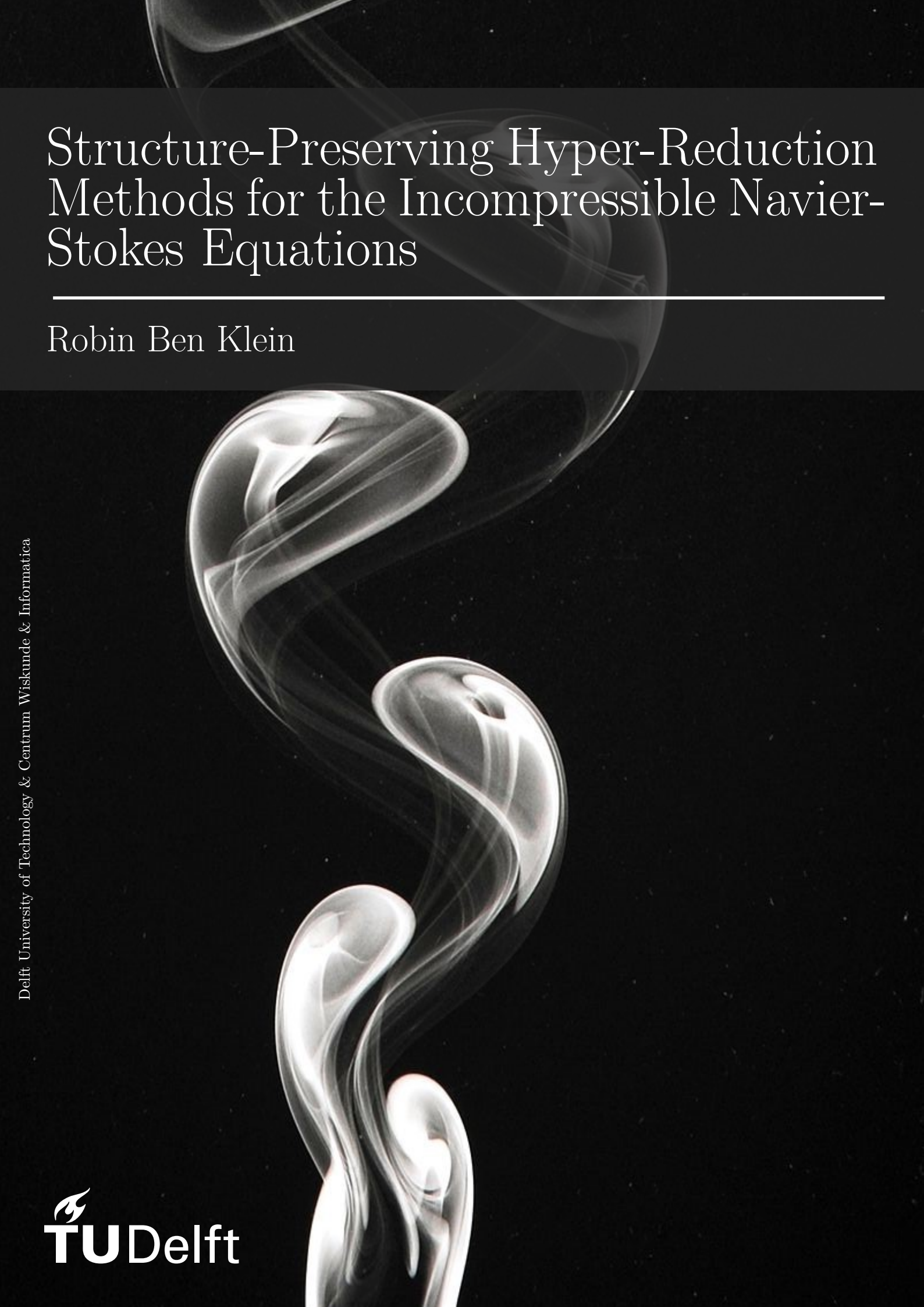
Structure-Preserving Hyper-Reduction Methods for the Incompressible Navier-Stokes Equations

Robin Ben Klein



Delft University of Technology & Centrum Wiskunde & Informatica

TU Delft

Structure-Preserving Hyper-Reduction Methods for the Incompressible Navier-Stokes Equations

by

Robin Ben Klein

to obtain the degree of Master of Science
at the Delft University of Technology,
presented and defended on Tuesday August 30, 2022 at 1:30 PM.

Student number:	4560884	
Project duration:	September 15, 2021 – July 15, 2022	
Thesis committee:	Prof. dr. ir. R. A. W. M. Henkes,	TU Delft, supervisor
	Dr. ir. B. Sanderse,	Centrum Wiskunde & Informatica, supervisor
	Prof. dr. ir. B.J. Boersma,	TU Delft
	Dr. C. Pagliantini,	TU Eindhoven

TU Delft

Acknowledgements

The completion of this thesis marks the nearing end of my time as a student; a great moment to reflect on the people in my life that I can be grateful to. My time as a student has truly been a valuable addition to my life and me as a person. At times studying has made me completely euphoric and at times it has made me utterly frustrated (many books have had the honor of exploring the fluid dynamics of the air in my room and the solid mechanics of its walls). It has allowed me to explore interests that I am willing to dedicate my life to and it has provided me with knowledge and skills that will allow me to contribute my part in making this world a better place. While studying I have had the pleasure of meeting so many amazing and great people with whom I hope I can spend the years to come. I have also lost people with whom I would have loved to share the moment of officially completing this time of my life, people who have nonetheless supported me countless of hours in the form of smiling pictures on my desk.

Regarding this thesis, there are two people that deserve my thanks most of all and those people are Benjamin and Mr. Henkes. I still remember the mail I sent you Benjamin in January 2021, applying for a PhD vacancy at CWI some twenty months before even writing the last few words of my master thesis. I will confess now that I knew it was a hopeless effort, but I figured it was worth a try anyway. Little did I know that taking this chance would lead to us setting up this master thesis project together. For that I would like to thank you. I would also like to thank you for all the time you spent helping me, giving me advice, reading through all the documents I sent you, reacting to the endless number of confusing Slack messages I sent you (even at moments you did not have to) and bearing with me when, against your sensible advice, I decided to program my own fluid dynamics solver in C++. All the time you took for me while being the father of a young family makes that all the more impressive, I want to wish you all the best with that. Mr. Henkes, I would like to thank you for all the effort you have put in helping me get the best possible result out of this master thesis project. Knowing of some of the endeavours you are involved in and of how many other people need your help, the readiness you had to help me whenever I needed that has amazed me. I hope I can spend more time working with both of you in the future.

I would also like to thank my friends with whom I share(d) my house, my time rowing and being a student and, last but not least, my time in my hometown Alphen aan den Rijn, for helping me not to forget there is a life outside of writing a thesis! Furthermore, I would like to thank the staff of the scientific computing department of CWI for all the great conversations about mathematics and for those not about mathematics. Being a mechanical engineering student living in a city full of economics and management students it has been difficult at times to find people that share my passion for mathematics, getting to share this passion with everyone at CWI has been a great experience.

Finally, I want to thank my family. I do not think I can put into words how thankful I am of you, not only for supporting me while writing this master thesis but also for giving me the chance to study in the first place. Without you, this would not have been possible. You read these words sometimes in other people's theses, but they only make sense after you finish one yourself. Writing out my second name, the name of my father, fully on the cover of this thesis I hope I can express some of the part you have played in me finishing this project.

Robin Ben Klein
Delft, August 2022

Abstract

The use of computational fluid dynamics (CFD) in modern engineering applications has greatly increased in recent years. However, many applications of a 'real-time' or 'multi-query' nature still pose a prohibitive computational cost for large scale simulations. Reduced order models (ROMs) have been proposed to alleviate this cost by finding approximate low-dimensional representations of fluid dynamical simulations that are cheaper to evaluate. This master thesis deals with ROMs of the POD-Galerkin kind, where POD stands for proper orthogonal decomposition and Galerkin refers to Galerkin projection.

An issue of increasing interest in projection-based reduced order modelling of conservation laws is the preservation of the conservative structure underlying such equations at the reduced level. A non-linearly stable POD-Galerkin ROM of the incompressible Navier-Stokes equations that globally conserves kinetic energy (in the inviscid limit), momentum and mass on periodic domains was constructed by Sanderse (2020). The quadratic nonlinearity in the convection operator was dealt with using an exact tensor decomposition to eliminate the dependence of the computational scaling of the ROM on the full order model (FOM) dimensions. However, such a cubic tensor decomposition is not always feasible: in case many POD modes are required (cases with slow Kolmogorov N-width decay), the exact decomposition becomes prohibitively expensive. One possible solution is the use of hyper-reduction methods such as the discrete empirical interpolation method (DEIM). The DEIM generally does not retain the conservative structure of the ROM to which it is applied and, as a consequence, non-linear stability of the ROM of the incompressible Navier-Stokes equations is no longer guaranteed.

In this master thesis several novel DEIM formulations are proposed that enable the construction of non-linearly stable hyper-reduced order models (hROMs) of the incompressible Navier-Stokes equations. The hROMs have the same mass, momentum and energy conservation properties as the previously proposed ROM, but they do not suffer of prohibitively expensive computational scaling when the number of POD modes is increased. The first of the proposed methods is the least-squares discrete empirical interpolation method (LSDEIM), which is based on a constrained minimization. The second method is the Sherman-Morrisson discrete empirical interpolation method (SMDEIM), which applies a rank-one correction to the conventional DEIM to conserve energy. The third method is the decoupled least-squares discrete empirical interpolation method (DLSDEIM), which is a generalization of the LSDEIM that allows increasing the size of the measurement space. All methods result in structure-preserving DEIM formulations that have an equivalent computational scaling as the conventional DEIM, but provide provably stable, structure-preserving hROMs. Furthermore, the use of the principal interval decomposition (PID) in the construction of the reduced and DEIM spaces is considered to beat the Kolmogorov barrier.

The methods are implemented in hROMs of the incompressible Navier-Stokes equations based on the previously proposed ROM. They are compared in terms of robustness, accuracy and efficiency using a shear layer roll-up as a test case. The best performing structure-preserving DEIM formulation is used in conjunction with the PID in a two-dimensional turbulence test case. This demonstrates the ability of the hROM to reproduce two-dimensional turbulence.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
2DT	Two-dimensional decaying turbulence
CFD	Computational fluid dynamics
DEIM	Discrete empirical interpolation method
DLSDEIM	Decoupled least-squares discrete empirical interpolation method
ECSW	Energy-conserving sampling and weighting
FOM	Full order model
FVM	Finite volume method
GB	Gigabytes
GL4	Gauss-Legendre 4
GMRES	Generalized minimal residual
GNAT	Gauss-Newton with approximated tensors
hROM	Hyper-reduced order model
LSDEIM	Least-squares discrete empirical interpolation method
LU	Lower upper
NaN	Not a number
ODE	Ordinary differential equation
PDE	Partial differential equation
PID	Principal interval decomposition
POD	Proper orthogonal decomposition
RK4	Runge-Kutta 4
ROM	Reduced order model
SLR	Shear layer roll-up
SMDEIM	Sherman-Morrison discrete empirical interpolation method
SPD	Symmetric positive definite
SVD	Singular value decomposition
Tens. Dec.	Tensor decomposition

Symbols

Symbol	Definition
$\langle \cdot, \cdot \rangle$	Euclidean inner product
$\|\|\cdot\|\|$	Euclidean norm
$\|\|\cdot\|\|_F$	Frobenius matrix norm
$\langle \cdot, \cdot \rangle_{L^2(\Omega)}$	L^2-inner product
$\|\|\cdot\|\|_{L^2(\Omega)}$	Norm induced by $L^2(\Omega)$-inner product
$\langle \cdot, \cdot \rangle_\Theta$	Θ-inner product
$\|\|\cdot\|\|_\Theta$	Induced Θ-norm
$\langle \cdot, \cdot \rangle_{\Omega_h}$	Ω_h-inner product
$\|\|\cdot\|\|_{\Omega_h}$	Ω_h-norm
$\|\| \cdot \|\|_\infty$	∞-norm
$(\cdot)^*$	Adjoint of operator $\cdot$
$(\cdot)^\dagger$	Moore-Penrose pseudoinverse of $\cdot$

