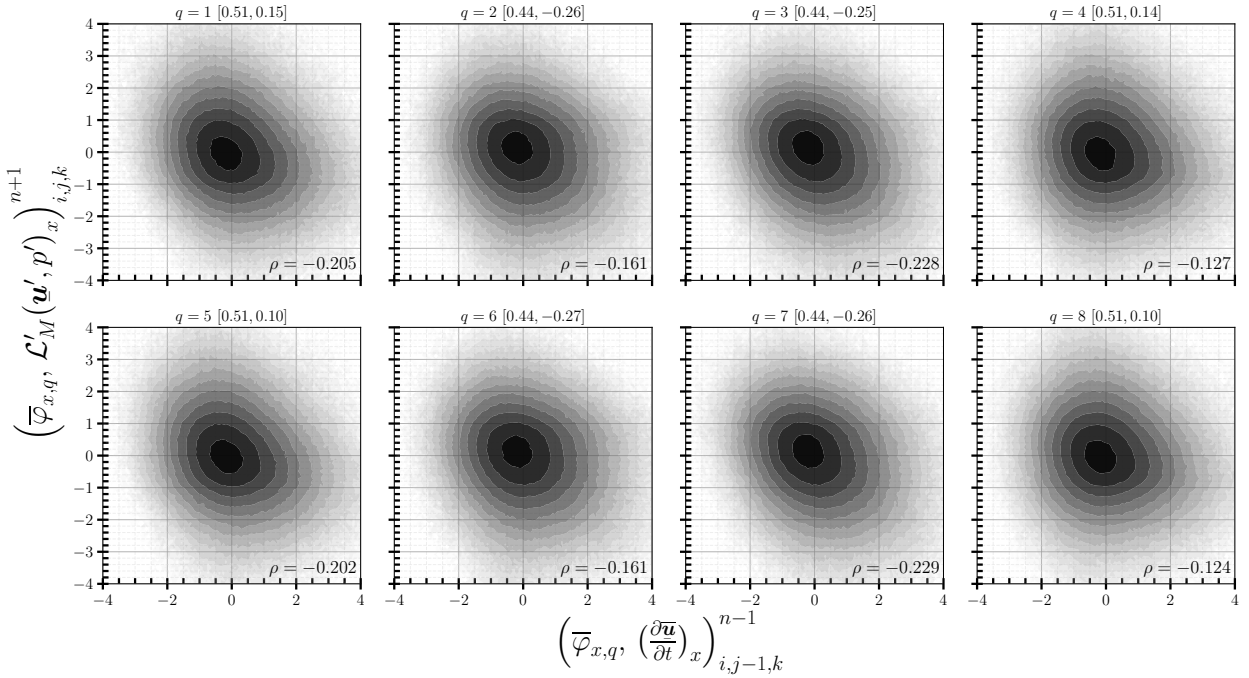


FIGURE A.325 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

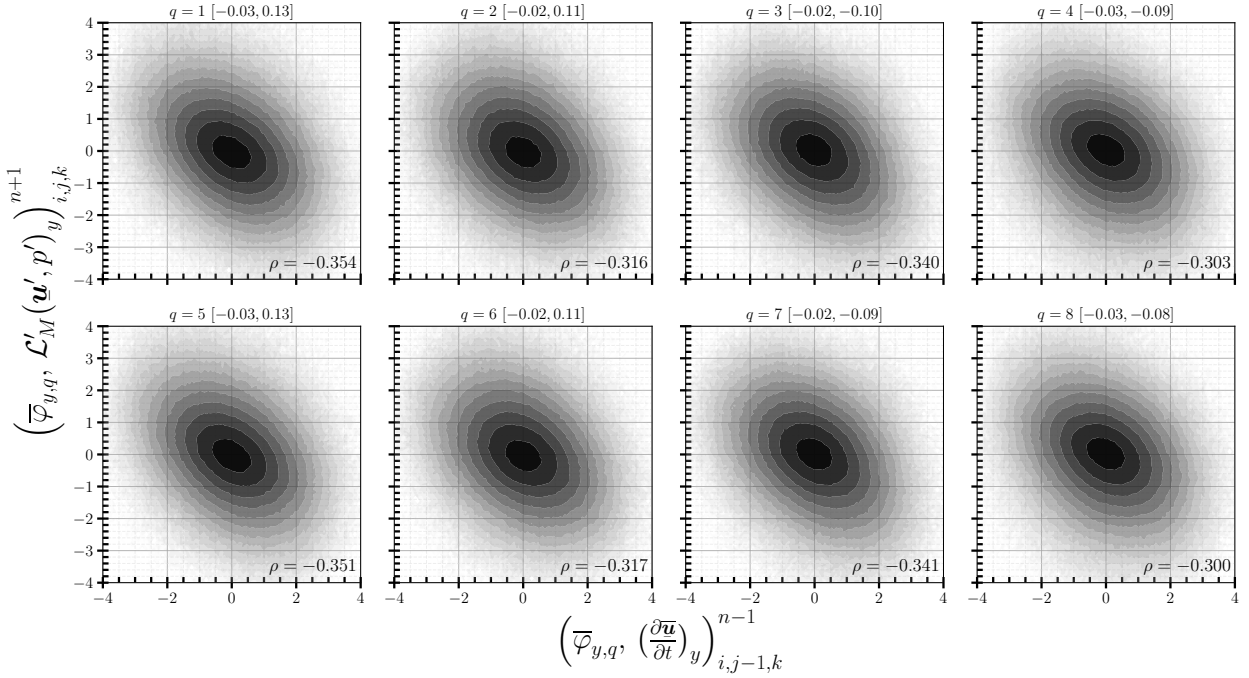


FIGURE A.326 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

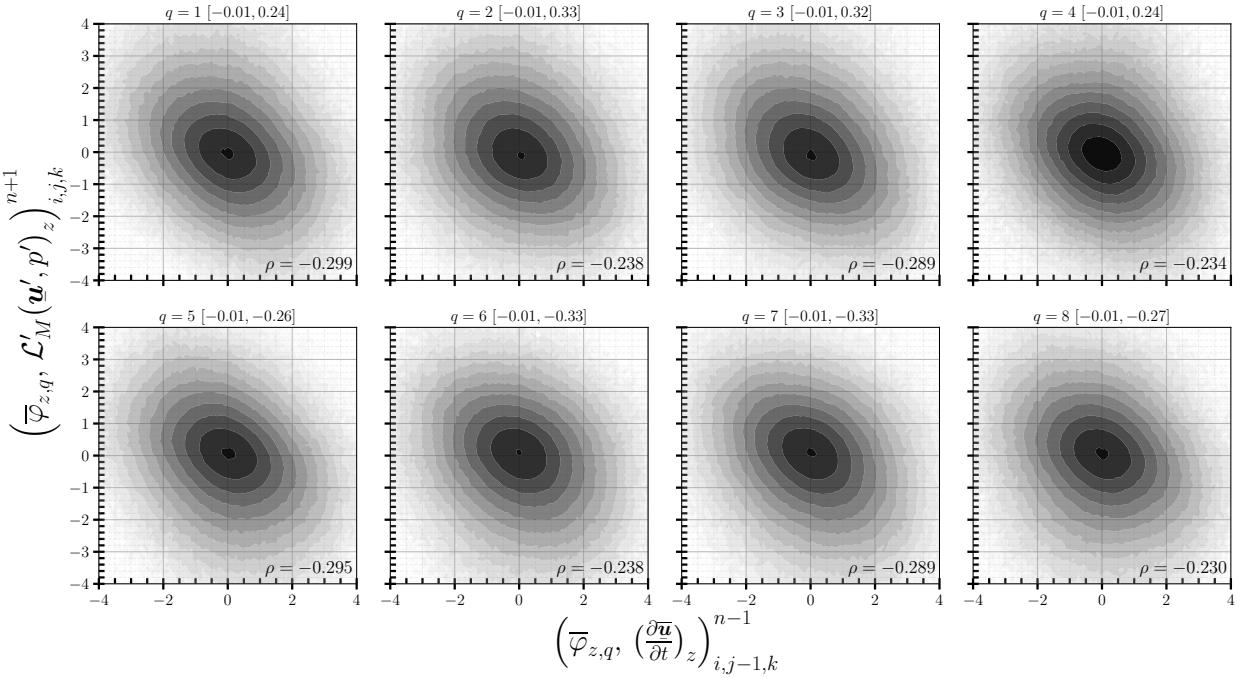


FIGURE A.327 Histograms of the t^{n-1} $(i, j-1, k)$-element *z*-momentum time derivative term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

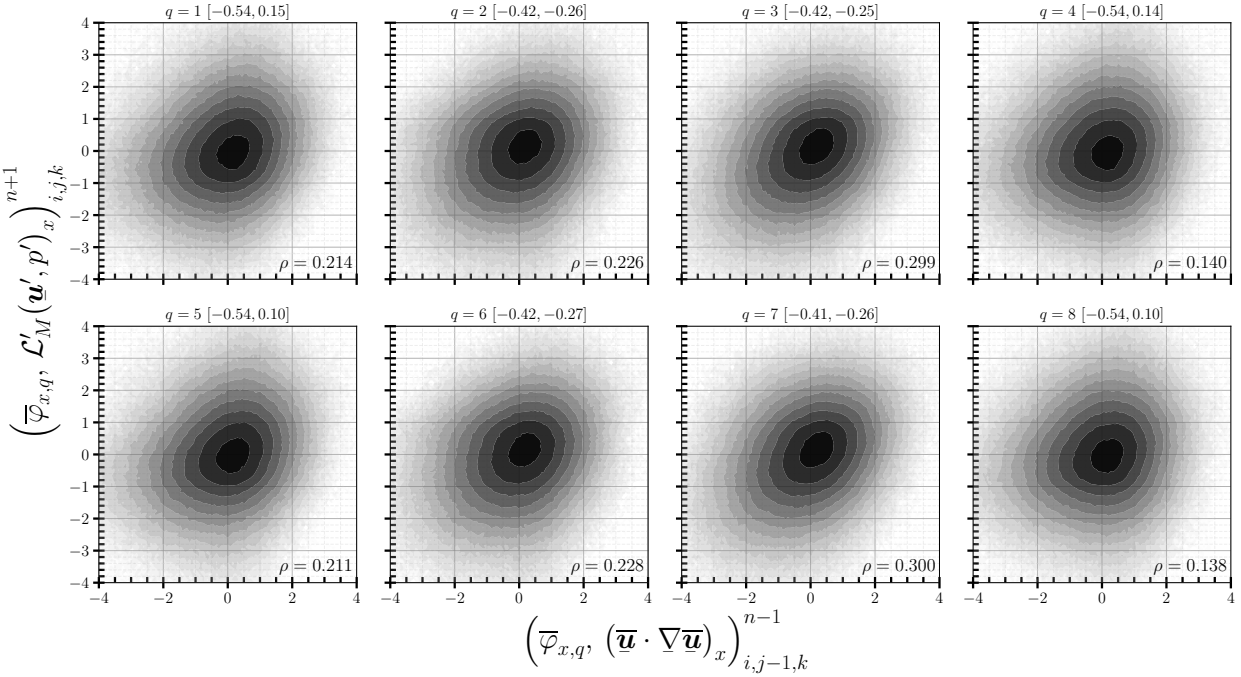


FIGURE A.328 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

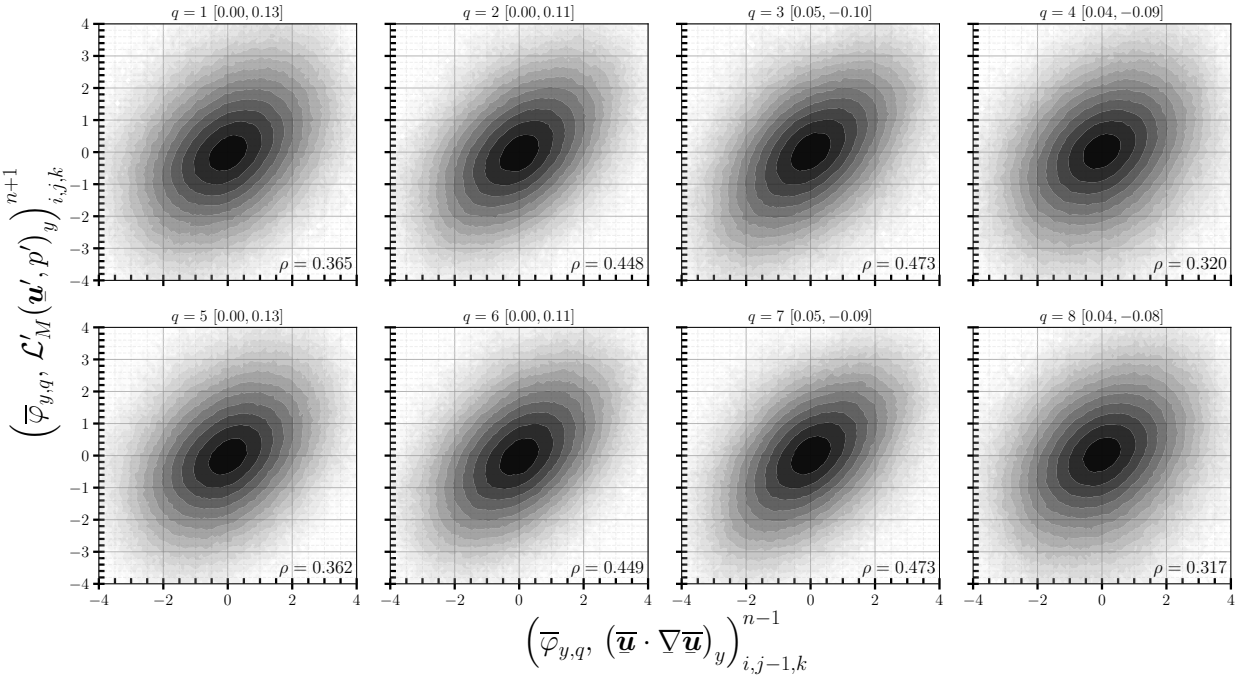


FIGURE A.329 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

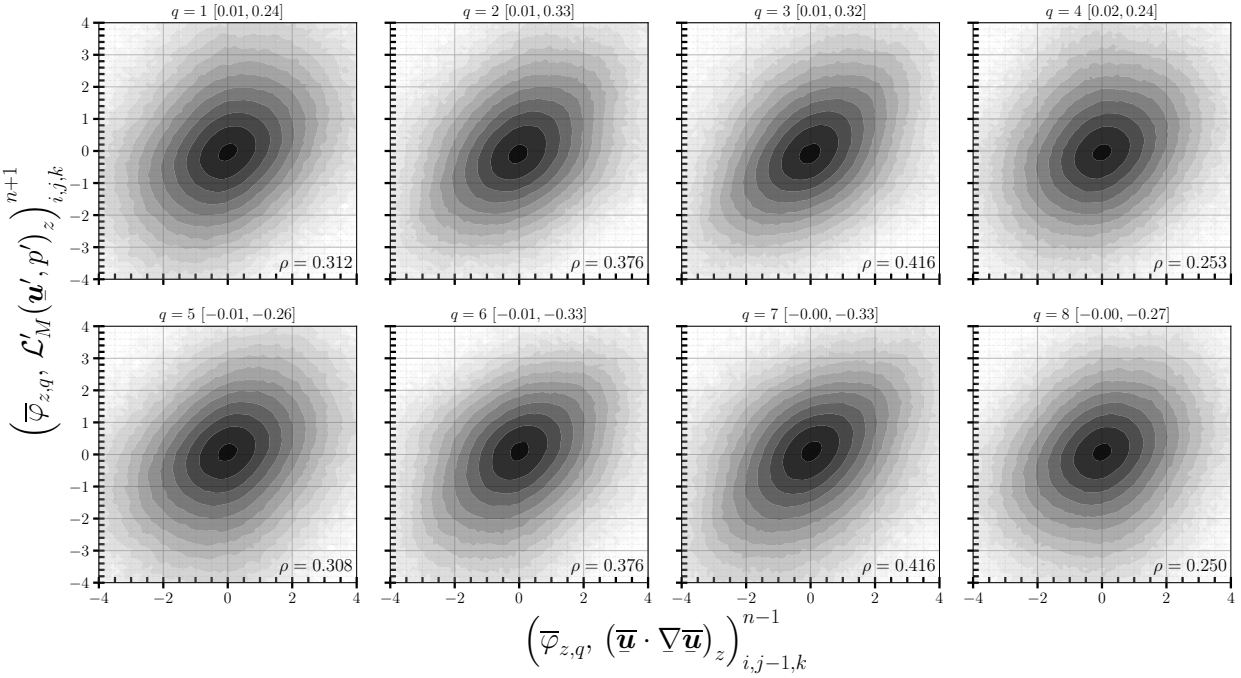


FIGURE A.330 Histograms of the t^{n-1} $(i, j-1, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