Symbol	Definition
$\overline{(\cdot)}$	Implicit numerical integration over associated pressure control volume face of $(\cdot)$
$\mathbf{0}$	zero vector
A	Matrix of Runge-Kutta coefficients / Arbitrary matrix / Symmetric LSDEIM matrix (clear from context)
$\boldsymbol{a}_0$	Generalized coordinates of initial condition
$\boldsymbol{a}_c$	Generalized coordinates of convecting reduced velocity
$\boldsymbol{A}_i$	i^{th} Runge-Kutta stage vector of generalized coordinates
a_{ij}	Runge-Kutta coefficient
$\boldsymbol{A}_i^k$	i^{th} Runge-Kutta stage vector of generalized coordinates at k^{th} Newton-Raphson iteration
$\boldsymbol{a}_i(t)$	Generalized coordinates in PID basis i
$\boldsymbol{A}^k$	Block-vector of stage vectors of generalized coordinates at k^{th} Newton-Raphson iteration
$\boldsymbol{a}^{n+1}$	Generalized coordinates at discrete time t^{n+1}
$\boldsymbol{a}(t)$	Generalized coordinates in $\mathcal{V}$
$\boldsymbol{a}_v$	Generalized coordinates of $\boldsymbol{v}$
B	Diagonal matrix of Runge-Kutta coefficients / LSDEIM matrix (clear from context)
$\boldsymbol{b}$	Vector of Runge-Kutta coefficients
$\boldsymbol{b}(\boldsymbol{a})$	LSDEIM short-hand notation
b_i	Runge-Kutta coefficient
$\mathcal{C}$	Continuous convection operator
$\mathcal{C}_{\text{adv}}$	Continuous convection operator advective form
$\mathcal{C}_{\text{div}}$	Continuous convection operator divergence form
CFL	Courant-Friedrichs-Lewy number
$C_h(\boldsymbol{u})$	Discrete convection operator
$\widetilde{C}_h(\boldsymbol{u})$	Quasi-linear discrete convection operator
c_i	Runge-Kutta coefficient
$\boldsymbol{c}_i(\boldsymbol{a}_i)$	DEIM coordinates in PID basis i
$\boldsymbol{c}_m(\boldsymbol{a})$	First m SMDEIM coordinates
$c_{m+1}(\boldsymbol{a})$	Last SMDEIM coordinate
$\boldsymbol{c}_o$	DEIM coordinates at LSDEIM optimum
$C_r(\boldsymbol{a})$	Reduced convection operator
$\widetilde{C}_r(\boldsymbol{a})$	Quasi-linear reduced convection operator
$C_r(\boldsymbol{a} \otimes \boldsymbol{a})$	Reduced convection operator using tensor decomposition
$\mathcal{C}_{\text{skew}}$	Continuous convection operator skew-symmetric form
$\boldsymbol{c}(\boldsymbol{u})$	DEIM coordinates
$C(\boldsymbol{x})$	Equality constraints of a minimization problem
d	Number of spatial dimensions
D	Diagonal matrix with singular values / Arbitrary symmetric matrix (clear from context)
$\mathcal{D}$	Continuous diffusion operator
D_h	Discrete diffusion operator
Diff	Diffusive stability limit
$\partial_k \Omega_i^u$	Face k of control volume Ω_i^u
$d(\mathcal{M}_u^h)$	Kolmogorov N-width
d_r	Rank of X
D_r	Reduced diffusion operator
D_r^i	Reduced diffusion operator in PID basis i
E	Matrix of momentum conserving POD modes
$\mathcal{E}$	Total enstrophy
$\boldsymbol{e}_i$	Zero vector with one at component i
$E(k), E(k,t)$	Continuous 2DT energy spectrum

Symbol	Definition
$E_k(t)$	Discrete 2DT energy spectrum
$\boldsymbol{e}_s$	Vector of ones with length s
$\boldsymbol{e}_u$	Zero vector with ones at u-unknowns
$\boldsymbol{e}_v$	Zero vector with ones at v-unknowns
$F_1(\boldsymbol{A})$	Implicit Runge-Kutta short-hand function
$F_1(\boldsymbol{U})$	Implicit Runge-Kutta short-hand function
$F_2(\boldsymbol{A})$	Implicit Runge-Kutta short-hand function
$F_2(\boldsymbol{U})$	Implicit Runge-Kutta short-hand function
$\mathcal{F}(\boldsymbol{a})$	Feasible set of LSDEIM
$\boldsymbol{f}^b(\boldsymbol{x},t)$	Body stress vector
$\mathcal{F}_C$	Feasible set associated to constraints in C
f_i	Arbitrary vector function evaluated at Runge-Kutta stage i
$\boldsymbol{f}^s(\boldsymbol{x},t)$	Surface stress vector
$f(\boldsymbol{u},t)$	Arbitrary vector function to evolve $\boldsymbol{u}$ in time
$f(\boldsymbol{x})$	Arbitrary objective function for a minimization problem
$\mathcal{G}$	Continuous pressure gradient operator
G_h	Discrete pressure gradient operator
h	Characteristic spatial stepsize
I	Identity tensor/matrix
i_k	Vector index of u_k or v_k where k is compass notation
$\mathcal{I}_k^u(i)$	Index mapping to associated neighbour of Ω_i^u
$\mathcal{I}_k^v(i)$	Index mapping to associated neighbour of Ω_i^v
I_s	$s \times s$ Identity matrix
$J_h(\boldsymbol{u})$	Jacobian of C_h
$J_m(\boldsymbol{a})$	DEIM Jacobian
$J_r(\boldsymbol{a})$	Reduced convection Jacobian
$J_r(I \otimes \boldsymbol{a} + \boldsymbol{a} \otimes I)$	Reduced convection Jacobian using tensor decomposition
$\mathcal{K}$	Total kinetic energy
k	Wavenumber
K_h^n	Discrete total kinetic energy at discrete time t^n
$K_h(t)$	Discrete total kinetic energy
$\widetilde{K}_h(t)$	Energy of Fourier representation of $\tilde{\boldsymbol{u}}(\boldsymbol{x},t)$
$k_{\max}$	Largest wavenumber on discrete domain
K_r	Reduced total kinetic energy
K_r^i	Reduced total kinetic energy in interval i
K_r^n	Reduced total kinetic energy at discrete time t^n
$K(t)$	Total kinetic energy
L	Largest length scales in 2DT flow
l	Length scales in intertial range of 2DT flow
$L^2(\Omega)$	The space of square-integrable functions defined on Ω
L_c	Characteristic length scale
$\mathcal{L}(\boldsymbol{c},\lambda)$	LSDEIM Lagrangian
L_h	Linear Poisson operator
$\mathcal{M}$	Continuous divergence operator
M	Matrix obtained from Runge-Kutta bilinear form / Basis to DEIM-space (clear from context)
m	Number of DEIM space dimensions
$\mathcal{M}_d$	DEIM space
$\mathcal{M}_{\mathcal{F}}$	Feasible LSDEIM approximations
M_h	Discrete divergence operator
$\mathcal{M}_h$	Mesh
M_i	DEIM basis of interval i
m_{ij}	Component of M in row i and column j
M_p	SMDEIM short-hand matrix

Symbol	Definition
m_p	Number of dimensions measurement space
$\mathcal{M}_u^h$	FOM solution manifold
$\mathbb{M}_u^N$	All solution manifolds in N-dimensional state space
N	Total number of velocity unknowns
$\hat{\boldsymbol{n}}_k$	Outward normal vector to face k of a control volume
N_p	Total number of pressure unknowns
n_p	Number of PID intervals
n_s	Number of solution snapshots
N_u	Total number u-unknowns
$N(\boldsymbol{u})$	Arbitrary nonlinear vector operator
N_v	Total number v-unknowns
$\hat{\boldsymbol{n}}(\boldsymbol{x}, t)$	Outward normal vector
N_x	Number of pressure control volumes in x-direction
N_y	Number of pressure control volumes in y-direction
$\mathcal{N}(\mu, \sigma)$	Normal distribution with mean μ and standard deviation σ
N_{Ω^p}	Number of non-overlapping pressure control volumes
P	Pressure Sobolev space / Measurement matrix (clear from context)
$\mathcal{P}_a$	Total palinstrophy
$\mathcal{P}_h$	Measurement space
$\mathcal{P}_h^i$	Measurement space of interval i
$\boldsymbol{p}_h(t)$	Discrete pressure vector
$\boldsymbol{P}_h(t)$	Discrete total momentum vector
$P_h^u(t)$	Discrete total momentum in u-direction
$P_h^v(t)$	Discrete total momentum in v-direction
p_i	Vector index of i^{th} DEIM measurement point
$(P_r^k)^i$	k^{th} Component of reduced total momentum in interval i
$\boldsymbol{P}_r(t)$	Reduced total momentum vector
$P_r^u(t)$	Reduced total momentum in u-direction
$P_r^v(t)$	Reduced total momentum in v-direction
$\boldsymbol{P}(t)$	Total momentum vector
$p(\boldsymbol{x}, t)$	Continuous pressure field
Q_h	Matrix satisfying $D_h = -Q_h^T Q_h$
Q_r	Matrix satisfying $D_r = -Q_r^T Q_r$
r	Number of reduced space dimensions
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive real numbers
$\boldsymbol{r}(\boldsymbol{a})$	SMDEIM short-hand vector
Re	Reynolds number
Re_h	Grid Reynolds number
r_i	Number of dimension in POD basis of interval i
$\boldsymbol{r}(t)$	Residual vector
s	Number of Runge-Kutta stages
$\mathcal{S}_a$	Palinstrophy sink term
$\mathcal{S}(t)$	Surface of $V(t)$
t	Time
T	End-time
$\mathcal{T}$	Characteristic timescale of turbulent eddy
$(\cdot)^T$	Transpose of linear operator $\cdot$
$\mathbb{T}^d$	d-Dimensional torus
t_i	Start-time of PID interval i
t^n	n^{th} discrete time point
T_s	Timescale significantly longer than $\mathcal{T}$
U	Velocity Sobolev space
$\boldsymbol{u}_0$	Initial condition vector

Symbol	Definition
u_C	u-Unknown central to computational stencil
U_c	Characteristic velocity
$\boldsymbol{u}_h^{n+1}$	Discrete velocity vector at discrete time t^{n+1}
$\boldsymbol{u}_h(t)$	Discrete velocity vector
$\boldsymbol{u}_{h,\Delta t=0.001}(t)$	Numerical velocity vector with $\Delta t = 0.001$
$\boldsymbol{U}_j^k$	j^{th} Runge-Kutta stage vector of k^{th} Newton-Raphson iteration
u_k	$k \in \{N, E, S, W\}$ u-unknowns relative to u_C
$\boldsymbol{U}_k$	k^{th} Runge-Kutta stage vector
$\boldsymbol{U}^k$	Runge-Kutta stage block-vector of k^{th} Newton-Raphson iteration
$\boldsymbol{u}_r(t)$	Reduced velocity vector
$\boldsymbol{u}_r^-(t_{i+1})$	Reduced velocity vector of interval i at interface at t_{i+1}
$\boldsymbol{u}_r^+(t_{i+1})$	Reduced velocity vector of interval $i+1$ at interface at t_{i+1}
$\boldsymbol{u}_{TG,h}(t)$	Discrete Taylor-Green solution
$\boldsymbol{u}^{n+1}$	Solution vector at discrete time t^{n+1}
$U(\boldsymbol{x}, t)$	Velocity magnitude field
$\boldsymbol{u}(\boldsymbol{x}, t)$	Continuous velocity field
$\tilde{\boldsymbol{u}}(\boldsymbol{x}, t)$	Continuous Fourier reconstruction of $\boldsymbol{u}_h$
$\mathcal{V}$	Reduced space
$\boldsymbol{v}$	Arbitrary element of $\mathcal{V}$
v_k	$k \in \{NW, NE, SE, SW\}$ v-unknowns relative to u_C
$V(t)$	Material volume
$\boldsymbol{x}$	Continuous spatial coordinate
x	First coordinate direction
X	Solution snapshot data set
$\widetilde{X}$	A remainder of snapshot matrix X
$\widetilde{X}_i$	A remainder of snapshot matrix X_i
x_k	x-Coordinate of u_k or v_k (clear from context)
X_{T_s}	Snapshot matrix of duration T_s
y	Second coordinate direction
y_k	y-Coordinate of u_k or v_k (clear from context)
$\boldsymbol{y}(t)$	Fluid particle trajectory
$\mathbb{Z}$	Set of integer numbers