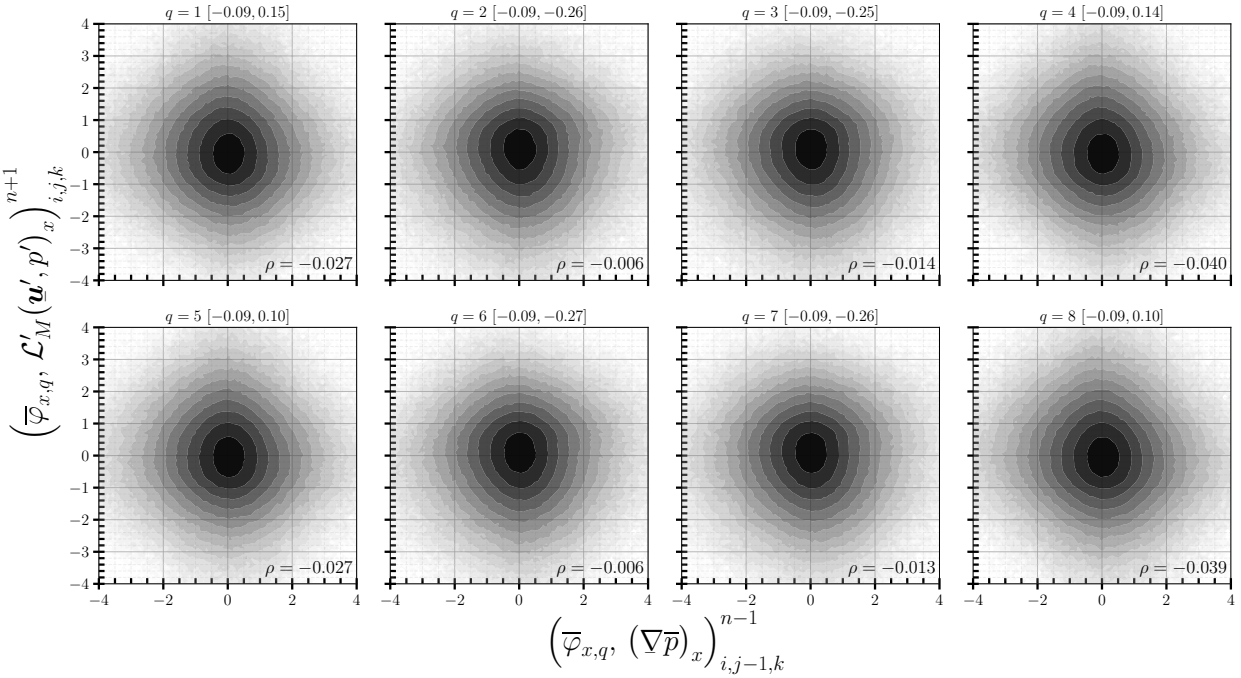


FIGURE A.331 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

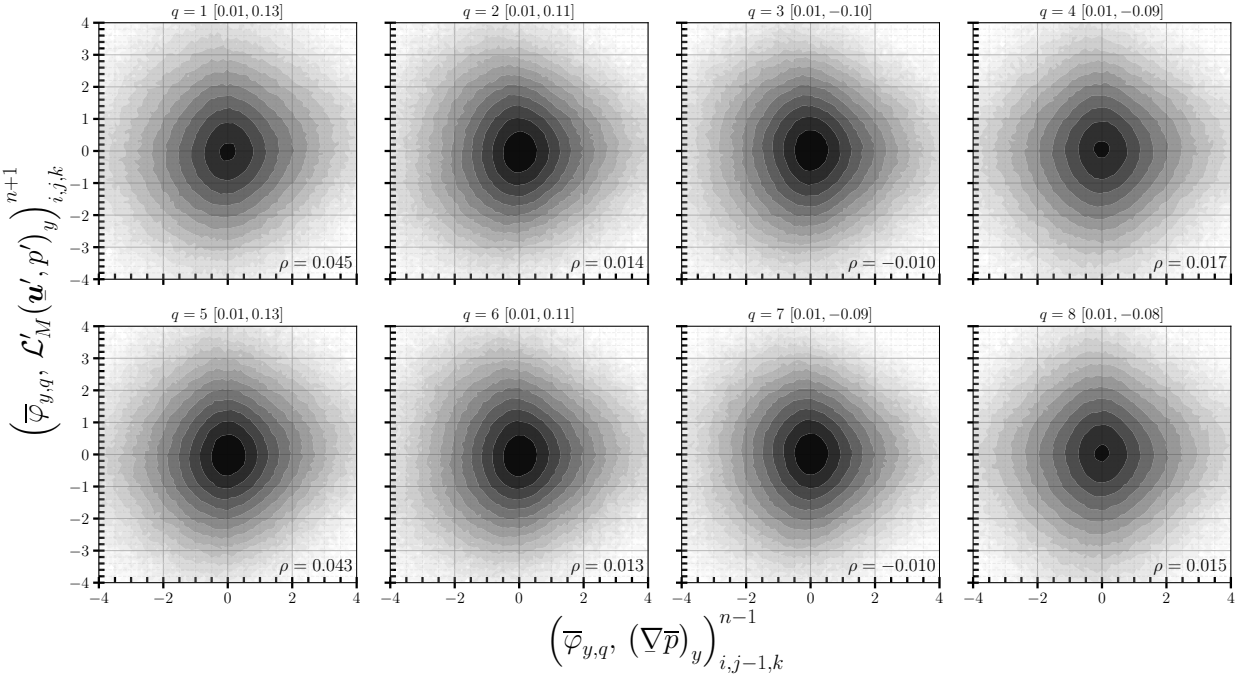


FIGURE A.332 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

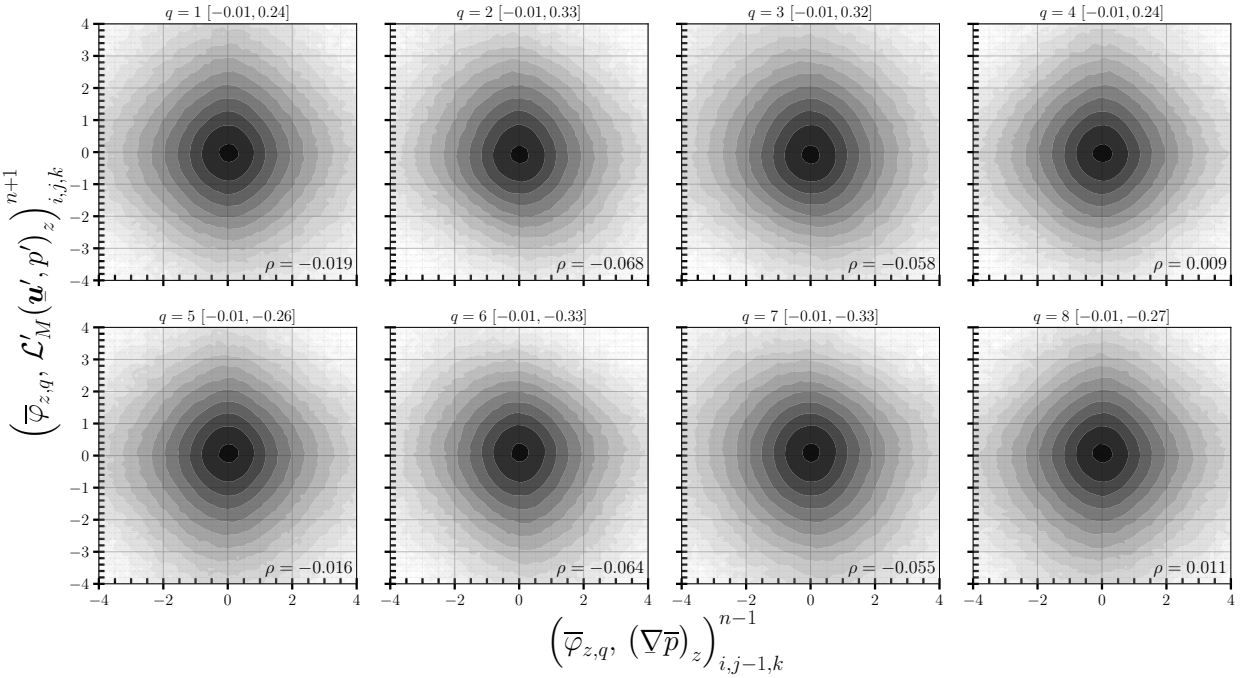


FIGURE A.333 Histograms of the t^{n-1} $(i, j-1, k)$-element z-momentum pressure gradient term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

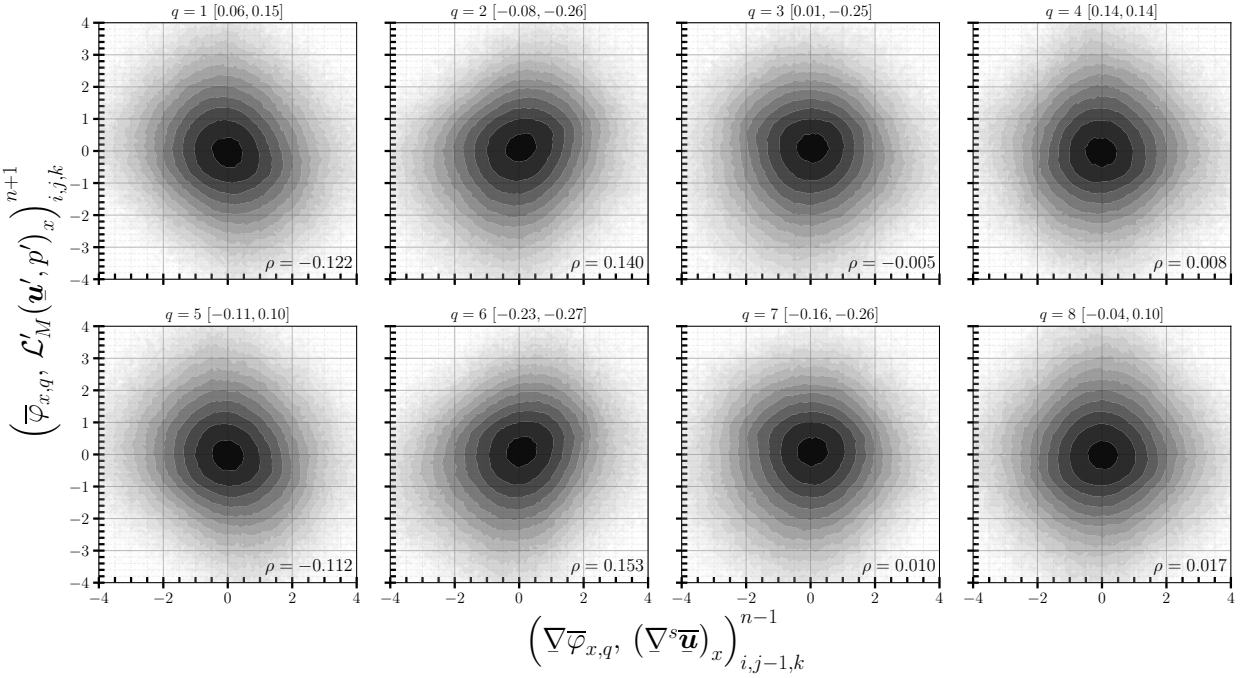


FIGURE A.334 Histograms of the t^{n-1} $(i, j-1, k)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

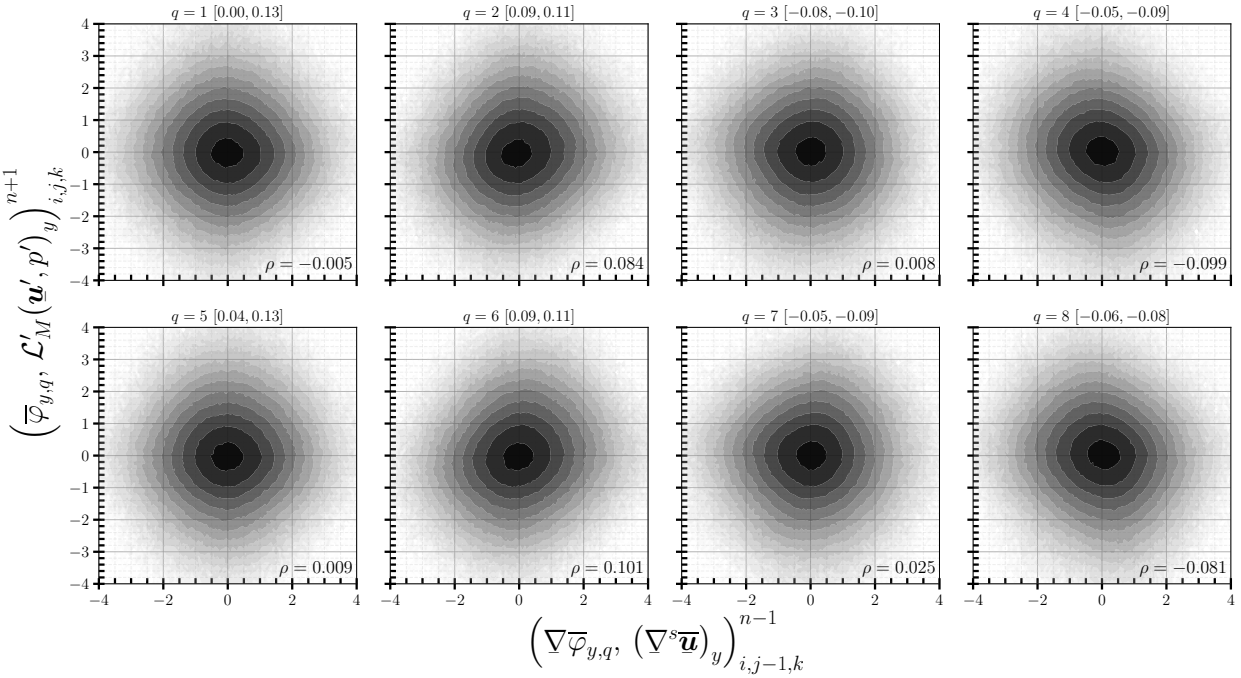


FIGURE A.335 Histograms of the t^{n-1} $(i, j-1, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

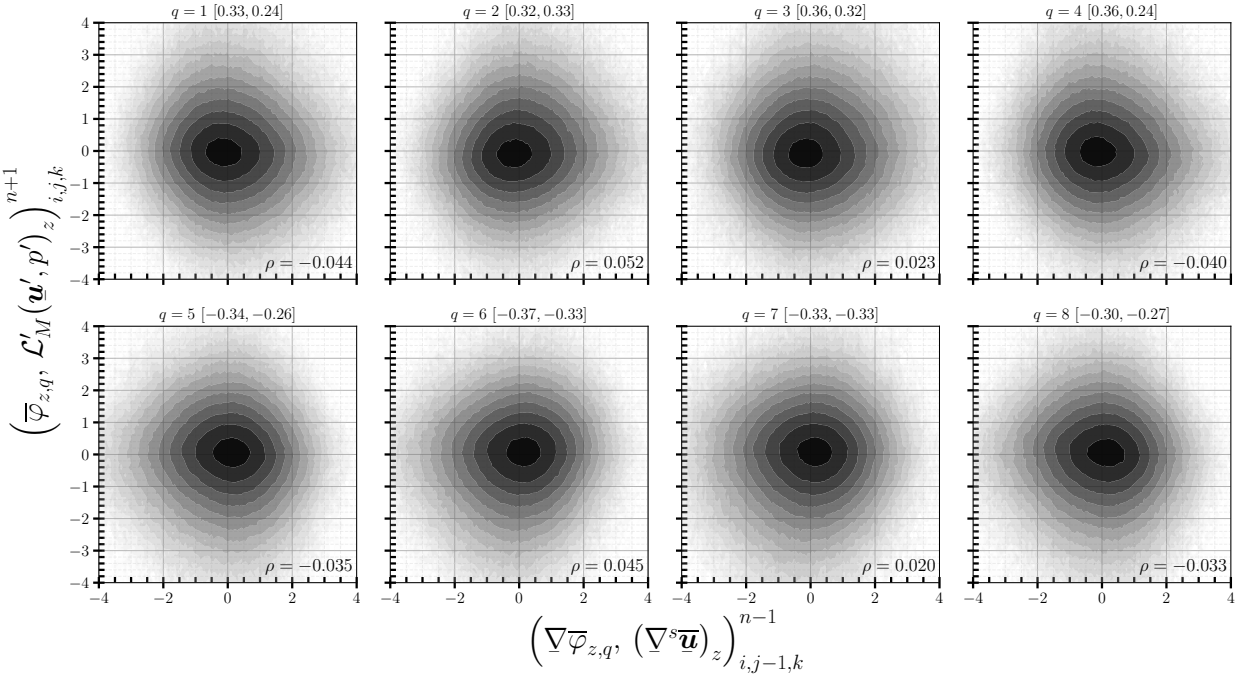


FIGURE A.336 Histograms of the t^{n-1} $(i, j-1, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.8 $(i, j+1, k)$ Histograms

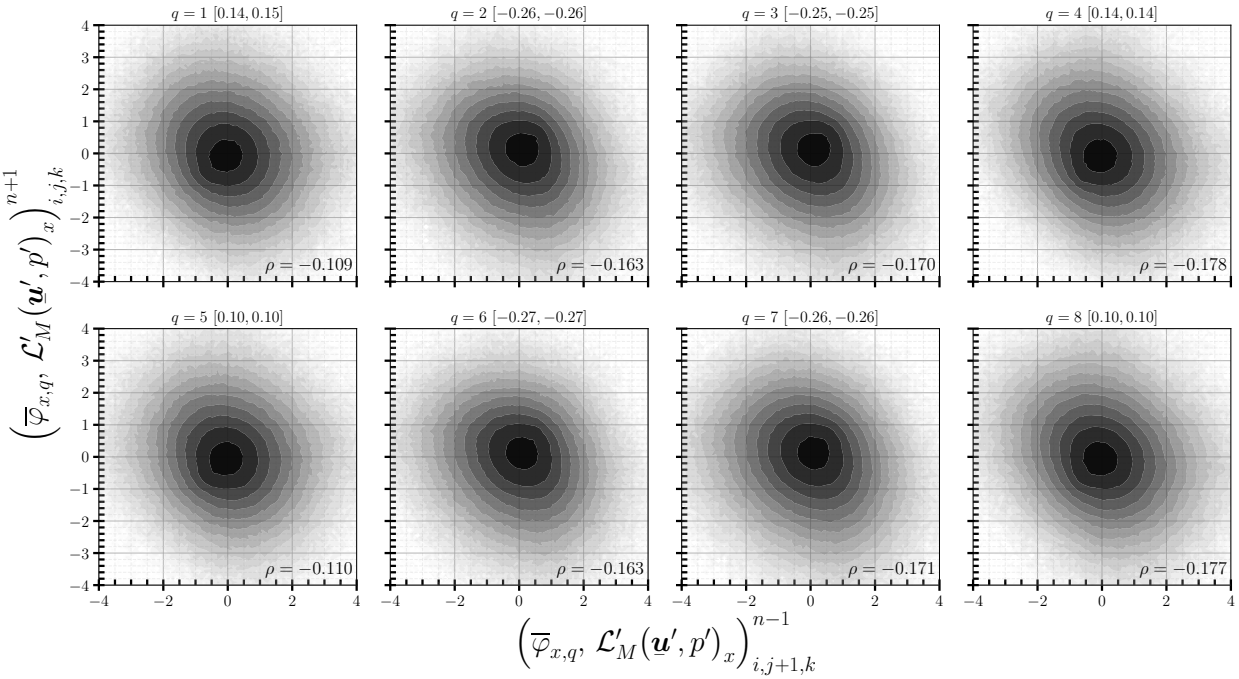


FIGURE A.337 Histograms of the t^{n-1} $(i, j+1, k)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