Greek Symbols

Symbol	Definition
β	Enstrophy dissipation scale
γ	Relaxation Runge-Kutta coefficient
δ	Shear layer thickness SLR
δ_i^j, δ_{ij}	Notations of Kronecker delta
Δt	Temporal stepsize
$\boldsymbol{\delta}_u$	Vector $[1, 0]^T$
$\boldsymbol{\delta}_v$	Vector $[0, 1]^T$
Δx_i^p	Face size in x-direction of control volume Ω_i^p
Δx_i^u	Face size in x-direction of control volume Ω_i^u
Δy_i^p	Face size in y-direction of control volume Ω_i^p
Δy_i^u	Face size in y-direction of control volume Ω_i^u
ϵ	Initial perturbation amplitude SLR
$\epsilon_b(t)$	Numerical error metric
$\epsilon_t(t)$	Numerical error metric
$\epsilon_u(t)$	Numerical error metric
$\epsilon_x(t)$	Numerical error metric

Symbol	Definition
η	Smallest length scales in 2DT flow
θ	Interpolation variable
λ	LSDEIM Lagrange multiplier
λ_j	j^{th} singular value of SVD of Ξ
λ_o	LSDEIM Lagrange multiplier at optimum
μ	Dynamic viscosity
ν	Kinematic viscosity
Π_a	Palinstrophy source term
$\Pi_{\mathcal{M}_\mathcal{F}}$	LSDEIM projector
$\rho(\boldsymbol{x}, t)$	Continuous density field
Σ	Matrix containing singular values
σ_i	Singular value i
$\boldsymbol{\sigma}(\boldsymbol{x}, t)$	Cauchy stress tensor
Φ	Basis to reduced space
$\widetilde{\Phi}$	Left singular vector of $\widetilde{X}$
$\hat{\Phi}$	Left singular vectors of SVD
$\boldsymbol{\phi}_i$	i^{th} Runge-Kutta stage variable / POD basis vector (clear from context)
Φ_i	POD basis of interval i
$\widetilde{\Phi}_i$	Left singular vector of $\widetilde{X}_i$
$\boldsymbol{\phi}^k$	Runge-Kutta stage block-vector of pressure-like variable of k^{th} Newton-Raphson iteration
$\boldsymbol{\phi}^{n+1}$	Compound Runge-Kutta stage variable at discrete time t^{n+1}
$\phi(\boldsymbol{x}, t)$	Arbitrary scalar field
Ξ	Operator snapshot data set
$\boldsymbol{\xi}_i$	i^{th} Left singular vector of SVD of Ξ
Ψ	Right singular vectors of SVD
$\psi_0(\boldsymbol{x})$	Initial streamfunction
$\hat{\psi}_{m,n}(t)$	Complex Fourier coefficient of streamfunction with indices m and n
$\psi(\boldsymbol{x}, t)$	Streamfunction
$\boldsymbol{\omega}$	Vorticity vector
ω	Vorticity (2D)
Ω	Continuous spatial domain
Ω_h	Diagonal matrix with FVM cell sizes
Ω_i^p	Pressure control volume with index i
Ω_i^u	u-Velocity control volume with index i
Ω_i^v	v-Velocity control volume with index i

1

Introduction

Computational fluid dynamics (CFD) has become an integral part of many modern engineering applications. The increase in computational power in recent decades has allowed engineers to model increasingly larger fluid dynamical systems. However, many modern applications are of a multi-query or real-time nature e.g. design optimization [80] and uncertainty quantification [34] for the former and real-time control [79, 68] and digital twin technology [45] for the latter, and still pose prohibitively large computational costs. Reduced order models (ROMs) have been proposed as a solution to this problem. A ROM is a type of surrogate model that approximates the high-dimensional full scale model, or full order model (FOM) as it is often referred to in the model reduction community, in a low-dimensional way by finding approximate formulations of involved quantities or operators. The low-dimensionality of the model consequently makes the ROM significantly cheaper to evaluate than the large scale model.

Traditionally in the CFD community, these ROMs have been constructed by projecting the fluid dynamics equations of interest onto low-dimensional linear spaces obtained from the proper orthogonal decomposition (POD) algorithm [47, 94], using either Galerkin [86, 79, 98, 50, 58] or Petrov-Galerkin [23, 41, 22] projection. More recently, alternatives have also been explored. ROMs have been constructed without availability of a FOM by inferring the ROM from data using operator inference methods [73, 92, 16, 61]. Machine learning methods like convolutional autoencoders have been used in a projective sense [55, 78] and also inference methods [66, 60] have been applied to obtain nonlinear low-dimensional approximations. Other nonlinear dimensionality reduction methods like diffusion maps [95] have also been leveraged. However, the traditional POD-Galerkin methods remain powerful and this thesis will primarily deal with these methods and their natural extensions like the principal interval decomposition (PID) [18, 49].

Nevertheless, the traditional methods do have limitations. For turbulent (and convection-dominated) flows that are of engineering interest it is well-known that linear, projection-based ROMs suffer from stability and accuracy issues [86, 38, 12, 6, 58]. Efforts have been made to solve this issue and an overview is provided by [38, 86]. A promising solution is structure-preservation, this entails constructing ROMs such that the underlying physics of fluid flows are respected. Especially conservation of kinetic energy is an important physical principle to uphold in a ROM with regards to stability as it bounds the norm of the solution [86, 2, 26, 99].

An investigation of the current state of the literature, reported in a separate literature overview document ('Methods for Energy-Conserving Model Reduction for Fluid Flow'), resulted in the conclusion that attempts at constructing structure-preserving ROMs of fluid flow fall in one of four categories. The first category is constrained optimization projection [24, 90, 42, 17, 48]; here model reduction is cast into an optimization problem and constrained to preserve structure. A second category is formed by symmetry-preservation [86, 2, 26, 99] where symmetries of the continuous analogues of ROM operators have been preserved resulting in the conservation of energy and nonlinear stability. In [40, 27, 76, 46, 1] Hamiltonian physics are preserved in a low-dimensional setting, forming a third category. Finally, in [64, 105, 56, 65, 13, 5] data-driven approaches using inference and machine learning methods have been adopted

establishing a fourth category.

It is clear that the issue of stability and its resolution through structure-preservation has been the subject of considerable research efforts. However, reduction of nonlinear models suffers also from another issue. Namely, evaluating nonlinear operators in a reduced setting using the POD requires intermediate lifting of the reduced representation to the high-dimensional (FOM) spaces. Thus, although the ROM is low-dimensional, the computational effort to evaluate it is still high-dimensional, defeating its purpose. Methods to overcome this problem are referred to as hyper-reduction methods [21]. For sufficiently simple cases exact hyper-reduction methods exist that eliminate computational dependence on the FOM dimensions [86, 4]. This can be done when the underlying nonlinearity is of polynomial nature. Yet, these methods can become prohibitively expensive for larger ROMs as noted in [86]. Hence, approximate hyper-reduction methods may be considered since these generally have better scaling properties. Examples of such approximate hyper-reduction methods are the discrete empirical interpolation method (DEIM) [29, 30, 28, 14], Gauss-Newton with approximated tensors (GNAT) [25, 23] and energy-conserving sampling and weighting[1] (ECSW) [41, 37, 36].

The usage of approximate hyper-reduction methods comes at a cost. Namely, out of the box many hyper-reduction methods do not preserve the structure of the operators to which they are applied and thus stability can be lost. The field of model reduction of Hamiltonian systems has offered some solutions for this; [27] proposes a structure-preserving DEIM variant for systems with Hamiltonian functionals with non-quadratic terms; [102] improves this DEIM variant and [63] proposes a structure-preserving DEIM variant for nonlinear Hamiltonian operators that preserves skew-symmetry. However, the fluid dynamics models of interest to this thesis have quadratic Hamiltonian functionals [70, 11] (kinetic energy) making the first two methods ([27, 102]) inapplicable. Furthermore, the approximate method proposed in [63] scales computationally as the prohibitively expensive exact method used in [86]. For this reason, none of these DEIM variants are applicable to the models of interest to this thesis and there is a gap in the present literature.

It is the previously discussed gap where the present thesis aims to provide a contribution to the literature. More specifically, this thesis aims at developing structure-preserving DEIM variants that are capable to robustly, accurately and efficiently deal with convection-dominated flows. Consequently, the following research goal is formulated:

> "To develop robust, accurate and efficient reduced order models for the incompressible Navier-Stokes equations."

This thesis will be organized as follows. First, the governing equations will be introduced in chapter 2 and matters important to structure-preservation will be highlighted. In chapter 3 a structure-preserving FOM will be introduced and its implementation will be verified. In chapter 4 a structure-preserving ROM will be introduced, alongside hyper-reduction. Following this, three structure-preserving hyper-reduction methods will be proposed which will be applied to the aforementioned ROM. Finally, in chapter 5 the performance of the proposed structure-preserving hyper-reduction methods will be analysed using two convection-dominated test cases.

[1] Contrary to what the name suggests ECSW does not preserve structure for the fluid dynamical simulations of interest to this thesis.

2

The Governing Equations

In this chapter the incompressible Navier-Stokes equations are introduced. These equations govern the motion of incompressible flows and are ubiquitous in fluid mechanics. From the incompressible Navier-Stokes equations additional equations will be derived that describe the evolution of two conserved quantities: total momentum and total kinetic energy. These derivations will require consideration of the mathematical properties of the operators in the incompressible Navier-Stokes equations. Conserving these quantities in reduced order models (ROMs) of the incompressible Navier-Stokes equations will be the topic of this thesis.

2.1. The Incompressible Navier-Stokes Equations

The incompressible Navier-Stokes equations describe the conservation of mass and momentum of a fluid in a domain $\Omega \subset \mathbb{R}^d$ during a time interval $[0, T]$, with $d \in \{2, 3\}$ being the dimensions of the spatial domain and $T \in \mathbb{R}^+$ being a time. The set of equations constituting the incompressible Navier-Stokes equations can be derived from first principles by considering Reynolds transport theorem:

$$\frac{d}{dt}\int_{V(t)} \phi dV = \int_{V(t)} \left(\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \boldsymbol{u} \right) dV, \tag{2.1}$$

where $V(t) : \mathbb{R}^+ \to \mathbb{R}^d$ is a volume of infinitesimal fluid particles moving with the flow referred to as a material volume, $\phi(\boldsymbol{x}, t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ is a scalar field describing a quantity related to the flow, $\boldsymbol{u}(\boldsymbol{x}, t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}^d$ is the fluid velocity vector, $\boldsymbol{x} \in \Omega$ is a spatial coordinate and $t \in \mathbb{R}^+$ is time. A derivation of (2.1) may be found in [103]. Considering the scalar field $\rho(\boldsymbol{x}, t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ describing the density of the flow, Reynolds transport theorem states:

$$\frac{d}{dt}\int_{V(t)} \rho dV = \int_{V(t)} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} \right) dV. \tag{2.2}$$

For any material volume the term on the left-hand side of (2.2) should be zero as a result of mass conservation, hence, using (2.2), it can be stated that:

$$\frac{d}{dt}\int_{V(t)} \rho dV = \int_{V(t)} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} \right) dV = 0. \tag{2.3}$$

Relation (2.3) equals zero for any arbitrary $V(t)$, therefore the integrand must itself be equal zero, resulting in:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0. \tag{2.4}$$

This partial differential equation (PDE) is referred to as the continuity equation. If, additionally, the density of each infinitesimal fluid particle in the flow remains constant, $\rho(\boldsymbol{y}(t), t) = \rho(\boldsymbol{y}(0), 0)$, where $\boldsymbol{y}(t) : \mathbb{R}^+ \to \mathbb{R}^d$ is the trajectory of an infinitesimal fluid particle, the continuity equation (2.4) simplifies further to:

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2.5}$$

If the fluid flow satisfies (2.5) it is said to be incompressible.
Considering now the scalar field of the momentum per unit volume in direction i, $\rho u_i(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$, Reynolds transport theorem states:

$$\frac{d}{dt}\int_{V(t)} \rho u_i dV = \int_{V(t)} \left(\frac{\partial \rho u_i}{\partial t} + \nabla \cdot \rho u_i \boldsymbol{u} \right) dV. \tag{2.6}$$

Using Newton's second law, the term on the left-hand side of (2.6) can be equated to the sum of forces in direction i as follows:

$$\frac{d}{dt}\int_{V(t)} \rho u_i dV = \int_{V(t)} \left(\frac{\partial \rho u_i}{\partial t} + \nabla \cdot \rho u_i \boldsymbol{u} \right) dV = \int_{S(t)} f_i^s dS + \int_{V(t)} \rho f_i^b dV, \tag{2.7}$$

where $S(t) : \mathbb{R}^+ \to \mathbb{R}^d$ is the surface of the material volume $V(t)$, $f_i^s(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ is the surface stress in direction i and $f_i^b(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ the body force per unit mass in direction i. Following [10], it can be stated that the vector of surface stresses $\boldsymbol{f}^s = [f_i^s]^T$ can be written as the contraction between a second order stress tensor, $\boldsymbol{\sigma}(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}^{d\times d}$, and the outward pointing normal vector to the material volume referred to as $\hat{\boldsymbol{n}}(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}^d$:

$$\boldsymbol{f}^s = \boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}.$$

Collecting all momentum equations (2.7) in a vector equation and applying the divergence theorem to the surface integral on the right-hand side of (2.7) results into:

$$\frac{d}{dt}\int_{V(t)} \rho \boldsymbol{u} dV = \int_{V(t)} \left(\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot \rho \left(\boldsymbol{u} \otimes \boldsymbol{u} \right) \right) dV = \int_{V(t)} \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{f}^b dV. \tag{2.8}$$

Relation (2.8) holds for any arbitrary material volume $V(t)$, hence it holds on a differential level. This allows the following to be written:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot \rho \left(\boldsymbol{u} \otimes \boldsymbol{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{f}^b. \tag{2.9}$$

Relation (2.9) is referred to as the Cauchy momentum equation. Typically (2.9) is supplemented with a constitutive relation for the stress tensor $\boldsymbol{\sigma}$. When the fluid is Newtonian the constitutive relation is [31, 103]:

$$\boldsymbol{\sigma} = -pI + 2\mu \left(\frac{1}{2} \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right) - \frac{1}{3} \nabla \cdot \boldsymbol{u} \right), \tag{2.10}$$

where $p(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ is the pressure, I is the d dimensional second order identity tensor and μ is the dynamic viscosity (assumed constant). Substituting (2.10) into (2.9) and considering (2.5), the incompressible Navier-Stokes equations are obtained:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot \rho \left(\boldsymbol{u} \otimes \boldsymbol{u} \right) = -\nabla p + \mu \Delta \boldsymbol{u} + \rho \boldsymbol{f}^b \tag{2.11}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.12}$$

where Δ denotes the Laplace operator. Furthermore, the incompressible Navier-Stokes equations can be normalized by introducing a characteristic velocity $U_c \in \mathbb{R}^+$ and a characteristic length scale $L_c \in \mathbb{R}^+$ determined from the physical setting of the flow. A normalization procedure is described in [103] and results into:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + Re^{-1} \Delta \boldsymbol{u} + \boldsymbol{f}^b \tag{2.13}$$

$$\nabla \cdot \boldsymbol{u} = 0,$$

where $Re = \rho U_c L_c / \mu$ is the so-called Reynolds number. Although the notation of the variables has not changed, implicitly they have been normalized following the procedure in [103]. To ease notation the inverse of the Reynolds number will be denoted $Re^{-1} := \nu$ in this thesis, as it is more compact. For the sake of brevity, this quantity will be referred to as the kinematic viscosity[1], although it should be understood that in fact it refers to Re^{-1}. Note that, whereas the physical kinematic viscosity has units $m^2 \cdot s^{-1}$, this kinematic viscosity is dimensionless, as it is normalized against $U_c L_c$.

[1] Instead of "the inverse of the Reynolds number" or "the inverted Reynolds number".

2.1.1. Mathematical Properties of Convection, Diffusion and Incompressibility

To derive the evolution equations of total momentum and total kinetic energy, a discussion of the mathematical properties of the spatial operators in (2.13) is in order. To facilitate this discussion the functional setting of relation (2.13) is considered. In operator form (2.13) is given by:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \mathcal{C}\left(\boldsymbol{u}, \boldsymbol{u}\right) = -\mathcal{G}p + \nu \mathcal{D}\boldsymbol{u} \tag{2.14}$$

$$\mathcal{M}\boldsymbol{u} = 0, \tag{2.15}$$

here $\mathcal{C} : U \times U \to U$ is the nonlinear convection operator, $\mathcal{G} : P \to U$ is the linear gradient operator, $\mathcal{D} : U \to U$ is the linear diffusion operator and $\mathcal{M} : U \to P$ is the linear divergence operator. Here P and U are suitably defined function spaces for the pressure and velocity respectively (for more details one can refer to [97, 77]).

Although in the previous paragraph an explicit formulation of the Navier-Stokes equations was derived, more formulations exist which are mathematically equivalent given sufficient smoothness and divergence-freeness of $\boldsymbol{u}$. Specifically, the convection operator can take several, mathematically identical, forms:

$$\begin{aligned}
\text{(Divergence form)} \quad & \mathcal{C}_{\text{div}}(\boldsymbol{c}, \boldsymbol{u}) && := \nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u}) \\
\text{(Advective form)} \quad & \mathcal{C}_{\text{adv}}(\boldsymbol{c}, \boldsymbol{u}) && := (\boldsymbol{c} \cdot \nabla)\boldsymbol{u} \\
\text{(Skew-symmetric form)} \quad & \mathcal{C}_{\text{skew}}(\boldsymbol{c}, \boldsymbol{u}) && := \frac{1}{2}\nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u}) + \frac{1}{2}(\boldsymbol{c} \cdot \nabla)\boldsymbol{u}
\end{aligned}$$

among others. It is straightforward to show that all these operators are identical. Applying the product rule to $\mathcal{C}_{\text{div}}(\boldsymbol{c}, \boldsymbol{u})$ results into:

$$\begin{aligned}
\mathcal{C}_{\text{div}}(\boldsymbol{c}, \boldsymbol{u}) &= (\boldsymbol{c} \cdot \nabla)\boldsymbol{u} + (\nabla \cdot \boldsymbol{c})\boldsymbol{u} \\
&= (\boldsymbol{c} \cdot \nabla)\boldsymbol{u} \\
&= \mathcal{C}_{\text{adv}}(\boldsymbol{c}, \boldsymbol{u})
\end{aligned}$$

since $\boldsymbol{c}$ satisfies (2.5). Equivalence for $\mathcal{C}_{\text{skew}}$ then follows trivially. The skew-symmetric form of the convection operator $\mathcal{C}_{\text{skew}}$ is useful for proving an important property of the convection operator. Namely, that the convection operator is skew-adjoint. To proof this property an inner product is required, to this end the $L^2(\Omega)$-inner product is used. The $L^2(\Omega)$-inner product will be denoted $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{L^2} = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v} d\Omega$ and its induced norm will be denoted as $||\boldsymbol{u}||_{L^2}$. With this choice of inner product the skew-adjoint property implies that for the adjoint operator $\mathcal{C}^* : U \times U \to U$ of the convection operator $\mathcal{C}$ satisfying $\langle \boldsymbol{v}, \mathcal{C}(\boldsymbol{c}, \boldsymbol{u}) \rangle_{L^2} = \langle \mathcal{C}^*(\boldsymbol{c}, \boldsymbol{v}), \boldsymbol{u} \rangle_{L^2}$ it holds that $\mathcal{C}^* = -\mathcal{C}$. The proof is given as follows. Firstly, the contraction between $\mathcal{C}_{\text{skew}}(\boldsymbol{c}, \boldsymbol{u})$ and $\boldsymbol{v}$ is calculated:

$$\begin{aligned}
\mathcal{C}_{\text{skew}}(\boldsymbol{c}, \boldsymbol{u}) \cdot \boldsymbol{v} &= \frac{1}{2}\left(\nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{u})\right) \cdot \boldsymbol{v} + \frac{1}{2}\left((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}\right) \cdot \boldsymbol{v} \\
&= \frac{1}{2}\left[((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v} + (\nabla \cdot \boldsymbol{c})(\boldsymbol{u} \cdot \boldsymbol{v})\right] + \frac{1}{2}\left[\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}) - \boldsymbol{u} \cdot (\nabla \cdot (\boldsymbol{c} \otimes \boldsymbol{v}))\right] \\
&= \frac{1}{2}\left[((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v} + (\nabla \cdot \boldsymbol{c})(\boldsymbol{u} \cdot \boldsymbol{v})\right] + \frac{1}{2}\left[\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}) - ((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}) \cdot \boldsymbol{u} - (\nabla \cdot \boldsymbol{c})(\boldsymbol{u} \cdot \boldsymbol{v})\right] \\
&= \frac{1}{2}\left[((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v}\right] + \frac{1}{2}\left[\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}) - ((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}) \cdot \boldsymbol{u}\right].
\end{aligned}$$

Note that in the third line, the second term is cancelled with the last term. It was not necessary to invoke $\nabla \cdot \boldsymbol{c} = 0$. Secondly, the integral of the contraction over the domain Ω is calculated to complete the $L^2(\Omega)$-inner product:

$$\begin{aligned}
\langle \mathcal{C}_{\text{skew}}(\boldsymbol{c}, \boldsymbol{u}), \boldsymbol{v} \rangle_{L^2} &= \int_\Omega \frac{1}{2}\left[((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}) \cdot \boldsymbol{v}\right] + \frac{1}{2}\left[\nabla \cdot ((\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{c}) - ((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}) \cdot \boldsymbol{u}\right] d\Omega \\
&= \frac{1}{2}\left\langle ((\boldsymbol{c} \cdot \nabla)\boldsymbol{u}), \boldsymbol{v} \right\rangle_{L^2} - \frac{1}{2}\left\langle ((\boldsymbol{c} \cdot \nabla)\boldsymbol{v}), \boldsymbol{u} \right\rangle_{L^2} \\
&= -\left\langle \boldsymbol{u}, \mathcal{C}_{\text{skew}}(\boldsymbol{c}, \boldsymbol{v}) \right\rangle_{L^2}.
\end{aligned}$$

The second term under the integral in the first line disappears for appropriate boundary conditions (periodic or homogeneous Dirichlet) as a result of the divergence theorem. As discussed in [86], it should be noted that the skew-symmetric form, $\mathcal{C}_{skew}$, is skew-adjoint regardless of the divergence-freeness of $\boldsymbol{c}$. The other forms of the convection operator are only skew-adjoint when $\nabla \cdot \boldsymbol{c} = 0$, since then they are mathematically equivalent to $\mathcal{C}_{\text{skew}}$[2].

Another important mathematical property of the spatial operators in (2.14) is that the diffusion operator is self-adjoint and negative-definite. Here, self-adjoint refers to the property that $\mathcal{D}^* = \mathcal{D}$. The proof of this is an exercise in integration by parts:

$$\begin{aligned}\langle \mathcal{D}\boldsymbol{u}, \boldsymbol{v}\rangle_{L^2} &= \int_\Omega \left(\nabla \cdot \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right) \cdot \boldsymbol{v} d\Omega \\ &= \int_\Omega \nabla \cdot (\nabla \boldsymbol{u}^T \boldsymbol{v}) - \nabla \boldsymbol{u} : \nabla \boldsymbol{v} d\Omega \\ &= -\nabla \boldsymbol{u} : \nabla \boldsymbol{v} d\Omega \\ &= \langle \boldsymbol{u}, \mathcal{D}\boldsymbol{v}\rangle_{L^2}.\end{aligned}$$

The first term under the integral in the second line disappears after using the divergence theorem for periodic or homogeneous Dirichlet boundary conditions. Taking $\boldsymbol{v} = \boldsymbol{u}$ gives:

$$\langle \mathcal{D}\boldsymbol{u}, \boldsymbol{u}\rangle_{L^2} = -\int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{u} d\Omega = -||\nabla \boldsymbol{u}||^2_{L^2} \leq 0,$$

proving the diffusion operator is negative-definite.