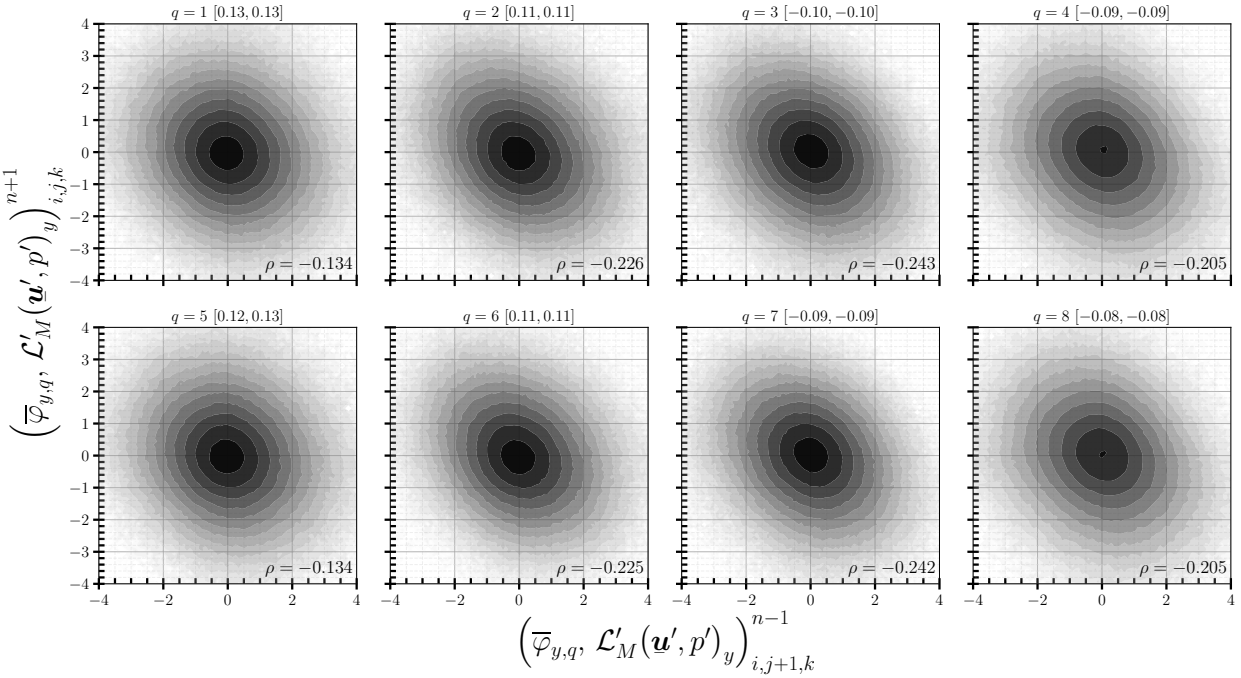


FIGURE A.338 Histograms of the t^{n-1} $(i, j+1, k)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

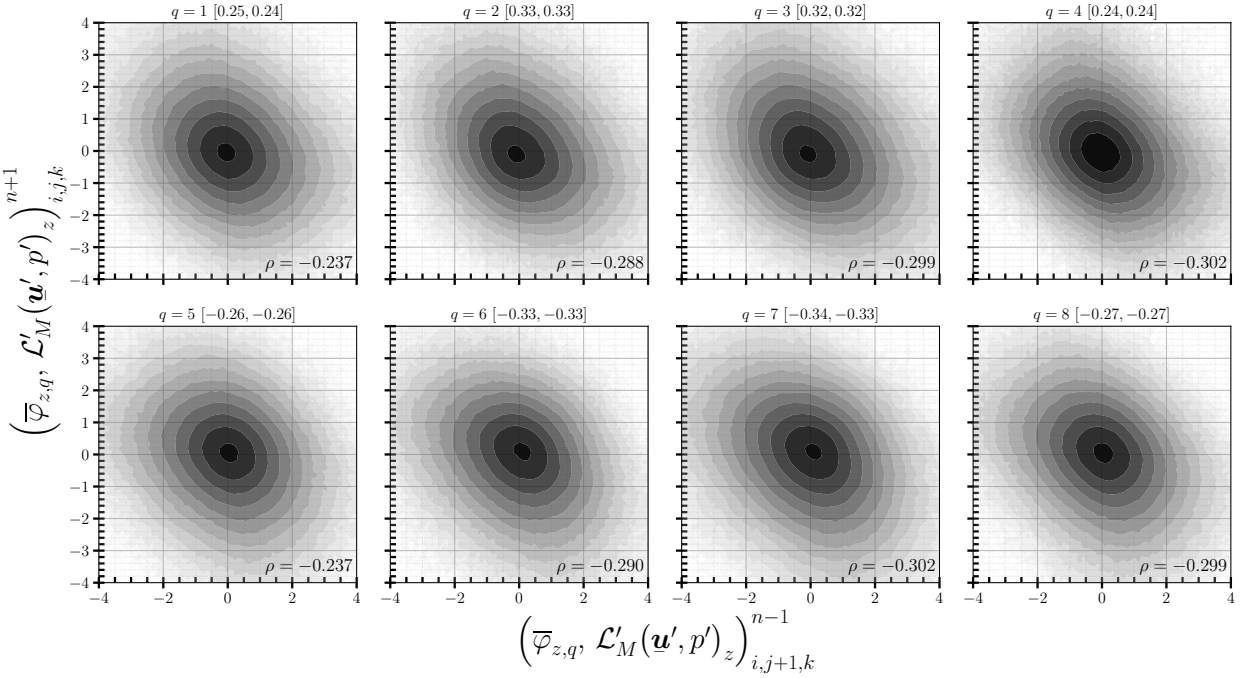


FIGURE A.339 Histograms of the t^{n-1} $(i, j+1, k)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

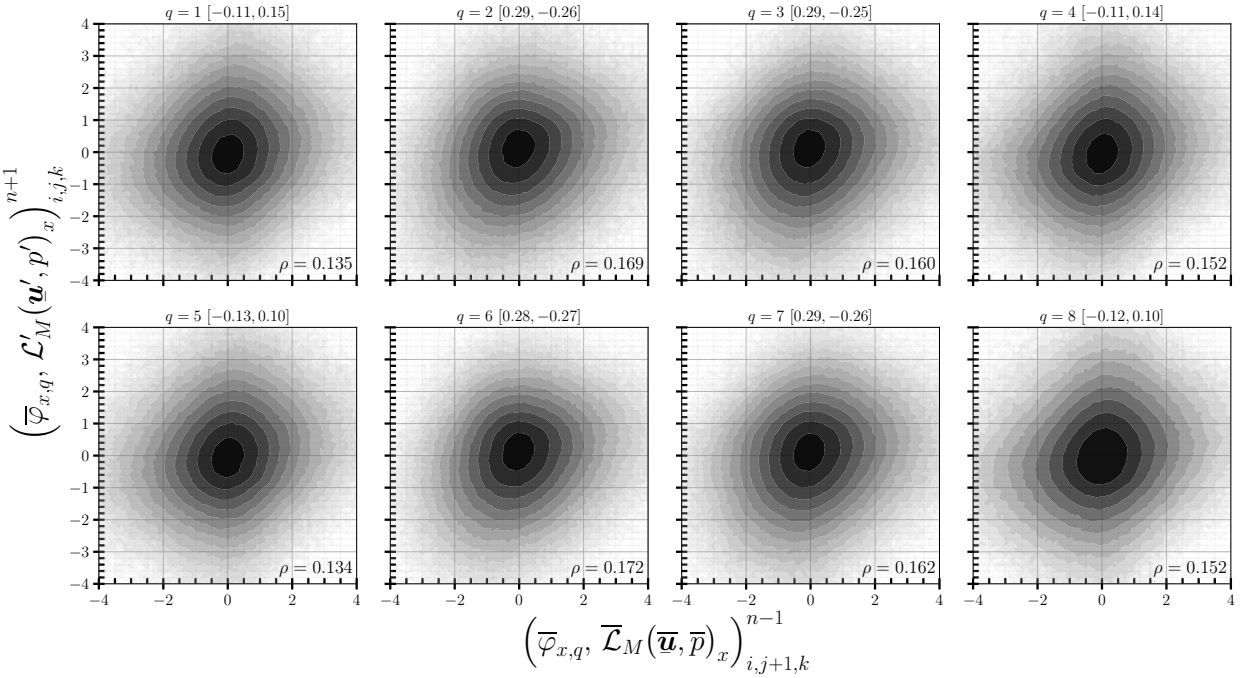


FIGURE A.340 Histograms of the t^{n-1} $(i, j+1, k)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

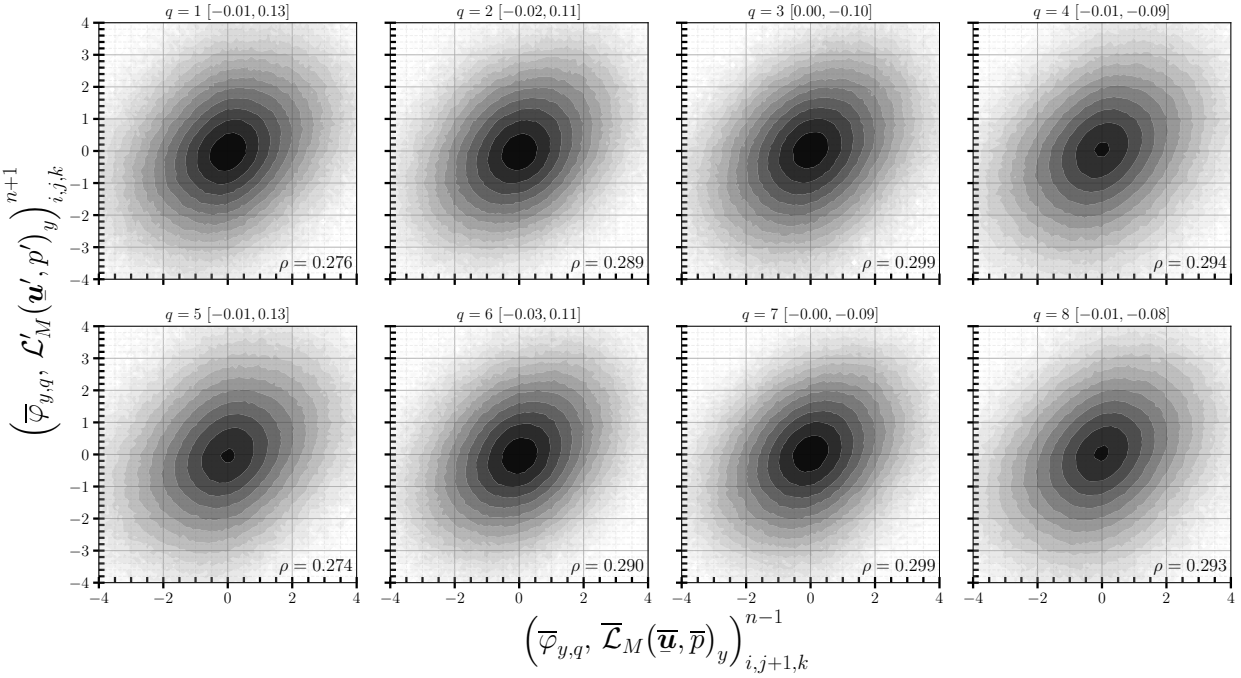


FIGURE A.341 Histograms of the t^{n-1} $(i, j+1, k)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

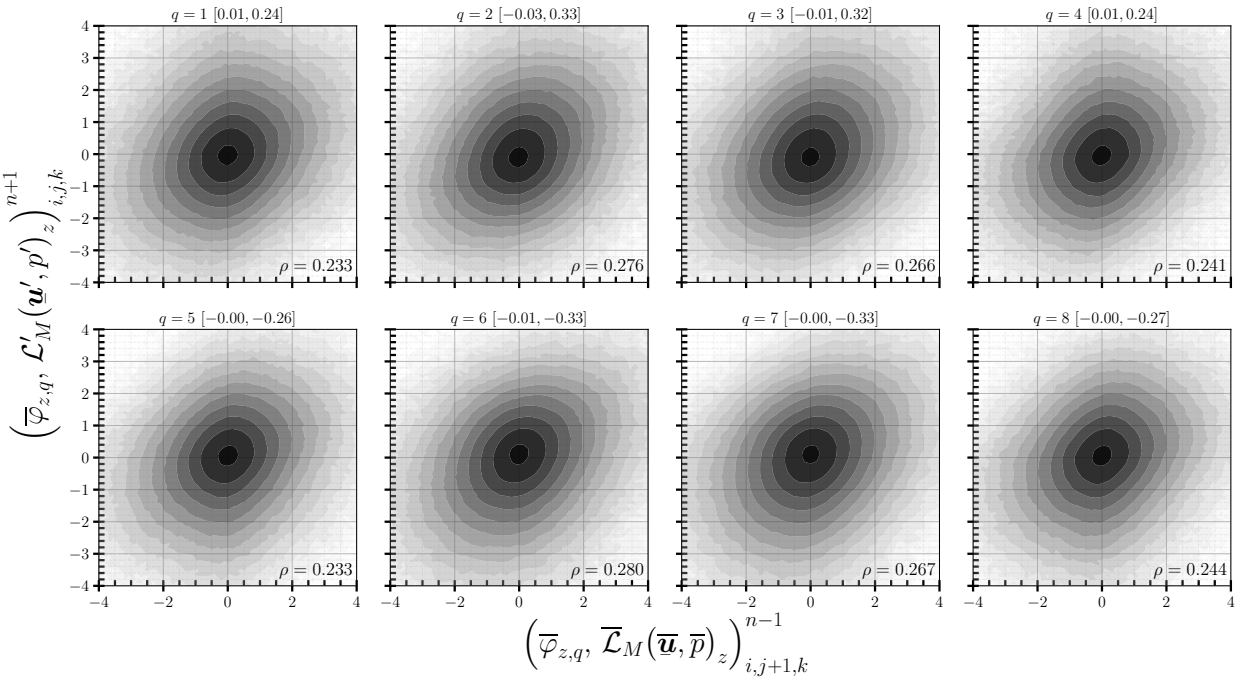


FIGURE A.342 Histograms of the t^{n-1} $(i, j+1, k)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

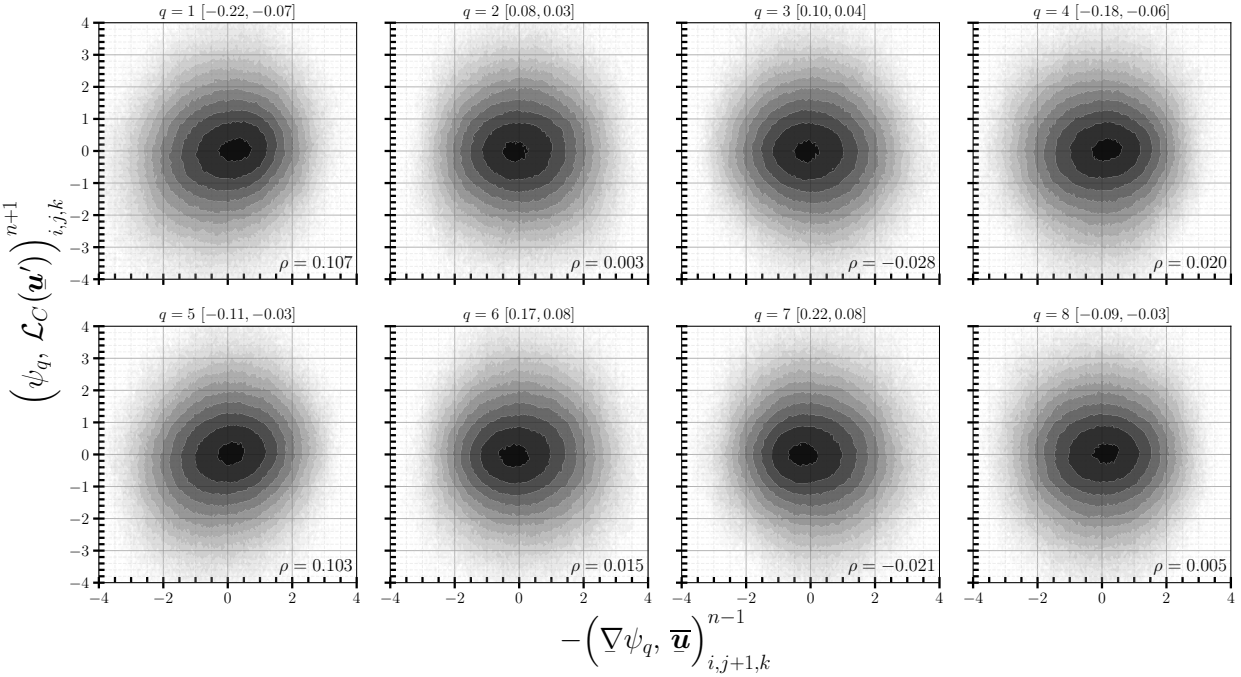


FIGURE A.343 Histograms of the t^{n-1} $(i, j+1, k)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

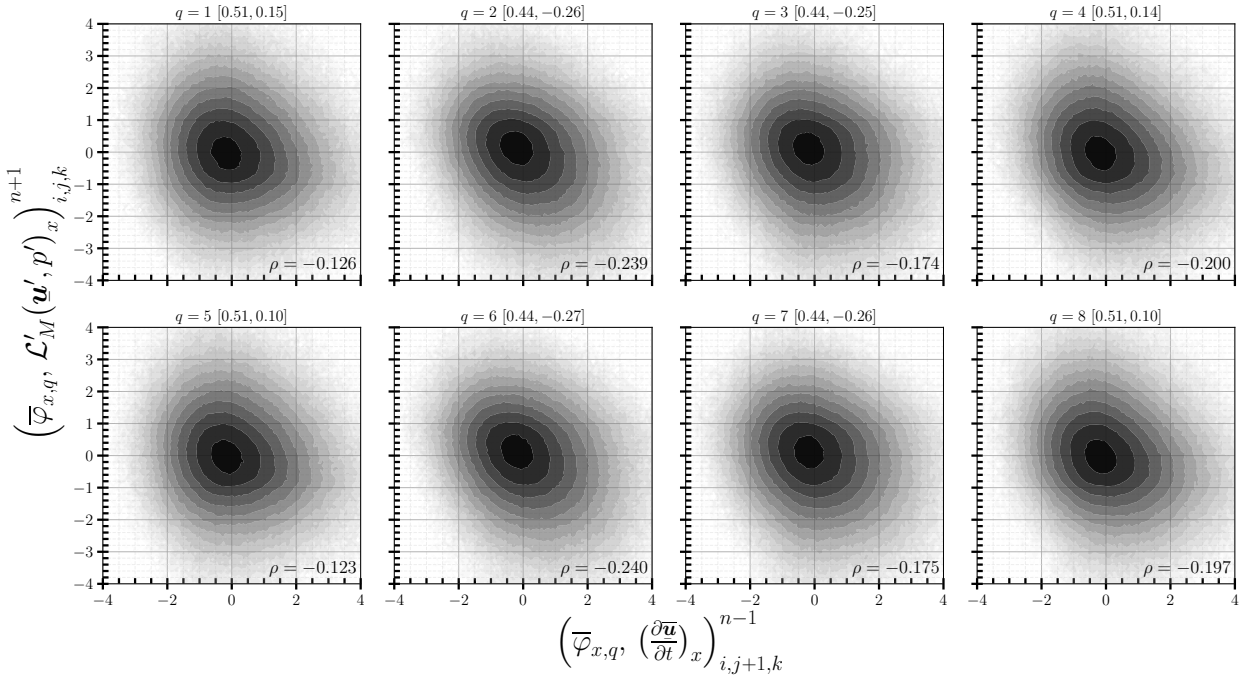


FIGURE A.344 Histograms of the t^{n-1} $(i, j+1, k)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

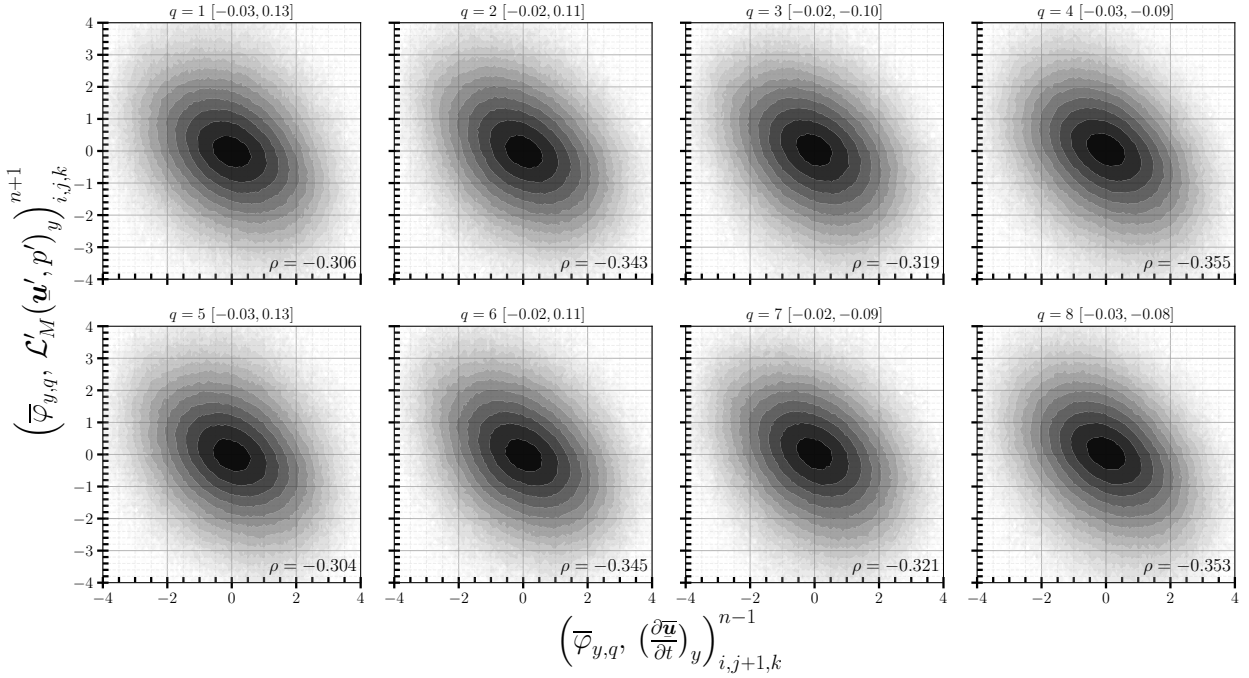


FIGURE A.345 Histograms of the t^{n-1} $(i, j+1, k)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

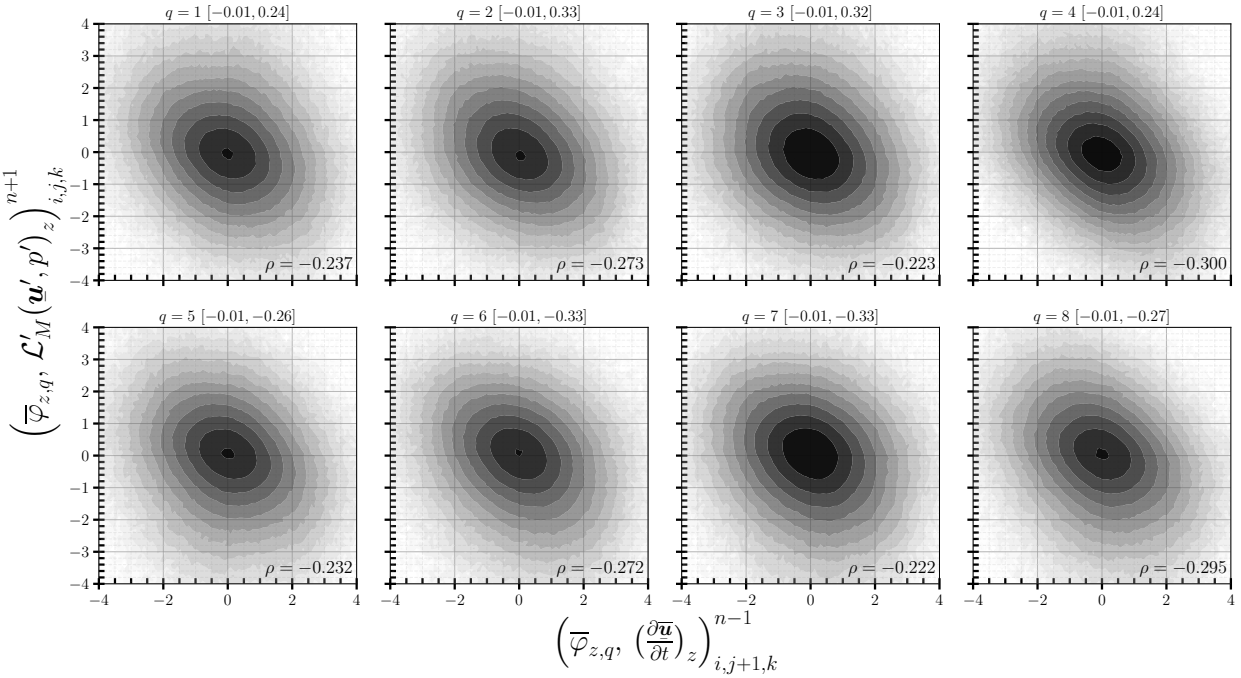


FIGURE A.346 Histograms of the t^{n-1} $(i, j+1, k)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

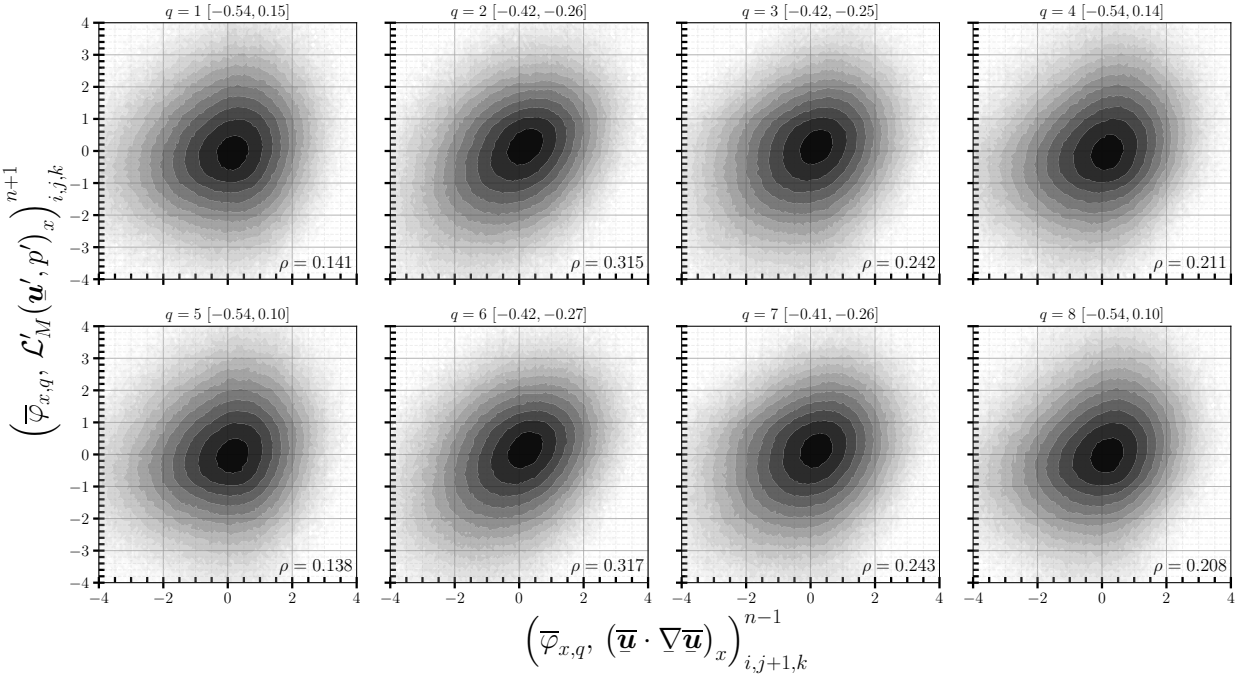


FIGURE A.347 Histograms of the t^{n-1} $(i, j+1, k)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

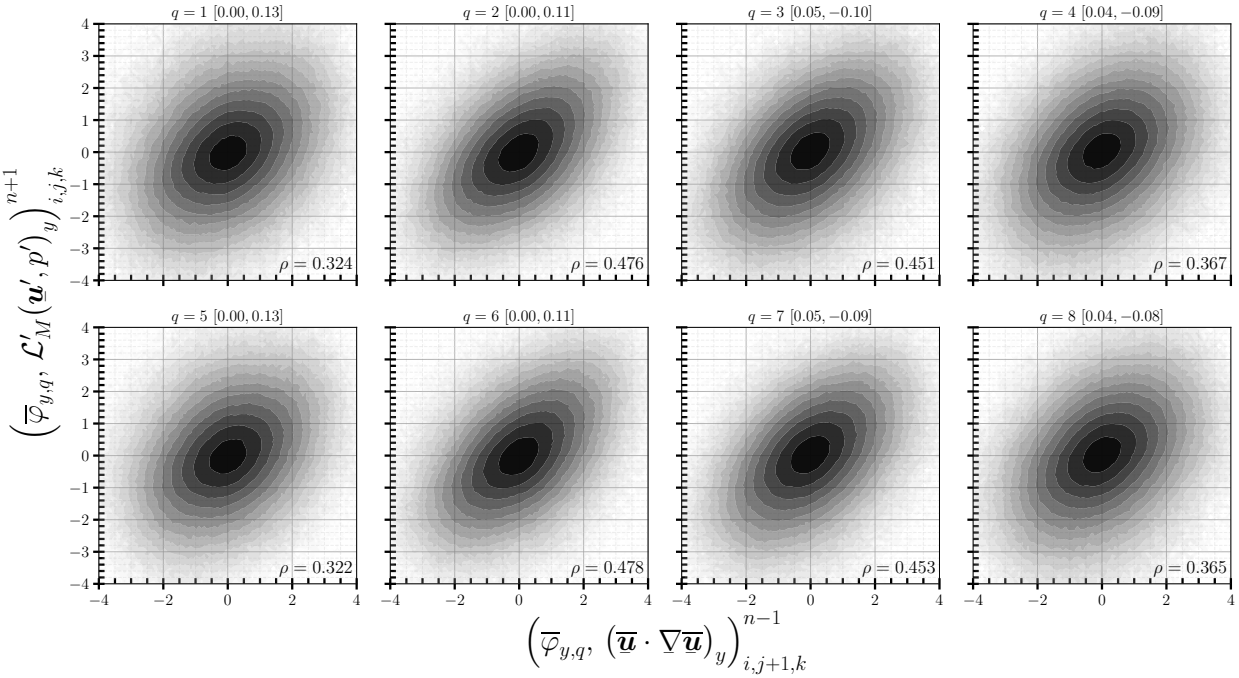


FIGURE A.348 Histograms of the t^{n-1} $(i, j+1, k)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

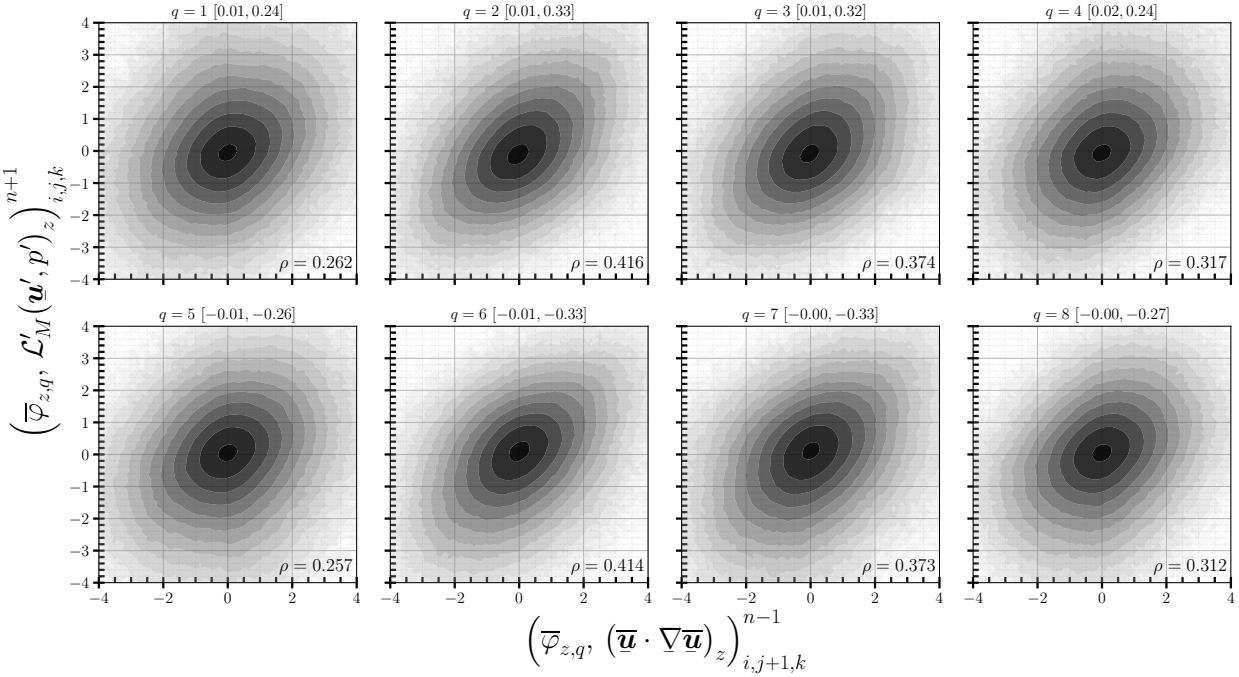


FIGURE A.349 Histograms of the t^{n-1} $(i, j+1, k)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

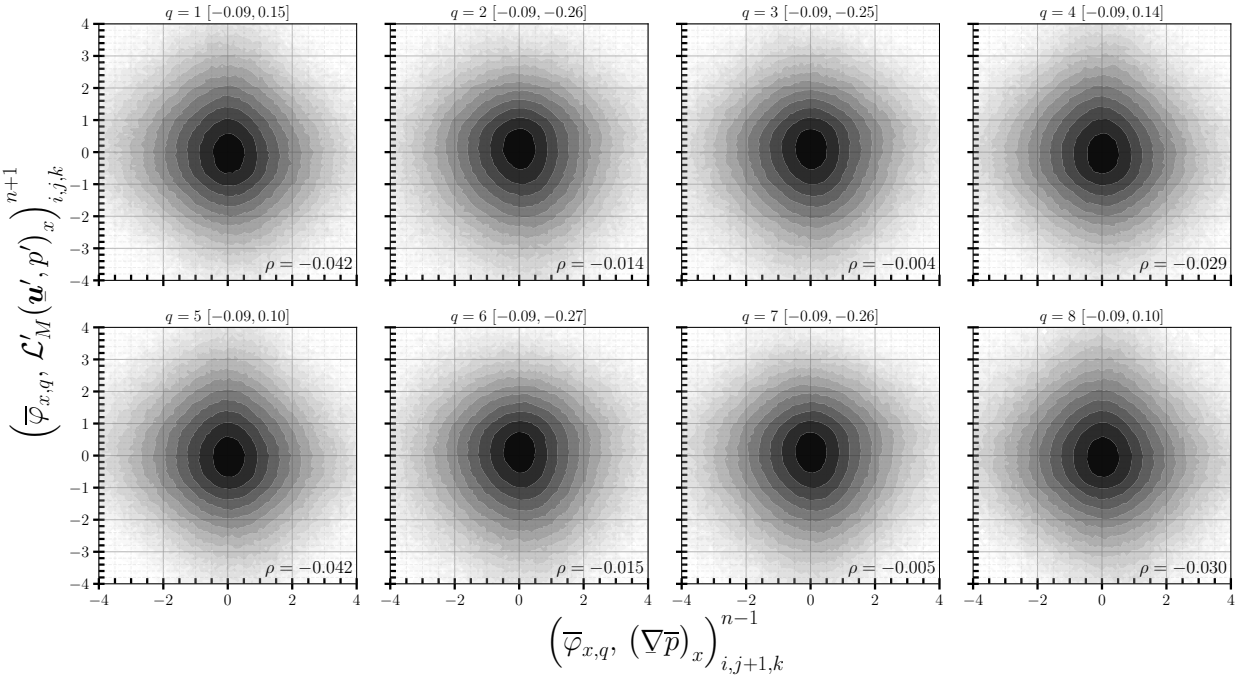


FIGURE A.350 Histograms of the $t^{n-1}\,(i, j+1, k)$-element x-momentum pressure gradient term against the $t^{n+1}\,(i, j, k)$-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