The last property that is of importance is that the divergence and the gradient are each other's negated adjoints, i.e. $\mathcal{G}^* = -\mathcal{M}$. For a scalar function $\phi \in P$ and a vector valued function $\boldsymbol{u} \in U$ the following holds:

$$\langle \boldsymbol{u}, \mathcal{G}\phi\rangle_{L^2} = \int_\Omega \boldsymbol{u} \cdot \mathcal{G}\phi d\Omega = \int_\Omega \nabla \cdot (\phi \boldsymbol{u}) - \phi(\nabla \cdot \boldsymbol{u}) d\Omega = -\langle \mathcal{M}\boldsymbol{u}, \phi\rangle_{L^2}, \tag{2.16}$$

proving the statement. The first term under the integral, $\nabla \cdot (\phi \boldsymbol{u})$, disappears after applying the divergence theorem for periodic or homogeneous Dirichlet boundary conditions in case $\phi = p$ and $\boldsymbol{u}$ is the velocity vector.

2.1.2. Total Momentum Conservation

The first conserved quantity that will be considered in this thesis is the total momentum. Total momentum $\boldsymbol{P}(t) : \mathbb{R}^+ \to \mathbb{R}^d$ will be defined as:

$$\boldsymbol{P}(t) = \int_\Omega \boldsymbol{u} d\Omega. \tag{2.17}$$

The momentum equations (2.13) can be integrated over the domain Ω to find an evolution equation for the total momentum $\boldsymbol{P}$:

$$\begin{aligned}\frac{d\boldsymbol{P}}{dt} = \frac{d}{dt}\int_\Omega \boldsymbol{u} d\Omega = \int_\Omega \frac{\partial \boldsymbol{u}}{\partial t} d\Omega &= \int_\Omega -\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla p + \mu \Delta \boldsymbol{u} + \boldsymbol{f}^b d\Omega \\ &= \int_\Omega \nabla \cdot \left[-(\boldsymbol{u} \otimes \boldsymbol{u}) - pI + \mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\right] d\Omega + \int_\Omega \boldsymbol{f}^b d\Omega,\end{aligned}$$

using the divergence theorem, the first integral on the right-hand side disappears for periodic boundary conditions. Setting $\boldsymbol{f}^b = 0$, it is found for periodic boundary conditions that:

$$\frac{d\boldsymbol{P}}{dt} = 0. \tag{2.18}$$

This implies that for periodic domains the total momentum is a conserved quantity given there are no body forces. This may be interpreted as a statement of Newton's second law. Namely, every stress vector is an internal stress vector on a periodic domain hence cancelling out. Furthermore, on periodic domains there is zero net inflow of momentum due to convection, such that indeed there is no net force on the fluid in the periodic domain $\Omega = \mathbb{T}^d$.

[2] Given the velocity field is sufficiently smooth.

2.1.3. Total Kinetic Energy Conservation

The second conserved quantity that will be considered in this thesis is the total kinetic energy. The total kinetic energy $K(t) : \mathbb{R}^+ \to \mathbb{R}^+$ is defined as:

$$K(t) = \frac{1}{2} ||\boldsymbol{u}||^2_{L^2} . \tag{2.19}$$

An evolution equation of the total kinetic energy can be found by taking the $L^2(\Omega)$-norm of relation (2.14):

$$\begin{aligned} \frac{dK}{dt} = \frac{1}{2}\frac{d}{dt} \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{L^2} = \left\langle \boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial t} \right\rangle_{L^2} &= \langle \boldsymbol{u}, -\mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) - \mathcal{G}p + \mu \mathcal{D}\boldsymbol{u} \rangle_{L^2} \\ &= - \langle \boldsymbol{u}, \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) \rangle_{L^2} - \langle \boldsymbol{u}, \mathcal{G}p \rangle_{L^2} + \mu \langle \boldsymbol{u}, \mathcal{D}\boldsymbol{u} \rangle_{L^2} . \end{aligned}$$

Since the convection operator is skew-adjoint it holds that $\langle \boldsymbol{u}, \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) \rangle_{L^2} = - \langle \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u} \rangle_{L^2} = - \langle \boldsymbol{u}, \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) \rangle_{L^2}$, which can only hold if $\langle \boldsymbol{u}, \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) \rangle_{L^2} = 0$. Additionally, since the gradient operator and the divergence operator are each other's negated adjoints it can be stated that $\langle \boldsymbol{u}, \mathcal{G}p \rangle_{L^2} = - \langle \mathcal{M}\boldsymbol{u}, p \rangle_{L^2}$. Condition (2.5) may now be invoked to state that this inner product also equals zero. Lastly, the negative-definiteness of the diffusion operator can be employed to simplify the evolution equation of the total kinetic energy further to:

$$\frac{dK}{dt} = -\nu ||\nabla \boldsymbol{u}||^2_{L^2} \leq 0. \tag{2.20}$$

Relation (2.20) implies that the kinetic energy, or the norm of the velocity field $\boldsymbol{u}$, is a monotonically decreasing quantity and is conserved in the inviscid case ($\nu = 0$) for periodic or homogeneous Dirichlet boundary conditions.

Although it is not strictly the case that kinetic energy is conserved in the inviscid limit for three-dimensional flows due to the so-called energy cascade [35], sometimes in this thesis the phrase "kinetic energy is conserved in the inviscid limit" will be used. From now on it will be clear that this phrase bares the same intention as "kinetic energy is conserved for an inviscid flow". There is some sense behind this as will be seen in chapter 5. Namely, in two spatial dimensions kinetic energy *is* conserved in the inviscid limit ($\nu \to 0$) and thesis will primarily deal with this simplification[3].

[3]Though, all results are equally valid in the three dimensional space.

3

A Structure-Preserving FOM of the Incompressible Navier-Stokes Equations

In this chapter a so-called structure-preserving full order model (FOM) will be introduced. The notion of structure-preservation will be discussed and its implications on numerical stability will be highlighted. Following this, a discussion on the spatial discretization and temporal discretization of the FOM is provided. The FOM is based on discretization algorithms provided in a set of papers [86, 85, 87]. In this master thesis research an implementation of these algorithms was made in the C++ programming language using a combination of the Armadillo template-based linear algebra library [89, 88] and the Lis library of iterative solvers for linear systems [67]. The implementation is verified using an exact solution of the two dimensional incompressible Navier-Stokes equation, called the Taylor-Green vortex, by analysing the spatial and temporal convergence behaviour of the numerical error. Upon this structure-preserving FOM, a structure-preserving reduced order model will be constructed in later chapters.

3.1. Structure-Preservation and Nonlinear Stability

Structure-preservation is understood as preserving at least part of the mathematical structure underlying a continuous model in a numerical discretization of the corresponding model. The mathematical structure of a continuous model is not a strictly defined object. For the model of interest to this research, the incompressible Navier-Stokes equations, a part of the mathematical structure is the conservation of mass, total momentum and total kinetic energy in the inviscid limit. However, this is not the full structure of the model; indeed there are other conserved quantities like helicity [106]. Furthermore, mathematical symmetries may be considered as part of the structure of a mathematical model e.g. those associated to Noether's theorem [69].

This thesis will focus on the preservation of mass (satisfying relation (2.5) discretely) and satisfying discrete analogues to the conservation of total momentum (2.17) and total kinetic energy (2.20). Especially, the conservation of a discrete analogue to the total kinetic energy will be of interest as it has strong implications on the numerical stability of the discretization schemes used [8, 75, 53, 33, 100]. Namely, the discrete kinetic energy will be defined as a norm of the numerical solution. Hence, if this kinetic energy is monotonically decreasing in time like (2.20) the norm of the numerical solution will be monotonically decreasing. This in turn provides point-wise, finite bounds on the numerical solution, proving stability. In contrast to linear stability, the type of stability discussed in this thesis does not concern linear perturbations around a steady state but refers to the full nonlinear numerical model. For this reason this type of stability is referred to as nonlinear stability.

3.2. Spatial Discretization

In order to discretize (2.13) the finite volume method (FVM) will be used which is a method especially suitable for conservation laws like the incompressible Navier-Stokes equations. The

method of lines will be used to split the discretization over a spatial and temporal discretization. In what follows the used spatial discretizations schemes for the incompressible Navier-Stokes equations in $d = 2$ dimensions will be exhibited. This will result in a system of coupled ordinary differential equations (ODEs) complemented by a set of linear conditions:

$$\Omega_h \frac{d\boldsymbol{u}_h}{dt} + C_h(\boldsymbol{u}_h) = -G_h \boldsymbol{p}_h + \nu D_h \boldsymbol{u}_h \tag{3.1}$$

$$M_h \boldsymbol{u}_h = 0. \tag{3.2}$$

Here $\boldsymbol{u}_h(t) : \mathbb{R}^+ \to \mathbb{R}^N$ are the numerical velocity values arranged in a vector, $\Omega_h \in \mathbb{R}^{N\times N}$ is a diagonal matrix originating from application of the FVM, $C_h(\boldsymbol{u}_h) : \mathbb{R}^N \to \mathbb{R}^N$ is the spatial discretization of the nonlinear convection operator, $G_h \in \mathbb{R}^{N\times N_p}$ is the spatial discretization of the gradient operator, $\boldsymbol{p}_h(t) : \mathbb{R}^+ \to \mathbb{R}^{N_p}$ are the numerical pressure values arranged in a vector, $D_h \in \mathbb{R}^{N\times N}$ is the spatial discretization of the diffusion operator, $M_h \in \mathbb{R}^{N_p\times N}$ is the spatial discretization of the divergence operator, $N = N_u + N_v$ is the total number of velocity unknowns and N_u and N_v are the numbers of velocity unknowns in the x and y directions respectively and N_p is the number of pressure unknowns. Relations (3.1)-(3.2) are in turn complemented by a vector of initial conditions $\boldsymbol{u}_0 = \boldsymbol{u}_h(0)$ and suitable boundary conditions. The choice of boundary conditions for the flows considered in this thesis will be periodic.

3.2.1. Staggered Grid

The FVM partitions the domain Ω into a mesh $\mathcal{M}_h$ of N_{Ω^p} non-overlapping control volumes $\Omega_i^p \subset \Omega$ such that $\Omega = \cup_{i=1}^{N_\Omega} \Omega_i^p$. The numerical solution values u_i and p_i in $\boldsymbol{u}_h = [u_i]^T$ and $\boldsymbol{p}_h = [p_i]^T$ are nodal values associated to nodes placed on this mesh. The exact arrangement of nodes is an important choice to make. A natural choice is the staggered grid where the unknowns are configured such that a control volume Ω_i^p has a pressure node in its centre and has velocity nodes on its faces. The specific velocity component on the volume face is chosen such that it is normal to the respective face it is on. In this thesis periodic domains are considered of which a period can be represented on a rectangular domain. As an example, a mesh of $N_x = 3$ by $N_y = 3$ finite volume cells with a staggered configuration of unknowns and periodic boundaries is shown in Figure 3.1; here N_x and N_y refer to the number of cells in the x and y directions respectively. The black cells in Figure 3.1 correspond to pressure unknowns. For the FVM finite volume cells centred on the velocity unknowns will also be required. In Figure 3.1 an example of a finite volume cell Ω_i^u corresponding to a u-unknown is given using a dashed red line. An example of a finite volume cell Ω_i^v corresponding to a v-unknown is given using a dashed blue line. Because the domain is considered to be periodic the u-unknowns on the left and right boundaries will be equal in value and should be considered as one column of unknowns. Similarly the v-unknowns on the upper and lower boundary are equal and should also be considered as only one row of unknowns. This treatment of periodic boundaries requires Ω_i^u and Ω_i^v finite volume cells on the boundary to be wrapped around their associated boundaries; this is also depicted for a Ω_i^u cell in dashed orange lines and a Ω_i^v cell in dashed green lines. Indeed, the choice will be made to use a staggered configuration of unknowns. This choice is beneficial for the construction of structure-preserving schemes [86] and furthermore it prevents well-known numerical problems like pressure-decoupling [44].

3.2.2. Finite Volume Formulation of the Incompressible Navier-Stokes Equations

Using the FVM, (2.13) is discretized by integrating it over a Ω_i^u cell for u-unknowns and a Ω_i^v cell for v-unknowns like:

$$\int_{\Omega_i^u} \frac{\partial u}{\partial t} d\Omega = \int_{\Omega_i^u} \nabla \cdot [-(u\boldsymbol{u}) - p\boldsymbol{\delta}_u + \nu \nabla u]\, d\Omega, \tag{3.3}$$

and similarly for v-unknowns. Here $\boldsymbol{\delta}_u = [1, 0]^T$ for $d = 2$ and equivalently for v-unknowns a vector $\boldsymbol{\delta}_v = [0, 1]^T$ will multiply the pressure. The divergence theorem can now be applied to

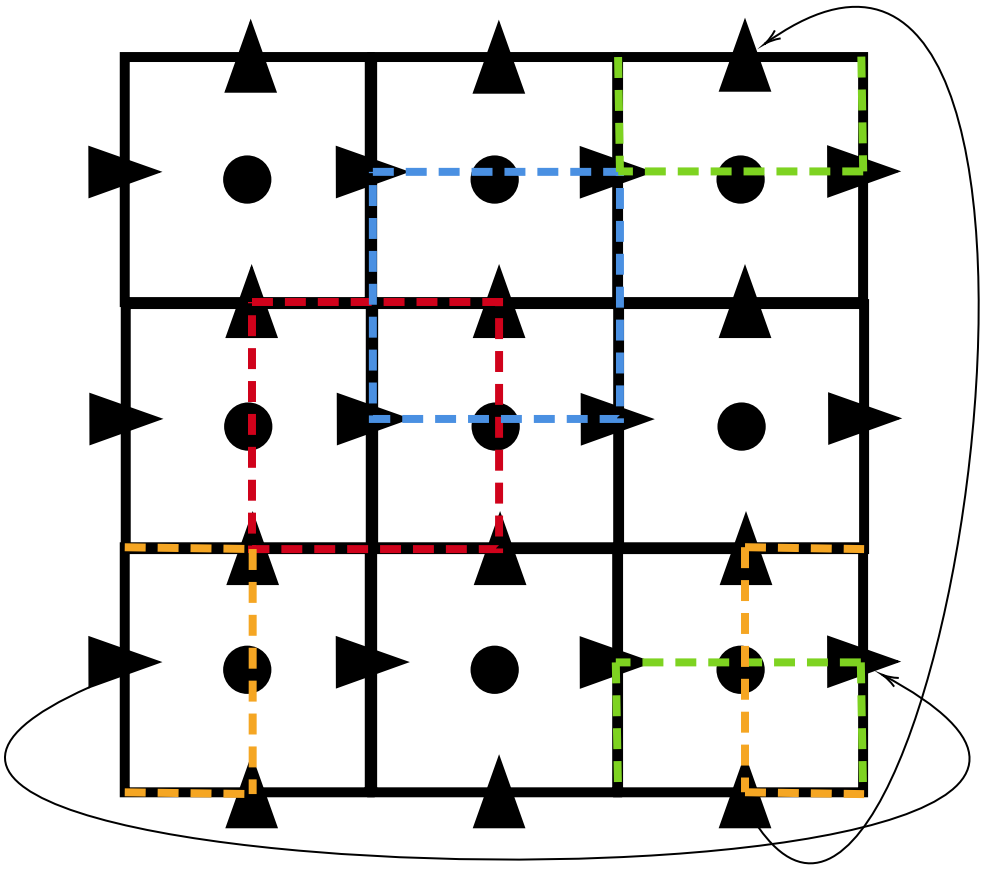


Figure 3.1: Staggered finite volume grid of $N_x = 3$ by $N_y = 3$. The black cells are pressure finite volumes Ω_i^p, the dashed red cell is a finite volume for a u unknown Ω_i^u and the blue cell is a finite volume of v unknown Ω_i^v, the green and orange cells are finite volumes belonging to v and u unknowns on the periodic boundary, these should be considered as connected.

(3.3) to obtain:

$$\int_{\Omega_i^u} \frac{\partial u}{\partial t} d\Omega = \frac{d}{dt} \int_{\Omega_i^u} u d\Omega = \sum_{k \in \{N,E,S,W\}} \int_{\partial_k \Omega_i^u} [-(u\boldsymbol{u}) - p\boldsymbol{\delta}_u + \nu \nabla u] \cdot \hat{\boldsymbol{n}}_k d\Gamma. \tag{3.4}$$

Note that due to the divergence theorem the volume integral has been rewritten to a surface integral over the faces of the finite volume Ω_i^u. These faces are denoted $\partial_k \Omega_i^u$ where the subscript k refers to the orientation of the face. The orientation will be referred to through using a compass-like naming convention. $\hat{\boldsymbol{n}}_k$ is a normal vector to face k. Note also that the temporal derivative has been taken outside of the volume integral on the left-hand side. This volume integral is approximated using the midpoint rule as:

$$\frac{d}{dt} \int_{\Omega_i^u} u d\Omega \approx \frac{d}{dt} \Delta x_i^u \Delta y_i^u u_C = \Delta x_i^u \Delta y_i^u \frac{du_C}{dt}.$$

Here, Δx_i^u and Δy_i^u are the face sizes of volume Ω_i^u. Indeed, the terms $\Delta x_i^u \Delta y_i^u$ are collected on the diagonal of Ω_h. The other terms under the surface integral in (3.4) will be dealt with in subsequent subsections.

In similar fashion (2.5) is integrated over a pressure volume Ω_i^p:

$$\int_{\Omega_i^p} \nabla \cdot \boldsymbol{u} d\Omega = \sum_{k \in \{N,E,S,W\}} \int_{\partial_k \Omega_i^p} \boldsymbol{u} \cdot \hat{\boldsymbol{n}}_k d\Omega = 0. \tag{3.5}$$

Following [86], the surface integrals on the right-hand side are approximated using the midpoint rule:

$$\sum_{k \in \{N,E,S,W\}} \int_{\partial_k \Omega_i^p} \boldsymbol{u} \cdot \hat{\boldsymbol{n}}_k d\Omega \approx v_N \Delta x_i^p + u_E \Delta y_i^p - v_S \Delta x_i^p - u_W \Delta y_i^p = 0, \tag{3.6}$$

here Δx_i^p and Δy_i^p are the cell face sizes of pressure volume Ω_i^p in the x and y-direction respectively. For simplicity the following notation is introduced:

$$\overline{u}_k := u_k \Delta y_i^p, \quad \overline{v}_k := v_k \Delta x_i^p,$$

thus, $\overline{(\cdot)}$ denotes numerical integration using the midpoint rule over the pressure volume face of the associated velocity unknown. Relation (3.6) can be written in matrix vector form when collecting the expression over all Ω_i^p in $\mathcal{M}_h$, taking appropriate care of the periodic boundaries. This results in (3.2).

3.2.3. Convection

Following [86], the term under the surface integral in (3.4) originating from the convection operator is discretized by introducing mesh-free interpolation procedures:

$$\begin{aligned} \sum_{k \in \{N,E,S,W\}} \int_{\partial_k \Omega_i^u} (u\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}_k d\Omega \approx & \frac{1}{2}(u_N + u_C)\frac{1}{2}(\overline{v}_{NW} + \overline{v}_{NE}) + \frac{1}{2}(u_E + u_C)\frac{1}{2}(\overline{u}_E + \overline{u}_C) \\ & - \frac{1}{2}(u_S + u_C)\frac{1}{2}(\overline{v}_{SW} + \overline{v}_{SE}) - \frac{1}{2}(u_W + u_C)\frac{1}{2}(\overline{u}_W + \overline{u}_C). \end{aligned} \tag{3.7}$$

The term u in the surface integral is approximated by averaging the nodal value u_C in the centre of Ω_i^u and the neighbour u_k for $\partial_k \Omega_i^u$. The term $\boldsymbol{u} \cdot \hat{\boldsymbol{n}}_k$ in the surface integral is approximated by averaging the neighbouring velocity values corresponding to orientation k, integrated over their associated pressure volume faces $\partial_k \Omega_i^p$. A similar procedure is used for v-unknowns. Rearranging (3.7) gives:

$$\begin{aligned} \sum_{k \in \{N,E,S,W\}} \int_{\partial_k \Omega_i^u} (u\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}_k d\Omega \approx & \frac{1}{4} u_C \left[(\overline{v}_{NW} + \overline{u}_C - \overline{v}_{SW} - \overline{u}_W) + (\overline{v}_{NE} + \overline{u}_E - \overline{v}_{SE} - \overline{u}_C) \right] \\ & + \frac{1}{4} u_N (\overline{v}_{NW} + \overline{v}_{NE}) + \frac{1}{4} u_E (\overline{u}_E + \overline{u}_C) \\ & - \frac{1}{4} u_S (\overline{v}_{SW} + \overline{v}_{SE}) - \frac{1}{4} u_W (\overline{u}_W + \overline{u}_C). \end{aligned}$$

Using (3.6) it can be stated that the terms in brackets multiplying u_C are both zero. Therefore, the final approximation of the convective term is:

$$\sum_{k\in\{N,E,S,W\}} \int_{\partial_k\Omega_i^u} (u\boldsymbol{u})\cdot\hat{\boldsymbol{n}}_k d\Omega \approx \frac{1}{4}u_N(\overline{v}_{NW}+\overline{v}_{NE})+\frac{1}{4}u_E(\overline{u}_E+\overline{u}_C)-\frac{1}{4}u_S(\overline{v}_{SW}+\overline{v}_{SE})-\frac{1}{4}u_W(\overline{u}_W+\overline{u}_C). \tag{3.8}$$

An equivalent expression for v-unknowns can also be found using this procedure. Collecting the expressions for all Ω_i^u and Ω_i^v in a vector function, $C_h(\boldsymbol{u}_h)$ is obtained. Additionally, the convected velocity components u_k (those without $(\bar{\cdot})$) can be extracted into a vector to write $C_h(\boldsymbol{u}_h)$ as a matrix vector product $\widetilde{C}_h(\boldsymbol{u}_h)\boldsymbol{u}_h$ with $\widetilde{C}_h(\boldsymbol{u}_h) : \mathbb{R}^N \to \mathbb{R}^{N\times N}$. As discussed in [86] the matrix $\widetilde{C}_h(\boldsymbol{u}_h)$ is skew-symmetric $\widetilde{C}_h(\boldsymbol{u}_h) = -\widetilde{C}_h(\boldsymbol{u}_h)^T$, this is a key-property for discrete total kinetic energy-conservation as will be shown in subsection 3.2.7. The boundary conditions can be taken care of by appropriately wrapping the discretization around the periodic domain.