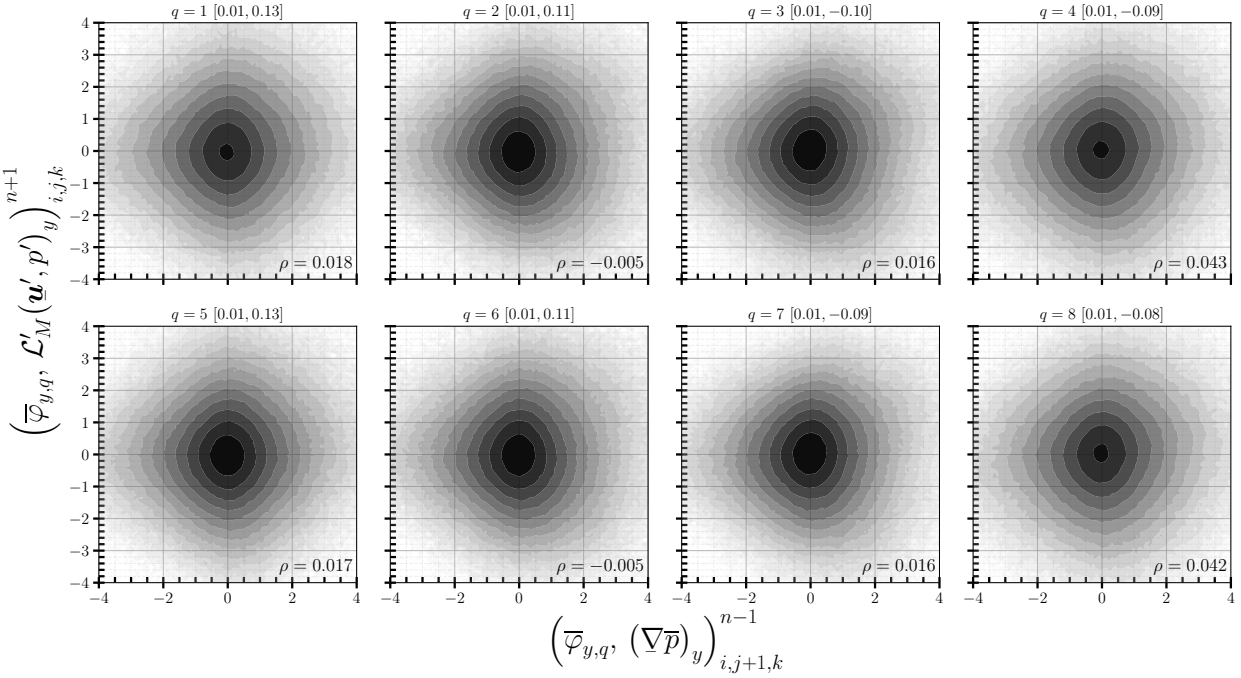


FIGURE A.351 Histograms of the $t^{n-1}\,(i, j+1, k)$-element y-momentum pressure gradient term against the $t^{n+1}\,(i, j, k)$-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

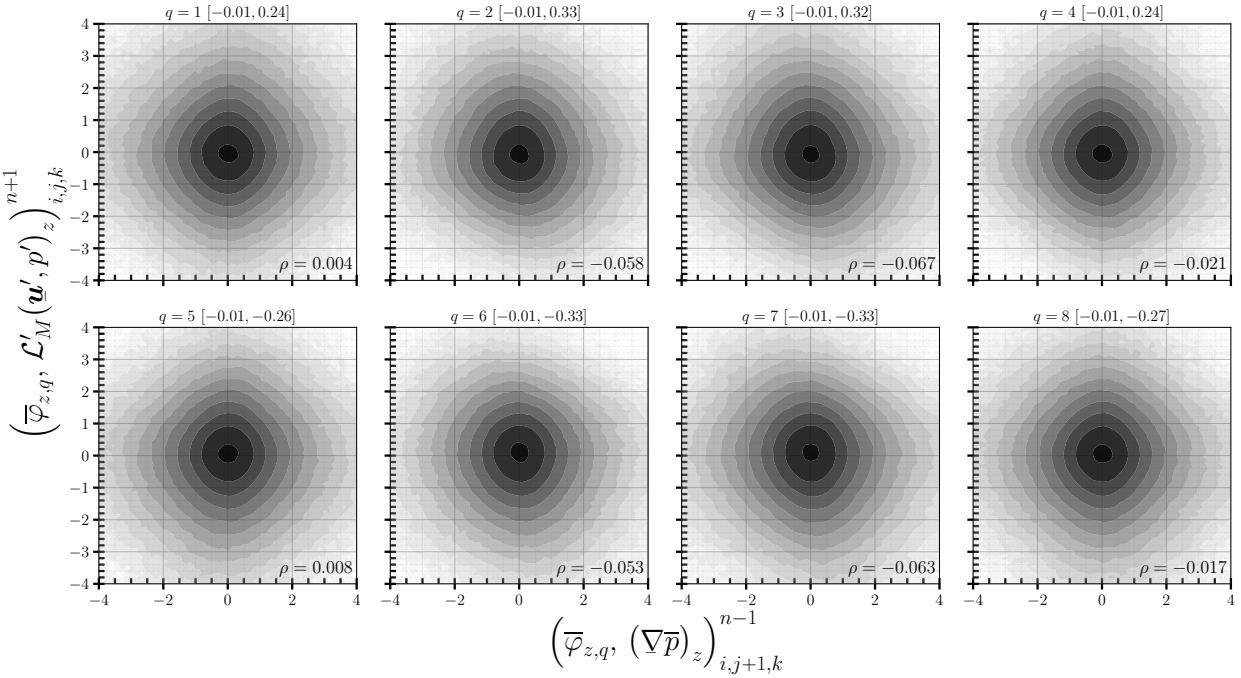


FIGURE A.352 Histograms of the $t^{n-1}\,(i, j+1, k)$-element z-momentum pressure gradient term against the $t^{n+1}\,(i, j, k)$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

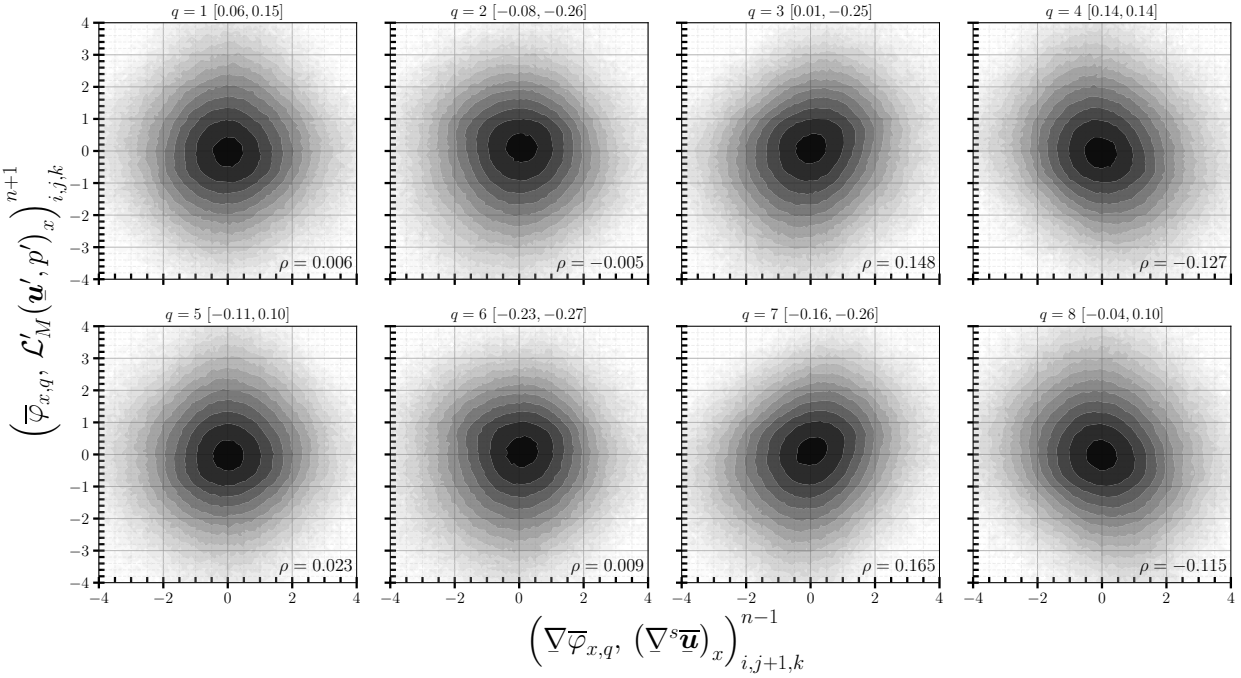


FIGURE A.353 Histograms of the $t^{n-1}\,(i, j+1, k)$-element x-momentum diffusion term against the $t^{n+1}\,(i, j, k)$-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

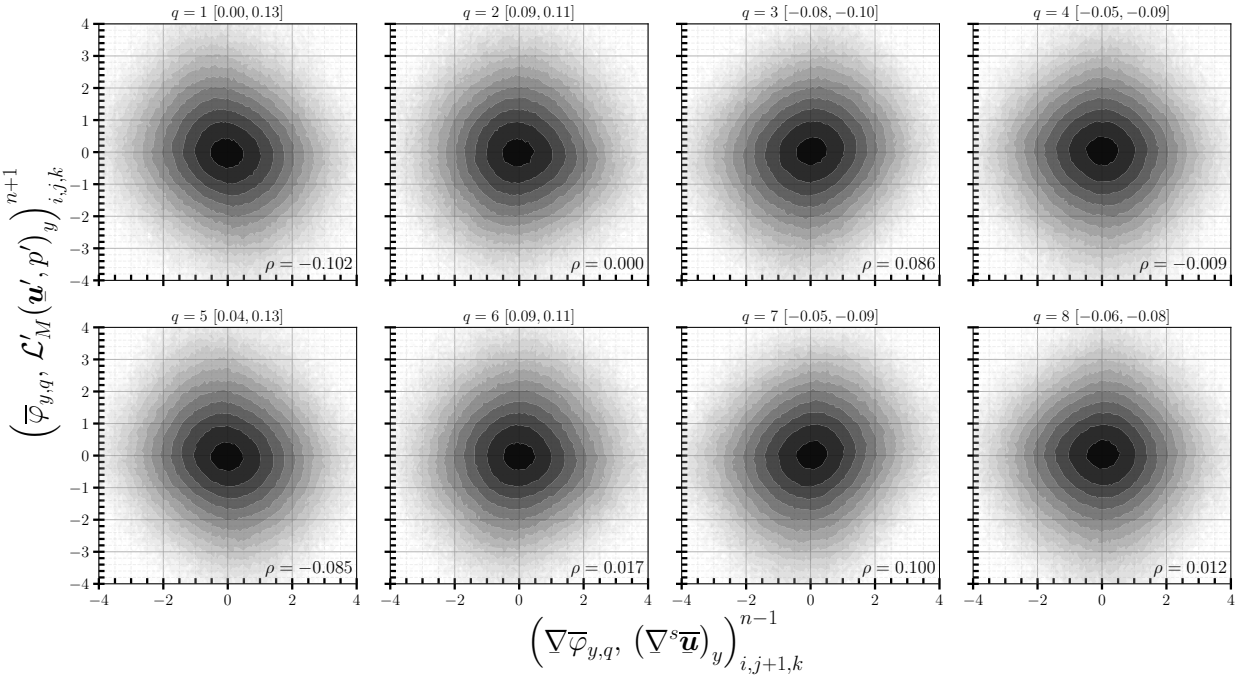


FIGURE A.354 Histograms of the t^{n-1} $(i, j+1, k)$-element y-momentum diffusion term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

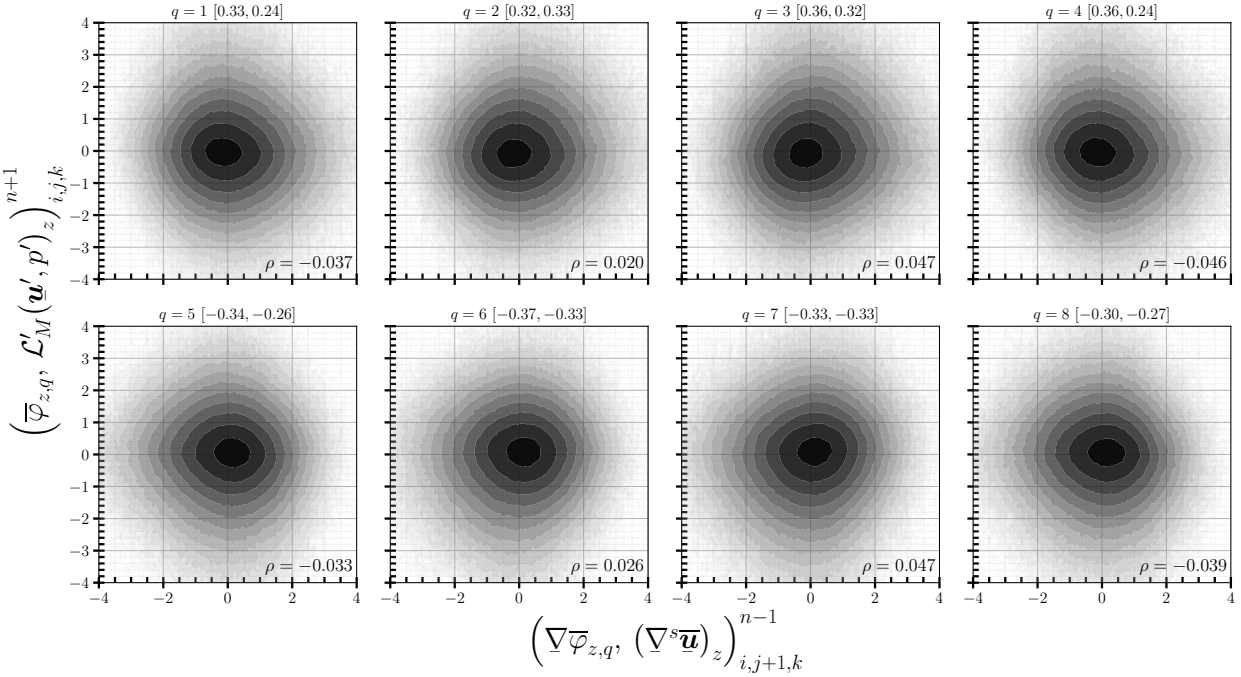


FIGURE A.355 Histograms of the t^{n-1} $(i, j+1, k)$-element z-momentum diffusion term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

A.3.9 $(i, j, k \pm 1)$ Histograms

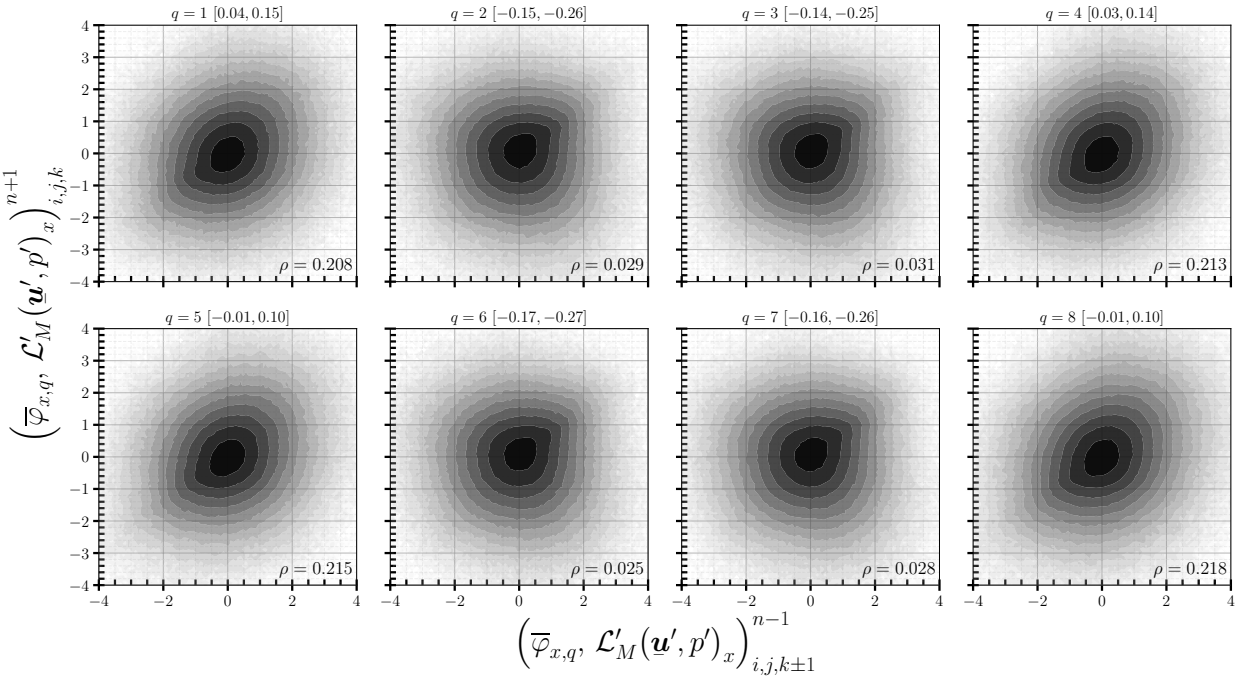


Figure A.356 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum closure term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

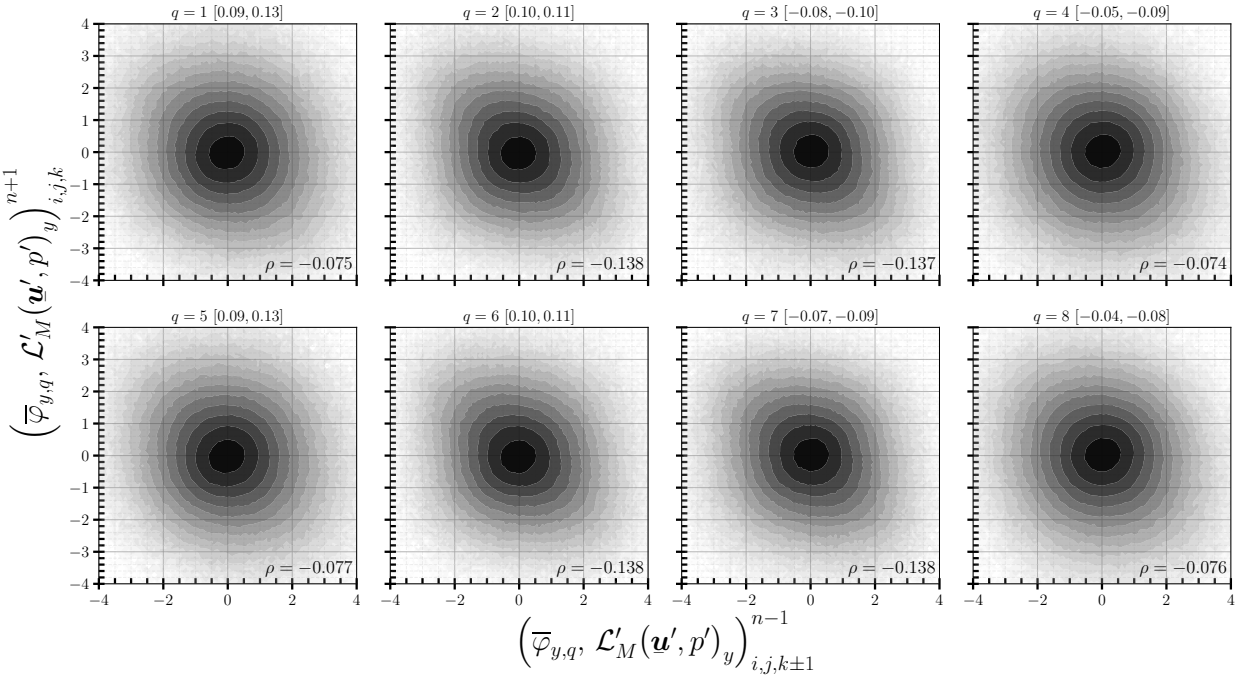


FIGURE A.357 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element y-momentum closure term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