3.2.4. Convection Jacobian

In the solution procedure of (3.1) it is required to solve a nonlinear set of equations when implicit time-stepping schemes are used. These nonlinear systems will be solved iteratively using the Newton-Raphson algorithm. The nonlinearity of this system stems from the discrete convection operator $C_h(\boldsymbol{u}_h)$. To apply the Newton-Raphson algorithm it will therefore be necessary to determine the Jacobian matrix $J_h(\boldsymbol{u}_h) : \mathbb{R}^N \to \mathbb{R}^{N\times N}$ (from here on referred to as the Jacobian) of the convection operator. The row of the Jacobian corresponding to the evolution equation of a u-unknown u_C has the following nonzero entries:

$$\begin{aligned}
\frac{\partial C_h}{\partial u_N} &= \frac{1}{4}(\overline{v}_{NW}+\overline{v}_{NE}), & \frac{\partial C_h}{\partial v_{NE}} &= \frac{1}{4}u_N\Delta x^p_{\mathcal{I}^u_E(i)}\\
\frac{\partial C_h}{\partial u_E} &= \frac{1}{4}(2\overline{u}_E+\overline{u}_C), & \frac{\partial C_h}{\partial v_{SE}} &= -\frac{1}{4}u_S\Delta x^p_{\mathcal{I}^u_E(i)}\\
\frac{\partial C_h}{\partial u_S} &= -\frac{1}{4}(\overline{v}_{SW}+\overline{v}_{SE}), & \frac{\partial C_h}{\partial v_{SW}} &= -\frac{1}{4}u_S\Delta x^p_{\mathcal{I}^u_W(i)}\\
\frac{\partial C_h}{\partial u_W} &= -\frac{1}{4}(2\overline{u}_W+\overline{u}_C), & \frac{\partial C_h}{\partial v_{NW}} &= \frac{1}{4}u_N\Delta x^p_{\mathcal{I}^u_W(i)}\\
\frac{\partial C_h}{\partial u_C} &= \frac{1}{4}(u_E-u_W)\Delta y^u_i. &&
\end{aligned}$$

Here $\mathcal{I}^u_k(i) : \mathbb{Z} \to \mathbb{Z}$ is a mapping from the index i of volume Ω^u_i to the index of the neighbouring pressure volume $\Omega^p_{\mathcal{I}^u_k(i)}$ in direction k. Following an equivalent procedure the nonzero entries of a row of $J_h(\boldsymbol{u}_h)$ associated to the evolution equation of a v-unknown can be calculated. A similar mapping $\mathcal{I}^v_k(i) : \mathbb{Z} \to \mathbb{Z}$ will need to be introduced to take care of the indices belonging to pressure volumes neighbouring the volume Ω^v_i.

3.2.5. Diffusion

The contribution of the diffusion operator in (3.4) is approximated using the midpoint rule for the surface integral and central differences for the directional derivative $\nabla u \cdot \hat{\boldsymbol{n}}_k$ as follows:

$$\sum_{k\in\{N,E,S,W\}} \int_{\partial_k\Omega_i^u} \nu\nabla u\cdot\hat{\boldsymbol{n}}_k d\Omega \approx \sum_{k\in\{N,E,S,W\}} \nu\frac{((\delta^N_k+\delta^S_k)\Delta x^u_i+(\delta^E_k+\delta^W_k)\Delta y^u_i)}{((\delta^N_k+\delta^S_k)|y_k-y_C|+(\delta^E_k+\delta^W_k)|x_k-x_C|)}(u_k-u_C).$$

Here δ^j_i is the Kronecker delta function and x_k and y_k are the spatial coordinates of the node associated to u_k. A similar expression can be derived for v-unknowns. Collecting the equations for all Ω^u_i and Ω^v_i in a matrix produces the discrete diffusion operator D_h. As stated in [86] this matrix is symmetric, $D_h = D_h^T$, and negative definite, $\boldsymbol{x}^T D_h\boldsymbol{x} \leq 0\ \forall \boldsymbol{x} \neq 0$. As a consequence of this there exists a matrix Q_h with N columns such that $D_h = -Q_h^T Q_h$.

3.2.6. Pressure

As noted in [86] the discrete gradient operator G_h can be taken equal to negated transpose of the discrete divergence operator M_h:

$$G_h = -M_h^T. \tag{3.9}$$

This is a discrete analogue to (2.16) as the adjoint of a linear operator in the form of a matrix is the respective matrix's transpose. Relation (3.9) is a natural result of using central discretizations on a staggered grid. Like the continuous case, this property will prove important in cancelling the contribution of the pressure gradient in (3.1) to the discrete analogue to total kinetic energy.

3.2.7. Conservation Properties

It will now be shown that the discretization (3.1)-(3.2) conserves mass, total momentum and kinetic energy. Discrete mass conservation is satisfied by virtue of (3.2). Now a discrete analogue to (2.17) is required. To this end the discrete total momenta:

$$P_h^u(t) := \boldsymbol{e}_u^T \Omega_h \boldsymbol{u}_h \tag{3.10}$$

$$P_h^v(t) := \boldsymbol{e}_v^T \Omega_h \boldsymbol{u}_h, \tag{3.11}$$

are defined. Where $P_h^u(t) : \mathbb{R}^+ \to \mathbb{R}$ and $P_h^v(t) : \mathbb{R}^+ \to \mathbb{R}$ are the discrete total momenta in the x and y-direction respectively and $\boldsymbol{e}_u, \boldsymbol{e}_v \in \mathbb{R}^N$ are vectors containing ones at indices corresponding to evolution equations of u and v-unknowns respectively and zeros otherwise. It is clear that the discrete total momenta approximate the integral over the domain in (2.17) using the composite midpoint rule. An evolution equation for these discrete momenta can be found by differentiating (4.39) and (4.40) with respect to time:

$$\frac{dP_h^u}{dt} = \boldsymbol{e}_u^T \Omega_h \frac{d\boldsymbol{u}_h}{dt} = -\boldsymbol{e}_u^T C_h(\boldsymbol{u}_h) - \boldsymbol{e}_u^T G_h \boldsymbol{p}_h + \nu \boldsymbol{e}_u^T D_h \boldsymbol{u}_h = 0 \tag{3.12}$$

$$\frac{dP_h^v}{dt} = \boldsymbol{e}_v^T \Omega_h \frac{d\boldsymbol{u}_h}{dt} = -\boldsymbol{e}_v^T C_h(\boldsymbol{u}_h) - \boldsymbol{e}_v^T G_h \boldsymbol{p}_h + \nu \boldsymbol{e}_v^T D_h \boldsymbol{u}_h = 0. \tag{3.13}$$

Where the equality with zero holds due to the well-known telescoping property of the flux terms used in the FVM. Thus, the discrete total momentum is a conserved quantity on periodic domains.

A definition of discrete total kinetic energy has to be made on the basis of an inner product, hence the Ω_h-inner product is introduced. The Ω_h-inner product and its induced norm are defined by:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\Omega_h} := \langle \boldsymbol{u}, \Omega_h \boldsymbol{v} \rangle, \quad ||\boldsymbol{u}||_{\Omega_h}^2 := \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\Omega_h},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product. After deriving an evolution equation for the discrete total kinetic energy it will be clear why this is a natural choice. Using the Ω_h-inner product the discrete total kinetic energy $K_h(t) : \mathbb{R}^+ \to \mathbb{R}^+$ is defined as:

$$K_h(t) := \frac{1}{2} ||\boldsymbol{u}_h||_{\Omega_h}^2.$$

An evolution equation is found by temporal differentiation of $K_h(t)$:

$$\begin{aligned} \frac{dK_h}{dt} = \frac{1}{2}\frac{d}{dt} \langle \boldsymbol{u}_h, \boldsymbol{u}_h \rangle_{\Omega_h} &= \left\langle \boldsymbol{u}_h, \Omega_h \frac{d\boldsymbol{u}_h}{dt} \right\rangle \\ &= \langle \boldsymbol{u}_h, -C_h(\boldsymbol{u}_h) - G_h \boldsymbol{p}_h + \nu D_h \boldsymbol{u}_h \rangle \\ &= -\left\langle \boldsymbol{u}_h, \widetilde{C}_h(\boldsymbol{u}_h) \boldsymbol{u}_h \right\rangle - \langle \boldsymbol{u}_h, G_h \boldsymbol{p}_h \rangle + \nu \langle \boldsymbol{u}_h, D_h \boldsymbol{u}_h \rangle \\ &= -\nu ||Q_h \boldsymbol{u}_h||^2 \leq 0, \end{aligned} \tag{3.14}$$

where $||\cdot||$ denotes the Euclidean norm. The last equality is a result of the following properties of the discrete operators; skew-symmetry of $\widetilde{C}_h(\boldsymbol{u}_h)$, the duality between the discrete divergence

and gradient in (3.9) and the symmetry and negative-definiteness of D_h. In detail, the inner product with the discrete convection operator satisfies:

$$\left\langle \boldsymbol{u}_h, \widetilde{C}_h(\boldsymbol{u}_h)\boldsymbol{u}_h \right\rangle = \boldsymbol{u}_h^T \widetilde{C}_h(\boldsymbol{u}_h)\boldsymbol{u}_h = -\boldsymbol{u}_h^T \widetilde{C}_h(\boldsymbol{u}_h)^T \boldsymbol{u}_h = -(\widetilde{C}_h(\boldsymbol{u}_h)\boldsymbol{u}_h)^T \boldsymbol{u}_h = -\boldsymbol{u}_h^T \widetilde{C}_h(\boldsymbol{u}_h)\boldsymbol{u}_h = 0,$$

where skew-symmetry of $\widetilde{C}_h(\boldsymbol{u}_h)$ was invoked in the right-hand sice of the second equality. The inner product with the discrete pressure gradient satisfies:

$$\langle \boldsymbol{u}_h, G_h \boldsymbol{p}_h \rangle = \boldsymbol{u}_h^T G_h \boldsymbol{p}_h = -\boldsymbol{u}_h^T M_h^T \boldsymbol{p}_h = -(M_h \boldsymbol{u}_h)^T \boldsymbol{p}_h = 0,$$

where (3.9) was used in the right-hand side of the second equality and (3.2) was used in the last equality. The last equality in (3.14) is obtained through considering the properties of the discrete diffusion operator:

$$\nu \langle \boldsymbol{u}_h, D_h \boldsymbol{u}_h \rangle = \nu \boldsymbol{u}_h^T D_h \boldsymbol{u}_h = -\nu \boldsymbol{u}_h^T Q_h^T Q_h \boldsymbol{u}_h = -\nu \left|\left| Q_h \boldsymbol{u}_h \right|\right|^2.$$

Indeed this term is always less than or equal to zero (where equality only happens when $Q_h \boldsymbol{u}_h = 0$). In this derivation the decomposition $D_h = -Q_h^T Q_h$ as described in subsection 3.2.5 was used after the second equality. It is clear that (3.14) is a discrete analogue to (2.20) and that the discrete total kinetic energy is conserved in the inviscid limit. Having derived an evolution equation for discrete total kinetic energy, the choice to use the Ω_h-inner product in the definition of K_h becomes evident. Namely, this prevented terms like Ω_h^{-1} to show up in the right-hand side of (3.14). Furthermore, using the Ω_h-inner product has the natural interpretation of a numerical approximation of integration over the domain Ω, mimicking the continuous definition of total kinetic energy in (2.19). Most importantly, the numerical solution of the semi-discrete systems (3.1) has been bounded in the Ω_h-norm by:

$$\left|\left| \boldsymbol{u}(t) \right|\right|_{\Omega_h} \leq \left|\left| \boldsymbol{u}_0 \right|\right|_{\Omega_h}.$$

3.3. Temporal Discretization

Having obtained a structure-preserving semi-discretization (3.1), it is now required to introduce a numerical time-integration scheme to solve the system. In this thesis the broad family of Runge-Kutta methods will be considered to this end. Two matters are now of importance. Firstly, is the matter of assuring condition (3.2) is satisfied for all newly predicted numerical solutions. Secondly, is the matter of assuring that the time-integration scheme preserves the invariance in the inviscid limit of K_h enabled by the previously discussed spatial discretization. Before these matters are discussed, a brief introduction to general Runge-Kutta methods is provided.

Runge-Kutta methods are time-integrations schemes for ODEs or systems of ODEs of the form:

$$\frac{d\boldsymbol{u}}{dt} = f(\boldsymbol{u}, t), \quad \boldsymbol{u}_0 = \boldsymbol{u}(t = 0), \tag{3.15}$$

where $f(\boldsymbol{u}, t) : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$ and n is the dimension of the system of ODEs (being $n = 1$ for a single ODE). These time-integration schemes predict a numerical solution at a new time by using the solution's temporal rate of change or 'slope' at different time points or 'stages' between the new and old time. In its generality a Runge-Kutta method for (3.15) takes the form:

$$\boldsymbol{U}_i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} f(\boldsymbol{U}_j, t_n + c_j \Delta t) \tag{3.16}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{i=1}^{s} b_i f(\boldsymbol{U}_i, t_n + c_i \Delta t). \tag{3.17}$$

Here the superscripts n and $n+1$ specify the time at which the numerical solution is given. The time instance $t = t^n$ belongs to superscript n and $n+1$ belongs to $t = t^{n+1} = t^n + \Delta t$, where the time-step Δt is defined as $\Delta t := t^{n+1} - t^n$. The vectors $\boldsymbol{U}_k$ are intermediate results and will be

c_1	a_{11}	a_{12}	$\dots$	a_{1s}
c_2	a_{21}	a_{22}	$\dots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	$\dots$	$\dots$	a_{ss}
	b_1	$\dots$	$\dots$	b_s

Table 3.1: A general Butcher tableau.