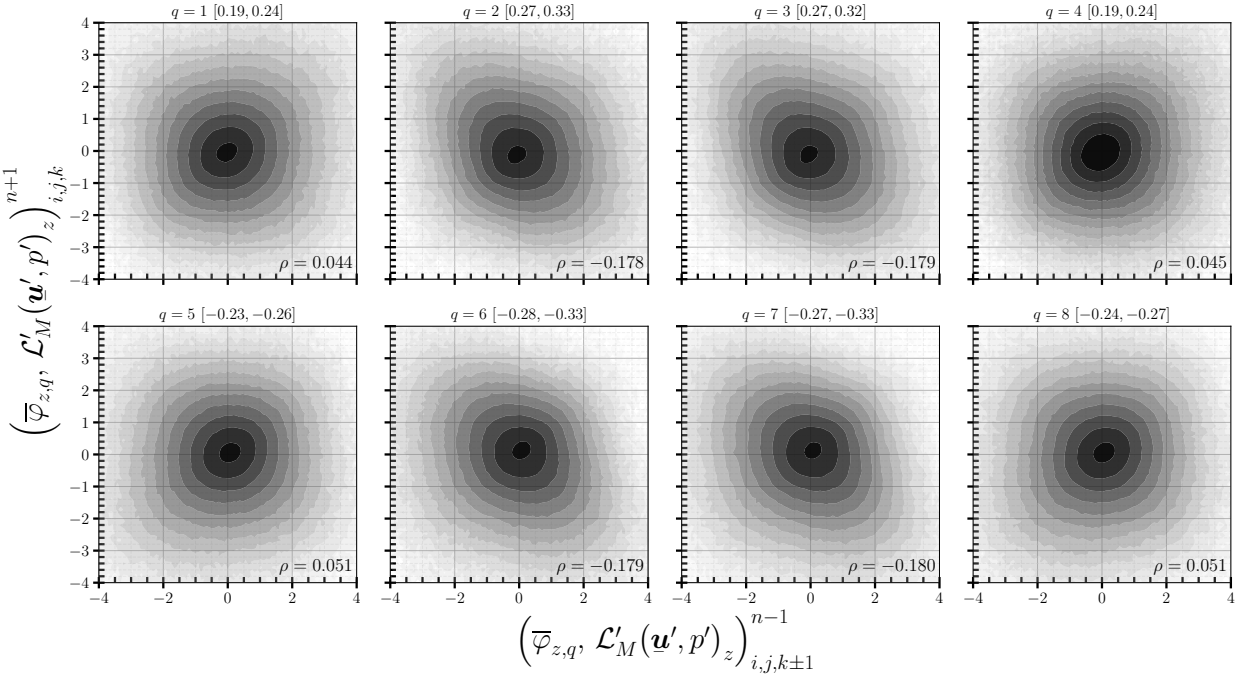


FIGURE A.358 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element z-momentum closure term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

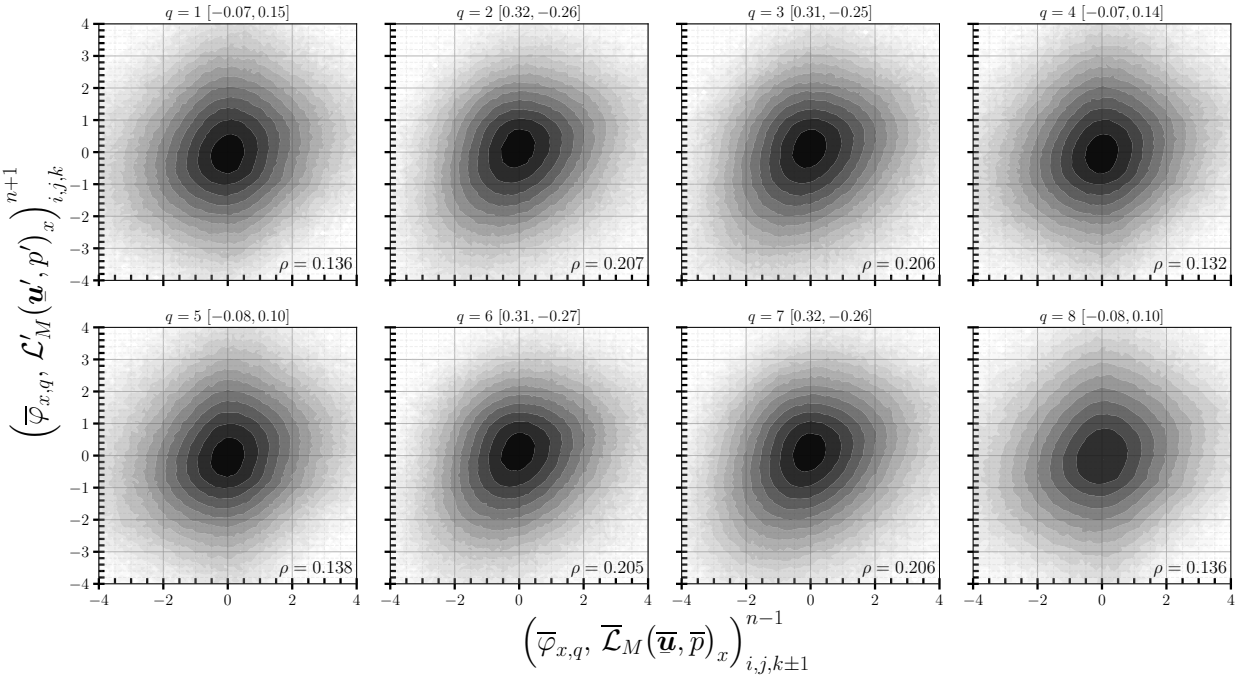


FIGURE A.359 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

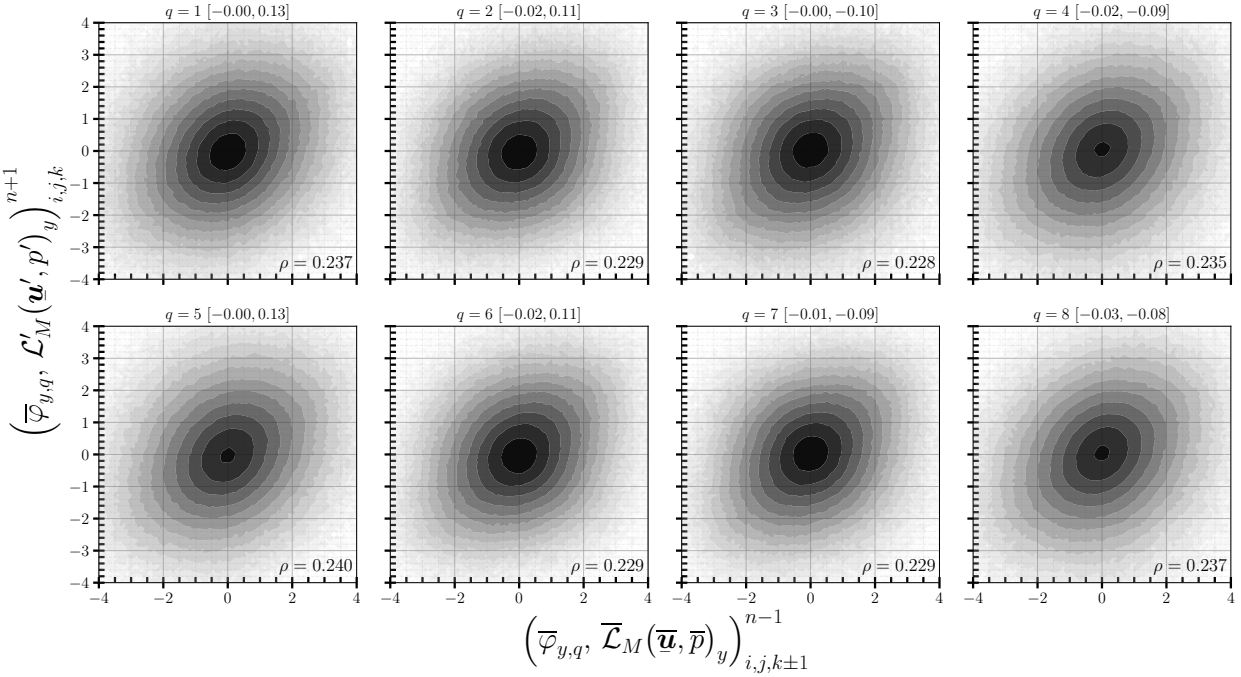


FIGURE A.360 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element y-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

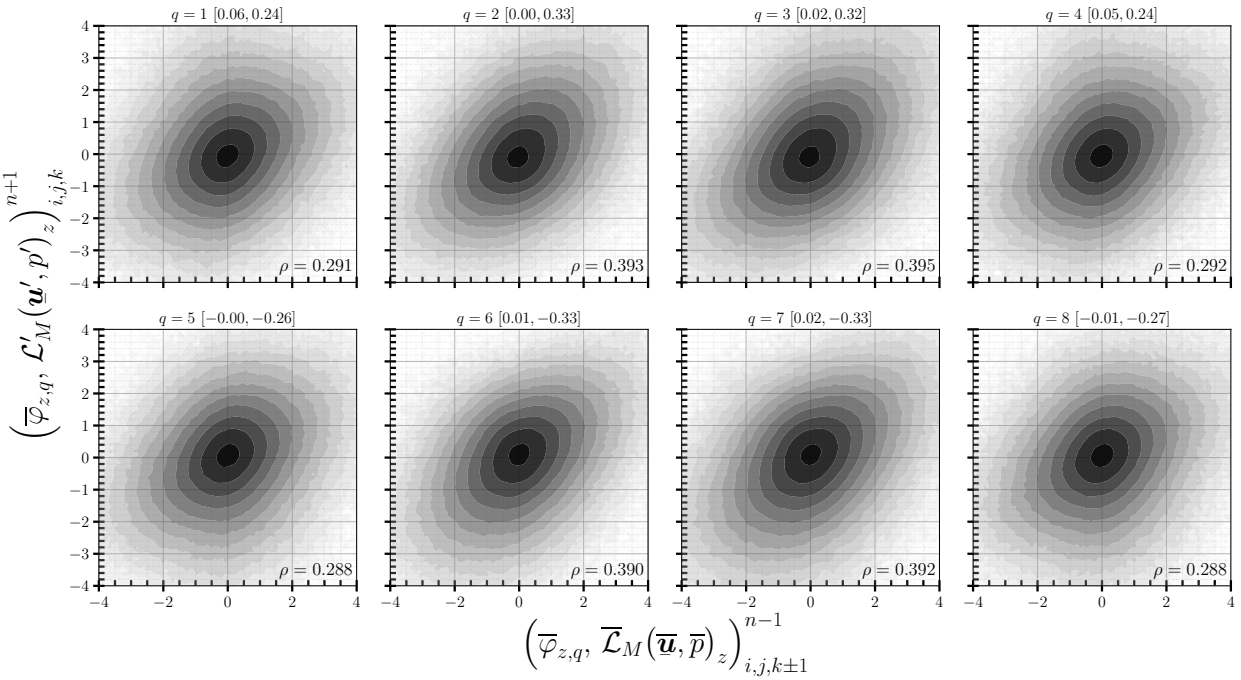


FIGURE A.361 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element z-momentum coarse-scale sum against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

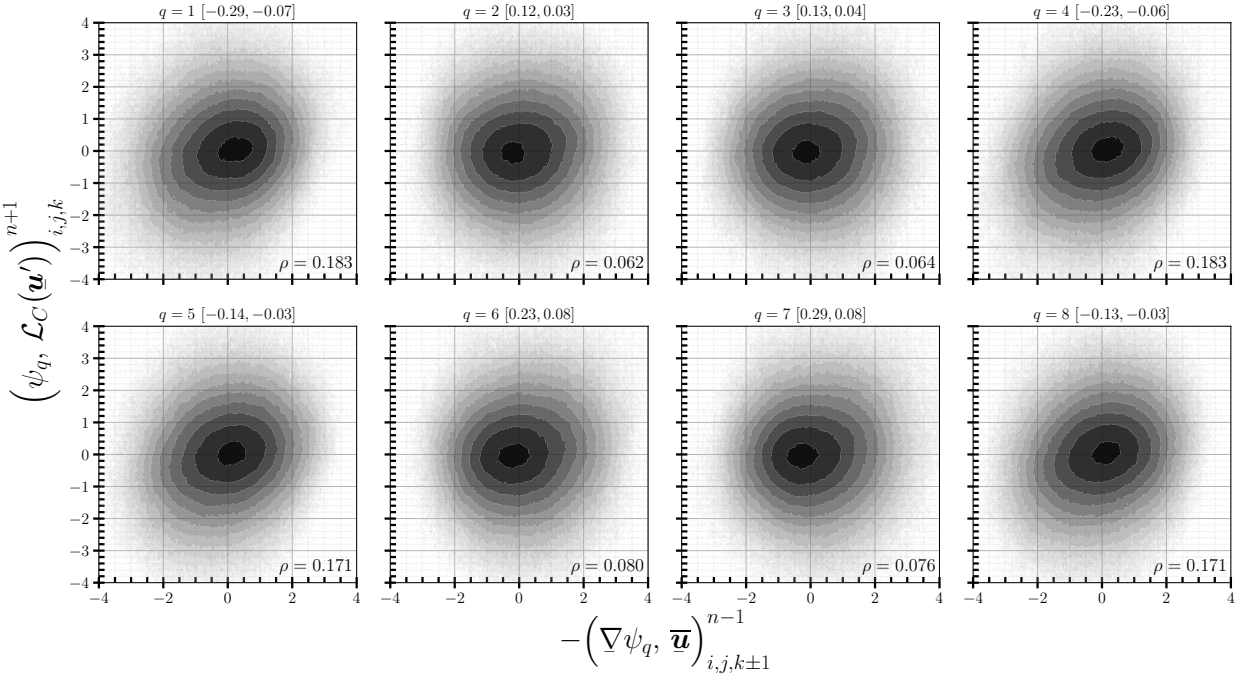


FIGURE A.362 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element continuity coarse-scale sum against the t^{n+1} (i, j, k)-element continuity closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

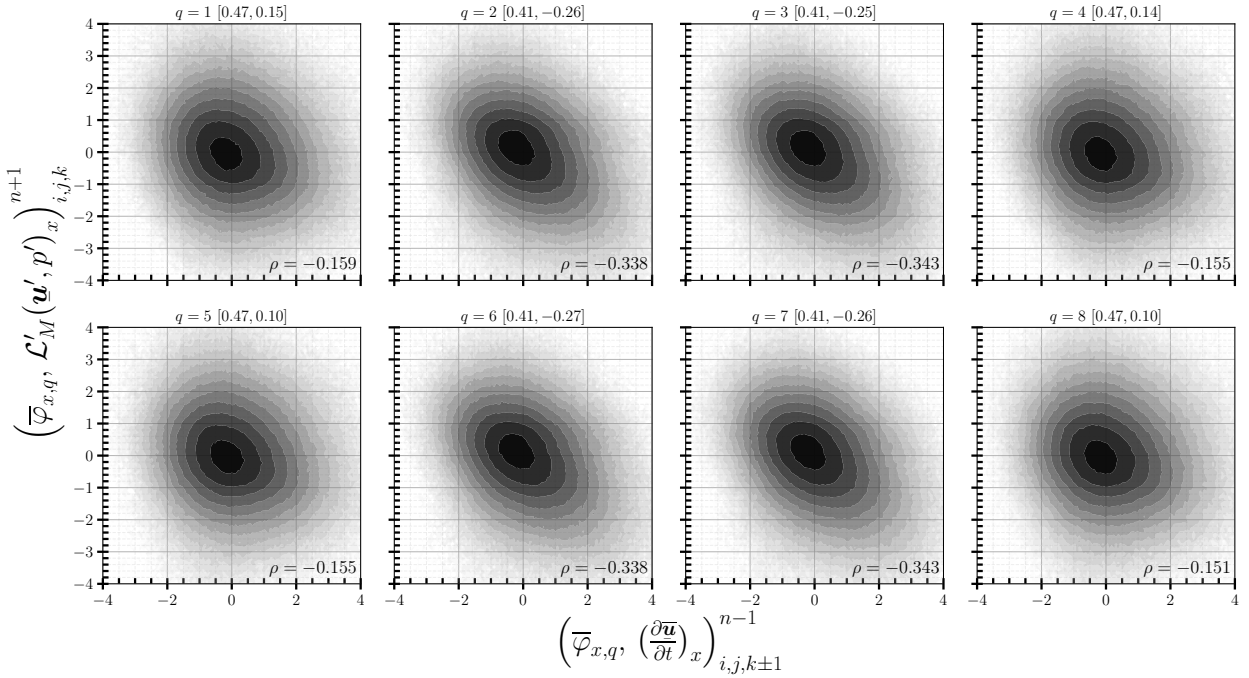


FIGURE A.363 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum time derivative term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

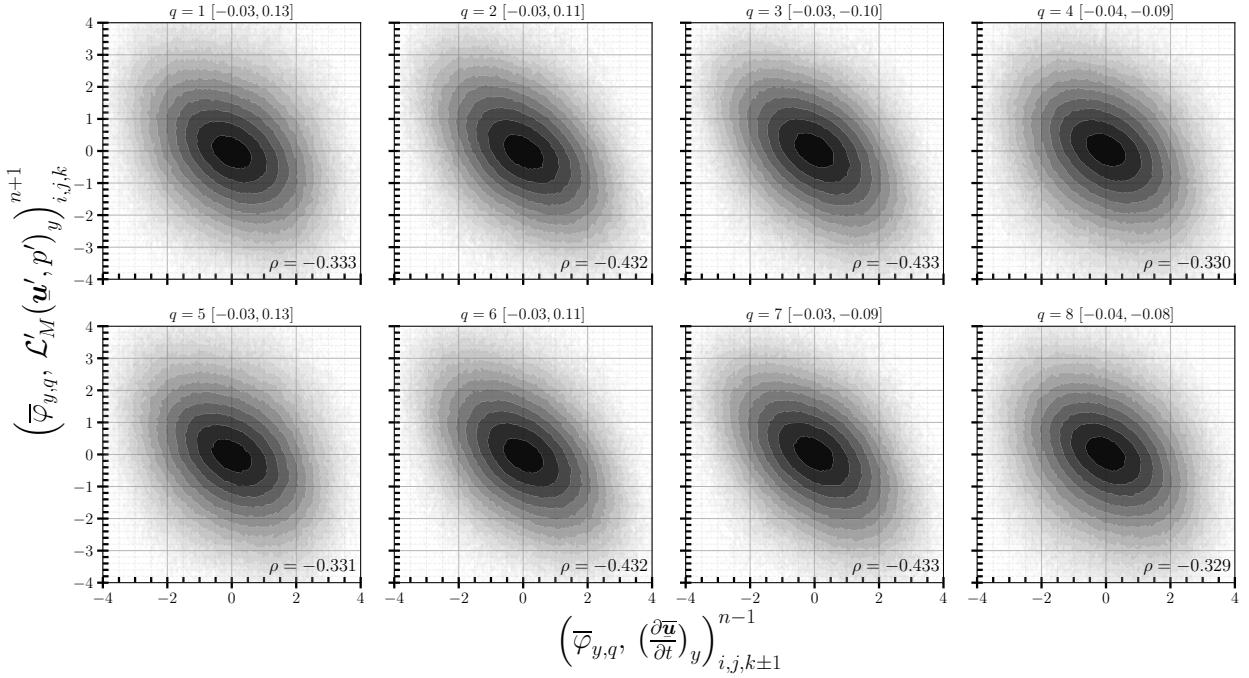


FIGURE A.364 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element y-momentum time derivative term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

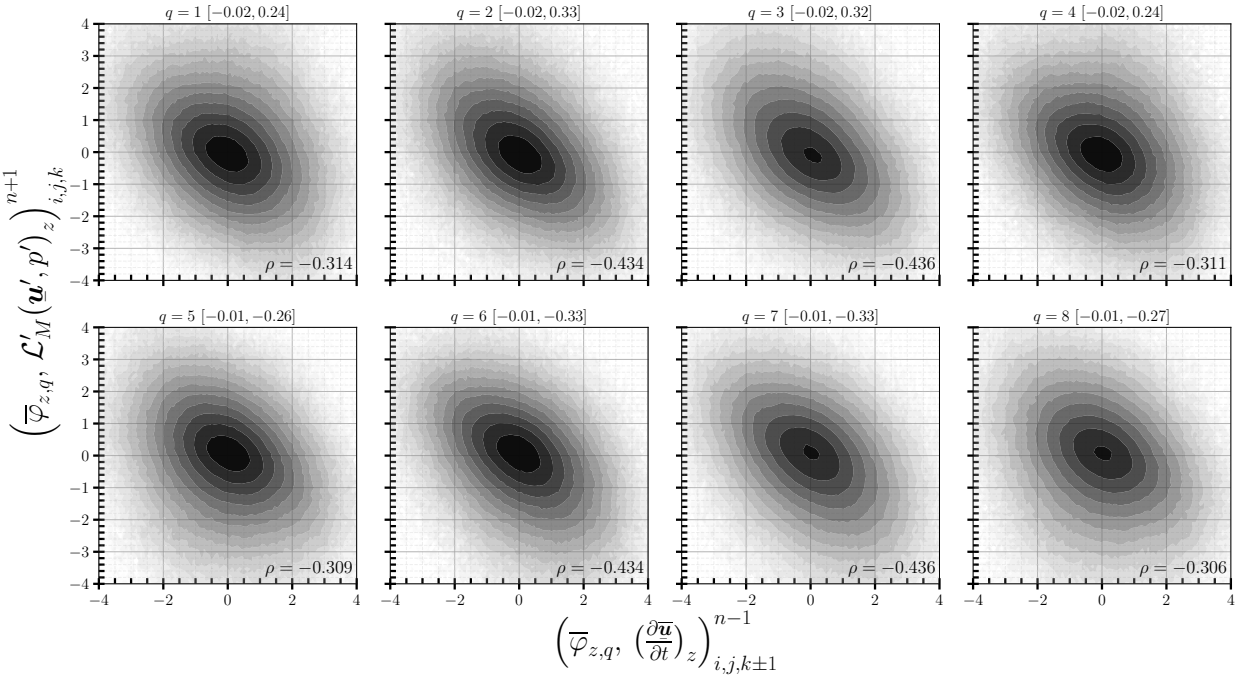


FIGURE A.365 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element z-momentum time derivative term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

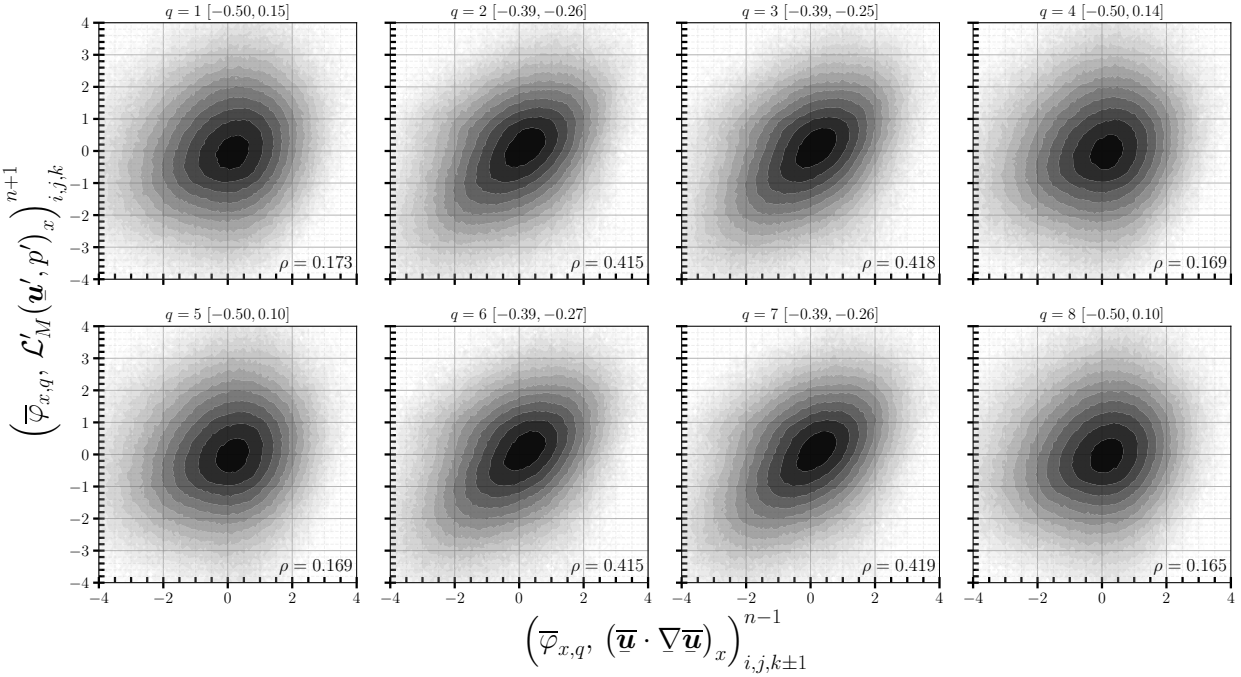


FIGURE A.366 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum convection term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

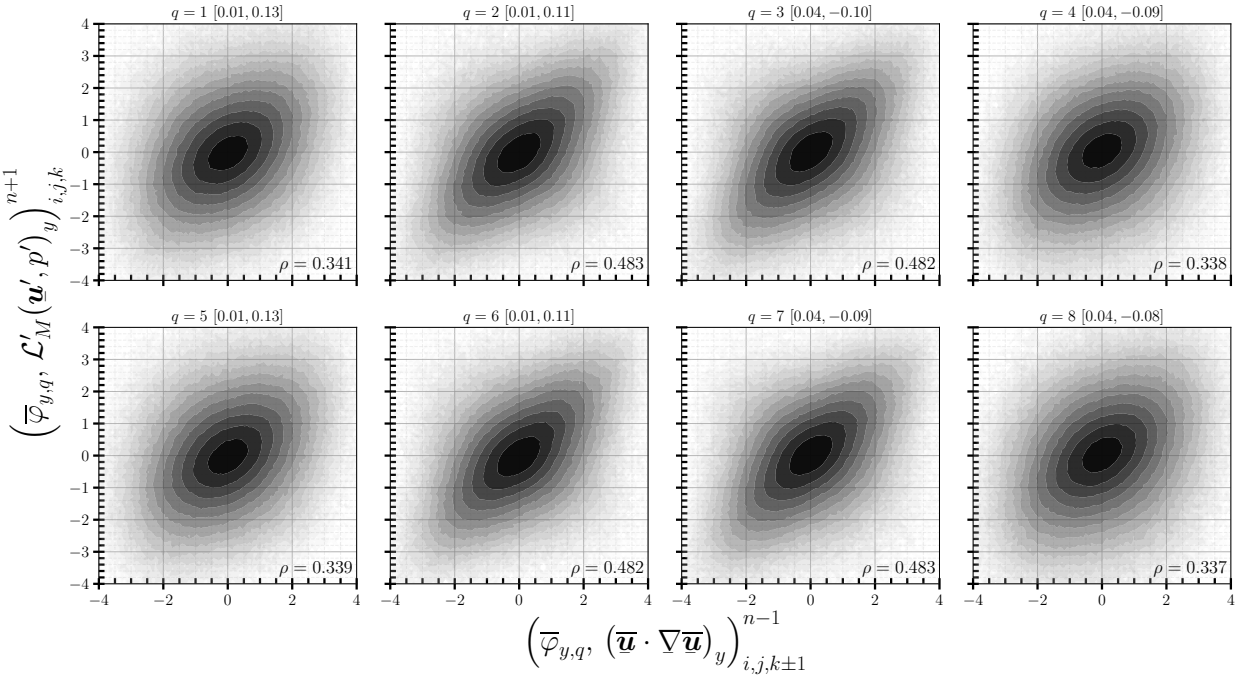


FIGURE A.367 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element y-momentum convection term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

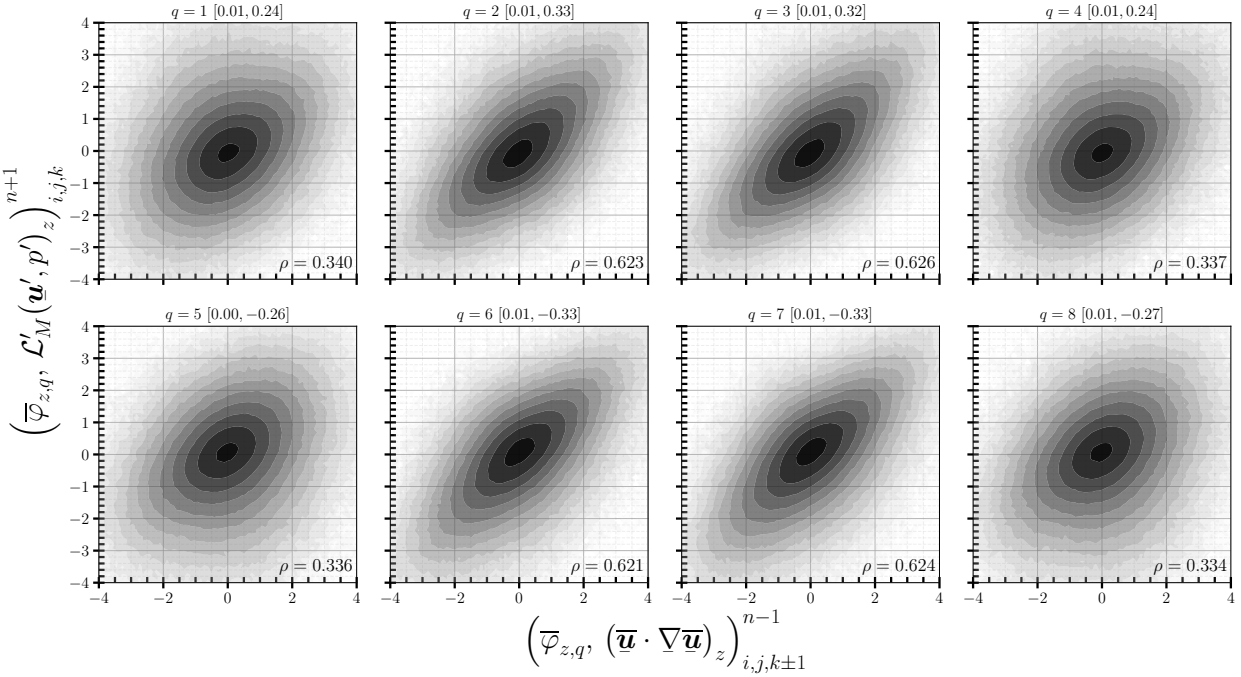


FIGURE A.368 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element z-momentum convection term against the t^{n+1} (i, j, k)-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

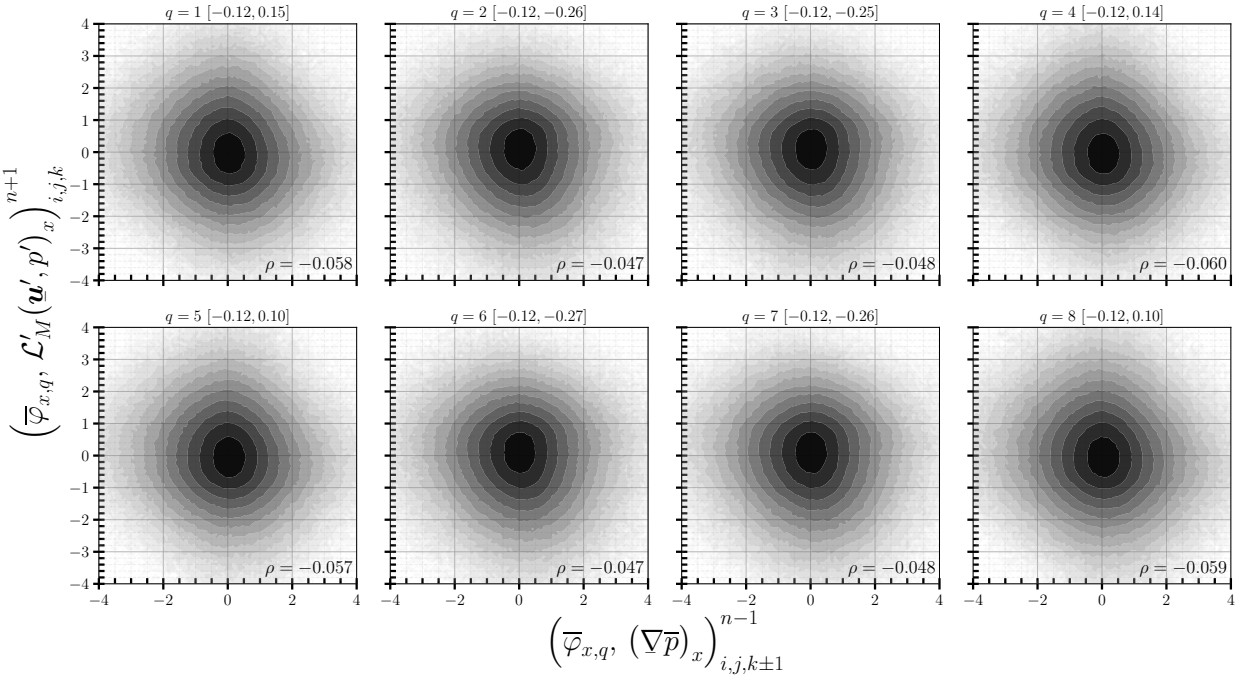


FIGURE A.369 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum pressure gradient term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

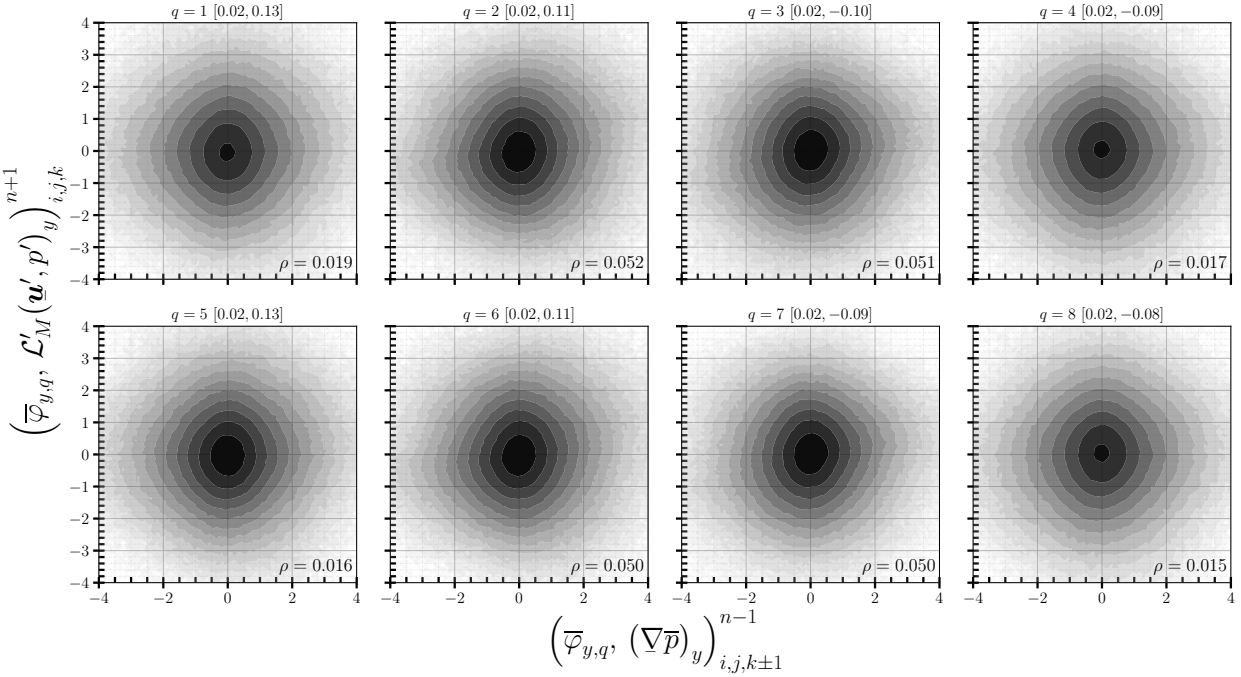


FIGURE A.370 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element y-momentum pressure gradient term against the t^{n+1} (i, j, k)-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

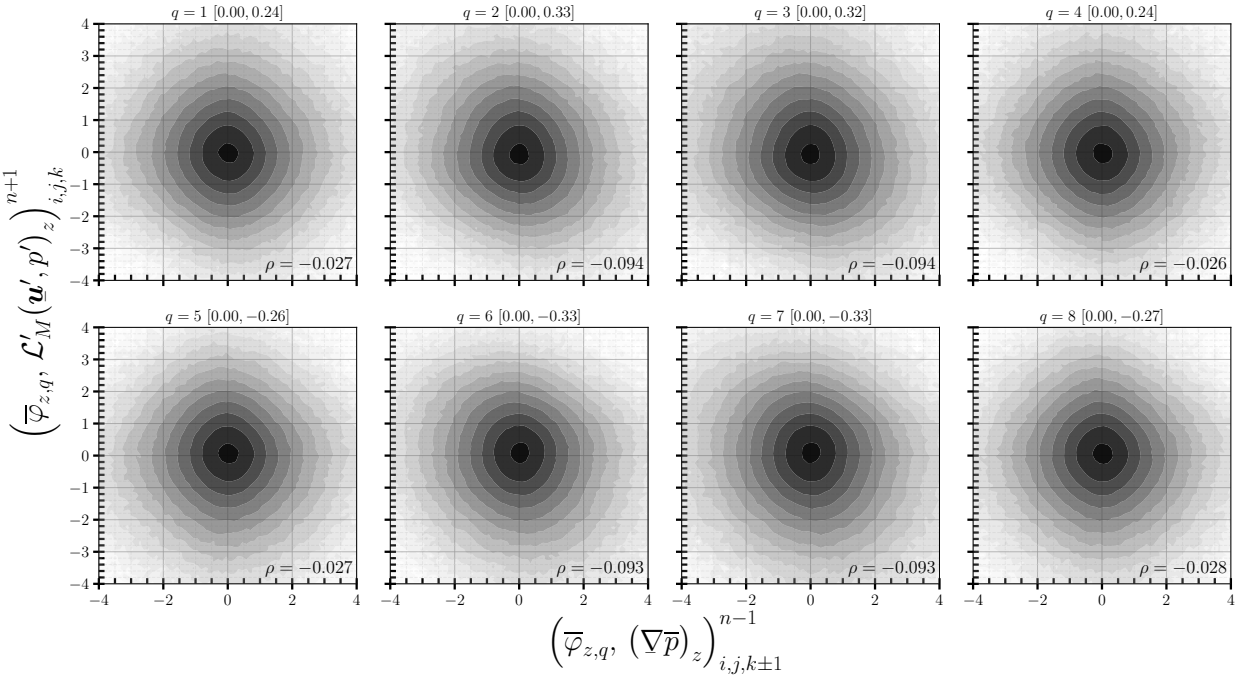


FIGURE A.371 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element *z*-momentum pressure gradient term against the t^{n+1} (i, j, k)-element *z*-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

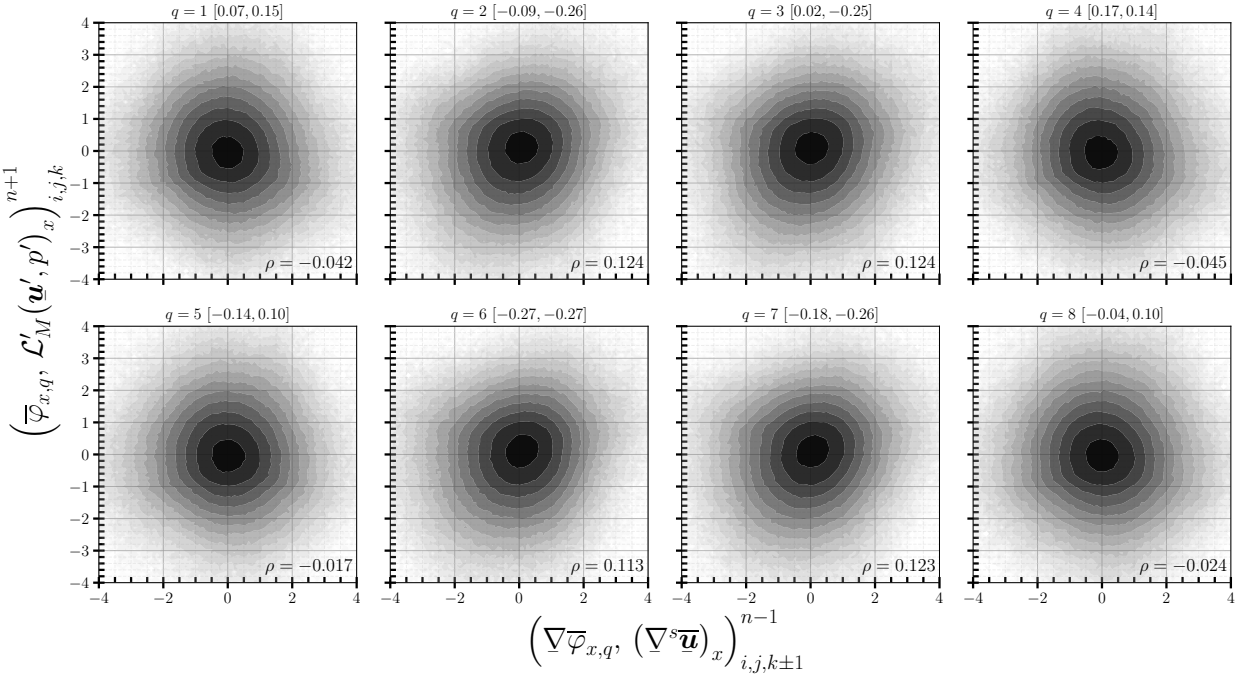


FIGURE A.372 Histograms of the t^{n-1} $(i, j, k \pm 1)$-element x-momentum diffusion term against the t^{n+1} (i, j, k)-element x-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

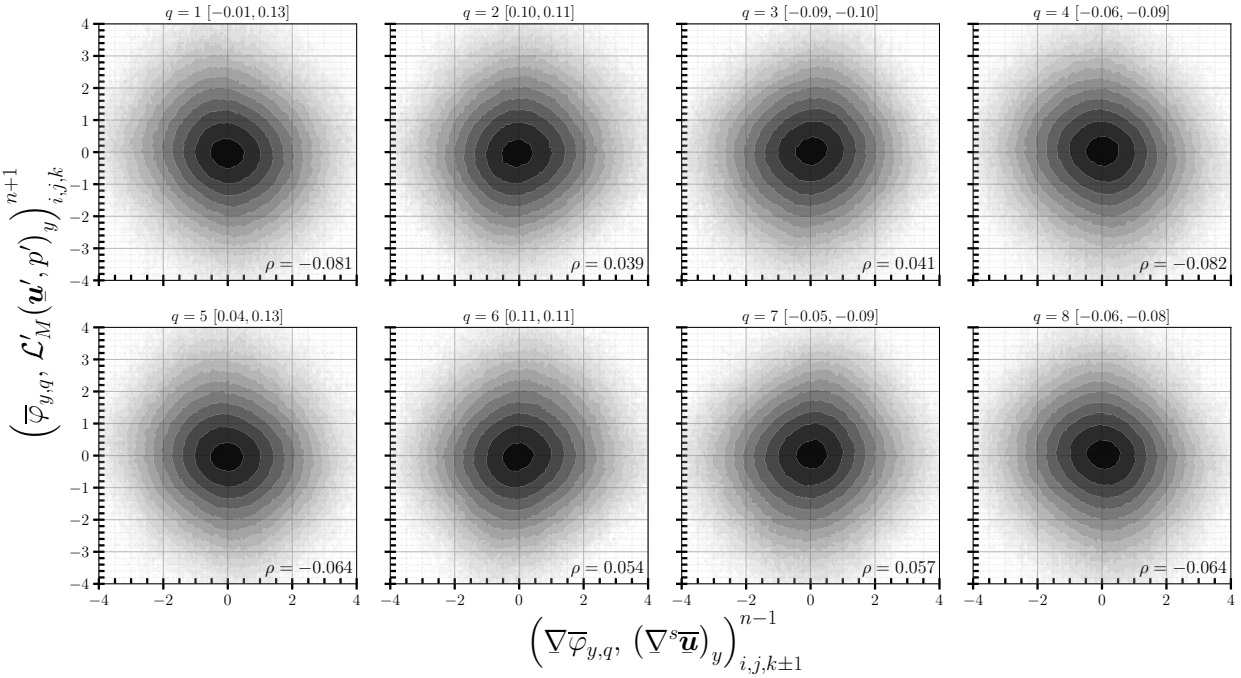


FIGURE A.373 Histograms of the t^{n-1} $(\mathrm{i},\mathrm{j},\mathrm{k}\pm1)$-element y-momentum diffusion term against the t^{n+1} $(\mathrm{i},\mathrm{j},\mathrm{k})$-element y-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

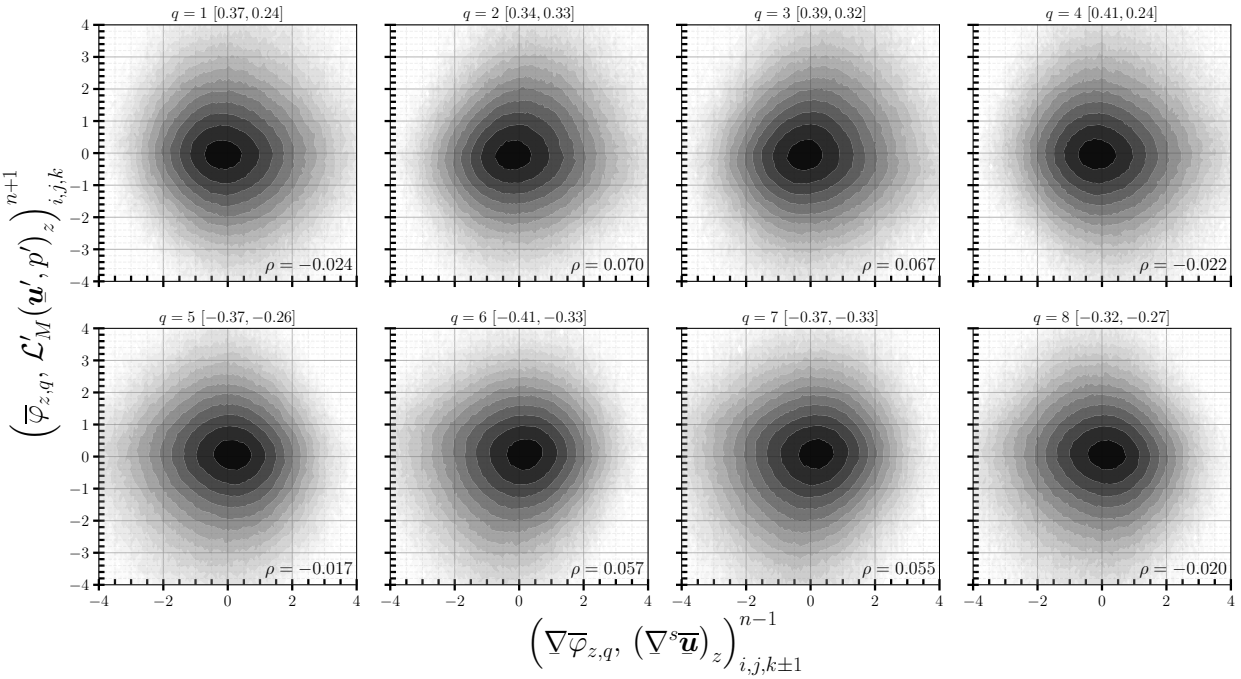
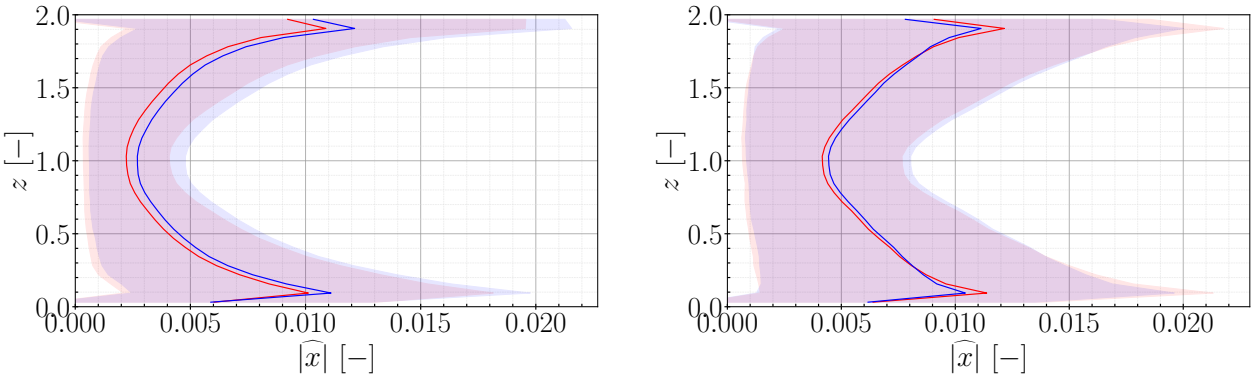


FIGURE A.374 Histograms of the t^{n-1} $(\mathrm{i},\mathrm{j},\mathrm{k}\pm1)$-element z-momentum diffusion term against the t^{n+1} $(\mathrm{i},\mathrm{j},\mathrm{k})$-element z-momentum closure term for each shape function q. Both data sets are normalised according to each k-level mean and standard deviation.

Appendix B Supplementary Figures

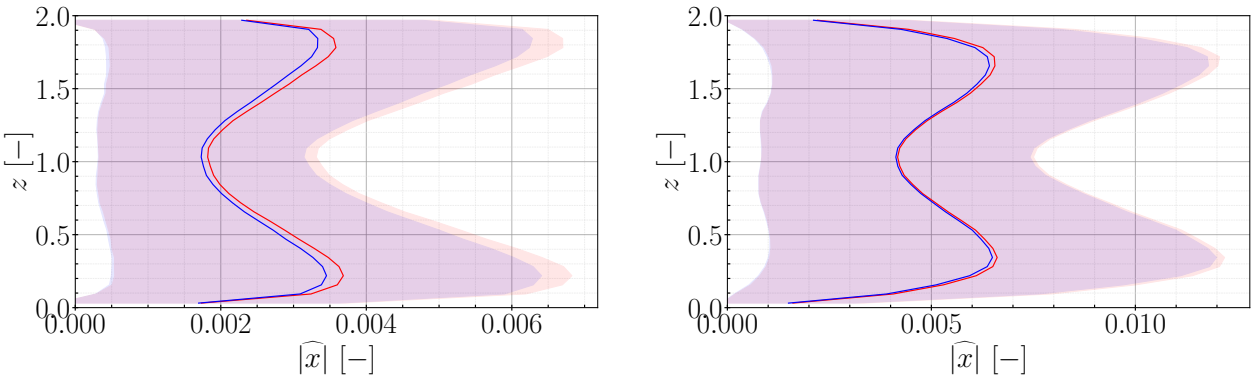
B.1 A priori Evaluation 261

B.1 A priori Evaluation



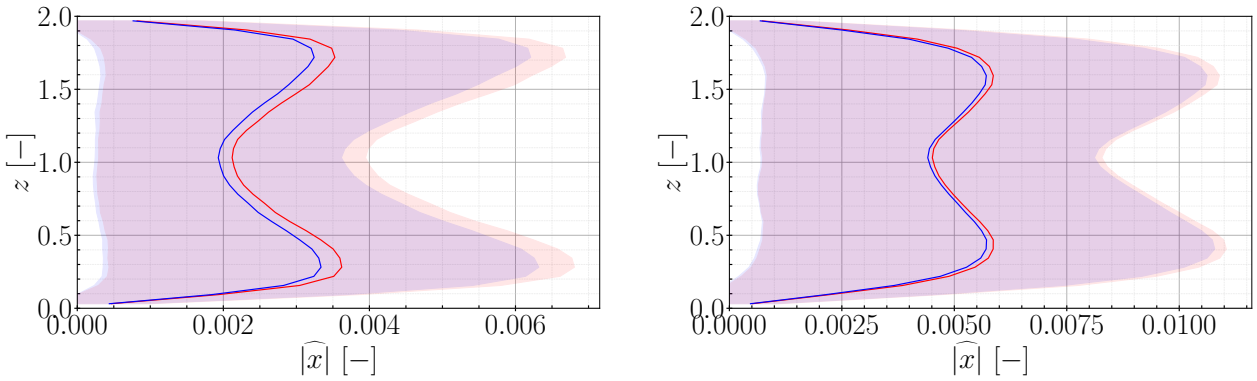
(A) q = 1 network **(B)** q = 2 network

FIGURE B.1 Channel profile of the mean x-momentum closure term. Shaded region denotes ±1 standard deviation. — Truth, — Prediction



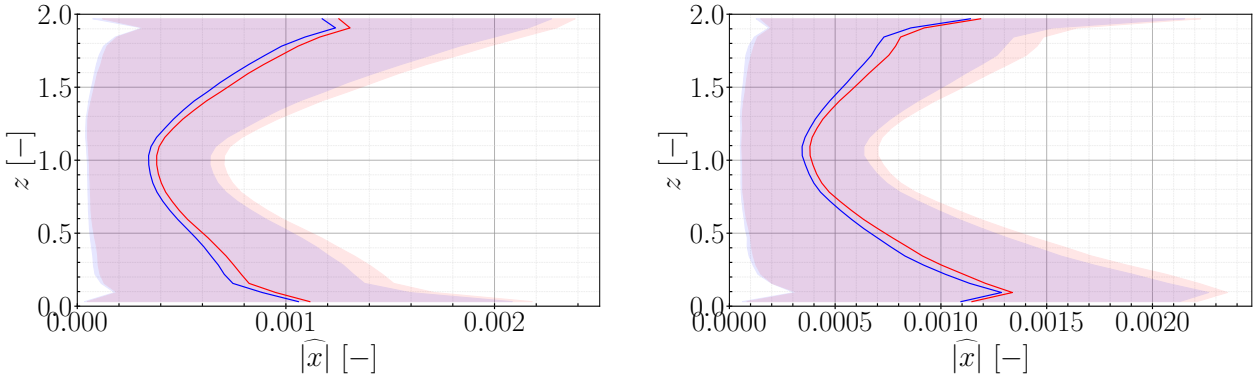
(A) q = 1 network **(B)** q = 2 network

FIGURE B.2 Channel profile of the mean y-momentum closure term. Shaded region denotes ±1 standard deviation. — Truth, — Prediction



(A) q = 1 network **(B)** q = 2 network

FIGURE B.3 Channel profile of the mean z-momentum closure term. Shaded region denotes ±1 standard deviation. — Truth, — Prediction



(A) q = 1 network **(B)** q = 2 network

FIGURE B.4 Channel profile of the mean continuity closure term. Shaded region denotes ±1 standard deviation. — Truth, — Prediction