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	a_{s1}	a_{s2}	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s

Table 3.2: A general Butcher tableau of an explicit Runge-Kutta method.

referred to as the stage vector at stage k. The constants a_{ij}, b_i and c_i fully determine a specific member of the Runge-Kutta family. Often these constant are summarized in so-called Butcher tableau's as in Table 3.1. A noteworthy convention for these methods is that $c_j = \sum_{j=1}^{s} a_{ij}$. Finally, s denotes the number of intermediate stages used to approximate $\boldsymbol{u}^{n+1}$.

3.3.1. Time-Integration and Enforcing Incompressibility

An added complexity for the incompressible Navier-Stokes equations over the time-integration procedure described for (3.15) is that every new numerical solution $\boldsymbol{u}_h^{n+1}$ has to satisfy (3.2). Methods to deal with this complexity that are of interest to this thesis have been proposed in [87] for explicit time-integration and in [85] for implicit time-integration. The overarching theme of these approaches is a pressure projection step. This step derives from the following relation. Considering (3.1) and (3.2), taking the discrete divergence of (3.1) results into:

$$M_h \frac{d\boldsymbol{u}_h}{dt} = -M_h \Omega_h^{-1} C_h(\boldsymbol{u}_h) - M_h \Omega_h^{-1} G_h \boldsymbol{p}_h + \nu M_h \Omega_h^{-1} D_h \boldsymbol{u}_h = 0,$$

since, $M_h \frac{d\boldsymbol{u}_h}{dt} = 0$. This equation can be interpreted as a discrete pressure Poisson equation [87], where $L_h := M_h \Omega_h^{-1} G_h$ is similar to a discrete Poisson operator. Rearranging then provides an equation for the numerical solution for the pressure $\boldsymbol{p}_h$:

$$L_h \boldsymbol{p}_h = -M_h \Omega_h^{-1} C_h(\boldsymbol{u}_h) + \nu M_h \Omega_h^{-1} D_h \boldsymbol{u}_h. \tag{3.18}$$

The solution $\boldsymbol{p}_h$ to relation (3.18) can be used to project the newly predicted numerical solution of the velocity $\boldsymbol{u}_h^{n+1}$ on a divergence-free space.

Explicit Runge-Kutta methods are characterized by a Butcher tableau taking a form as in Table 3.2. Using such an explicit time-integration method for (3.1) in combination with relation (3.18) to eliminate the pressure results in a formula for the stages $\boldsymbol{U}_i$:

$$\boldsymbol{U}_i = \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1} \right) \left(-C_h(\boldsymbol{U}_j) + \nu D_h \boldsymbol{U}_j \right), \quad i \in \{2, ..., s\}. \tag{3.19}$$

The importance of constructing divergence-free stage vectors in addition to a divergence-free $\boldsymbol{u}_h^{n+1}$ is highlighted in [87], namely that it is required to obtain correct convergence orders. As it stands (3.19) requires the solution of a Poisson problem for every evaluation of the slope, indeed this is undesirable from a computational perspective. [87] solves this using a rearrangement of (3.19) followed by the introduction of a new pressure-like variable $\boldsymbol{\phi}$ as follows. Firstly, (3.19)

is rearranged as:

$$\boldsymbol{U}_i = \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(-\Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu \Omega_h^{-1} D_h \boldsymbol{U}_j\right) - \Omega_h^{-1} G_h L_h^{-1} \left(\Delta t \sum_{j=1}^{i-1} a_{ij} \left[-M_h \Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu M_h \Omega_h^{-1} D_h \boldsymbol{U}_j\right] \right).$$

Secondly, the stage variable $\boldsymbol{\phi}_i$ is introduced:

$$L_h \boldsymbol{\phi}_i = \Delta t \sum_{j=1}^{i-1} a_{ij} \left[-M_h \Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu M_h \Omega_h^{-1} D_h \boldsymbol{U}_j\right], \quad i \in \{2, ..., s\}, \tag{3.20}$$

such that (3.19) can be written as:

$$\boldsymbol{U}_i = \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(-\Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu \Omega_h^{-1} D_h \boldsymbol{U}_j\right) - \Omega_h^{-1} G_h \boldsymbol{\phi}_i. \tag{3.21}$$

Following [87], this motivates the following solution procedure, based on a prediction-correction algorithm:

$$\begin{aligned} \boldsymbol{V}_i &= \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(-\Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu \Omega_h^{-1} D_h \boldsymbol{U}_j\right) \\ L_h \boldsymbol{\phi}_i &= M_h \boldsymbol{V}_i \\ \boldsymbol{U}_i &= \boldsymbol{V}_i - \Omega_h^{-1} G_h \boldsymbol{\phi}_i, \end{aligned}$$

where it is required that $M_h \boldsymbol{u}^n = 0$. Similarly, $\boldsymbol{u}_h^{n+1}$ can now be determined as:

$$\begin{aligned} \boldsymbol{v}^{n+1} &= \boldsymbol{u}_h^n + \Delta t \sum_{i=1}^{s} b_i \left(-\Omega_h^{-1} C_h(\boldsymbol{U}_i) + \nu \Omega_h^{-1} D_h \boldsymbol{U}_i\right) \\ L_h \boldsymbol{\phi}^{n+1} &= M_h \boldsymbol{v}^{n+1} \\ \boldsymbol{u}_h^{n+1} &= \boldsymbol{v}^{n+1} - \Omega_h^{-1} G_h \boldsymbol{\phi}^{n+1}, \end{aligned} \tag{3.22}$$

to obtain a divergence-free prediction to the numerical solution at $t = t^{n+1}$. A version of this algorithm that deals with the presence of body forces and non-periodic boundary conditions is discussed in [87]. Moreover, the Poisson problems for $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}^{n+1}$ can be solved very efficiently in the Fourier domain by virtue of the periodic boundary conditions [39].

The implementation of implicit Runge-Kutta methods follows [85]. The Butcher tableau of implicit methods takes the more general form as in Table 3.1. Similarly to explicit Runge-Kutta methods, a pressure-like variable $\boldsymbol{\phi}$ is introduced at every stage to obtain the following:

$$\boldsymbol{U}_i = \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{s} a_{ij} \left(-\Omega_h^{-1} C_h(\boldsymbol{U}_j) + \nu \Omega_h^{-1} D_h \boldsymbol{U}_j\right) - \Omega_h^{-1} G_h \boldsymbol{\phi}_i, \quad i \in \{1, ..., s\}. \tag{3.23}$$

The derivation of (3.23) is analogous to (3.21), with the only difference being the use of the general Butcher tableau in Table 3.1 instead of Table 3.2. The primary difficulty in using implicit methods is that all stage vectors $\boldsymbol{U}_j$ are required for the evaluation of (3.23) for any i. Hence, all stage vectors must be solved for simultaneously. Because (3.23) is a nonlinear set of equation it will be solved using the iterative Newton-Raphson procedure. Denoting the iteration index using a superscript k, the Newton-Raphson procedure takes the form:

$$\boldsymbol{U}_i^{k+1} = \boldsymbol{u}_h^n + \Delta t \sum_{j=1}^{s} a_{ij} \left(-\Omega_h^{-1} \left[C_h(\boldsymbol{U}_j^k) + J_h(\boldsymbol{U}_j^k) \left(\boldsymbol{U}_j^{k+1} - \boldsymbol{U}_j^k\right)\right] + \nu \Omega_h^{-1} D_h \boldsymbol{U}_j^{k+1}\right) - \Omega_h^{-1} G_h \boldsymbol{\phi}_i^{k+1}.$$

To complete the set of equation, $s \cdot N_p$ more equations are necessary. Instead of using (3.20) to complete the set of equations, [85] uses the condition (3.2) applied to all $\boldsymbol{U}_j^{k+1}$ individually. A compact expression can now be set up for the iterative procedure:

$$\begin{bmatrix} I - \Delta t A \circ F_1(\boldsymbol{U}^k) & -I_s \otimes (\Omega_h^{-1} G_h) \\ I_s \otimes M_h & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^{k+1} \\ \boldsymbol{\phi}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_s \otimes \boldsymbol{u}_h^n + \Delta t A F_2(\boldsymbol{U}^k) \\ 0 \end{bmatrix}, \tag{3.24}$$

where $\boldsymbol{U}^k \in \mathbb{R}^{s \cdot N}$ and $\boldsymbol{\phi}^k \in \mathbb{R}^{s \cdot N_p}$ are block-vectors containing the corresponding stage vectors in chronological order, I is the $(s \cdot N) \times (s \cdot N)$ identity matrix, I_s is the $s \times s$ identity matrix, $\boldsymbol{e}_s \in \mathbb{R}^s$ is a vector of all ones, $A \in \mathbb{R}^{s \times s}$ is a matrix such that $(A)_{ij} = a_{ij}$, $(\cdot) \otimes (\cdot)$ here denotes the Kronecker product, $F_1(\boldsymbol{U}) : \mathbb{R}^{s \cdot N} \to \mathbb{R}^{(s \cdot N) \times (s \cdot N)}$ is a function producing an $s \times s$ block matrix with blocks satisfying:

$$(F_1(\boldsymbol{U}))_{ij} = \left(-\Omega_h^{-1} J_h(\boldsymbol{U}_j) + \nu \Omega_h^{-1} D_h\right) \quad \forall i \in \{1, ..., s\},$$

$F_2(\boldsymbol{U}) : \mathbb{R}^{s \cdot N} \to \mathbb{R}^{s \cdot N}$ is function producing an s-dimensional block vector with blocks satisfying:

$$(F_2(\boldsymbol{U}))_i = -\Omega_h^{-1} \left(C_h(\boldsymbol{U}_i) - J_h(\boldsymbol{U}_i) \boldsymbol{U}_i\right),$$

$(\cdot) \circ (\cdot)$ denotes the block-element-wise product and $AF_2(\boldsymbol{U}^k)$ is a partitioned matrix vector product such that:

$$(AF_2(\boldsymbol{U}))_i = \sum_{j=1}^{s} (A)_{ij} (F_2(\boldsymbol{U}))_j.$$

In the current implementation of the algorithm it was found that solving this system using a preconditioned GMRES [81] sparse linear solver provided by the Lis [67] linear algebra library is a robust out-of-the-box solution algorithm. The specific preconditioner is a standard incomplete LU preconditioner. Although in this research the choice was made to use out-of-the-box methods, specified algorithm have been developed to attain faster convergence and higher efficiencies for constrained saddlepoint problems like (3.24), e.g. [96]. Once all stage vectors $\boldsymbol{U}_j$ are solved for to a predetermined convergence condition, $\boldsymbol{u}_h^{n+1}$ is found using the procedure (3.22).

3.3.2. Energy-Conserving Runge-Kutta Methods

To have a fully structure-preserving numerical scheme for solving the incompressible Navier-Stokes equations, it is necessary that the discrete total kinetic energy, K_h, is also conserved in the inviscid limit for a fully-discrete scheme. This means that the time-integration method has to conserve the norm of the solution at a discrete level when the semi-discretization preserves this norm. The analysis of a change in the norm of a numerical solution from one time step to the next using Runge-Kutta methods is a classical analysis and provided here too for (3.15):

$$\begin{aligned} \left|\left|\boldsymbol{u}^{n+1}\right|\right|_\Theta^2 - ||\boldsymbol{u}^n||_\Theta^2 &= \left|\left|\boldsymbol{u}^n + \Delta t \sum_{i=1}^{s} b_i f_i\right|\right|_\Theta^2 - ||\boldsymbol{u}^n||_\Theta^2 \\ &= ||\boldsymbol{u}^n||_\Theta^2 + 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{u}^n, f_j \right\rangle_\Theta + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \left\langle f_i, f_j \right\rangle_\Theta - ||\boldsymbol{u}^n||_\Theta^2 \\ &= 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{U}_i, f_i \right\rangle_\Theta + 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{u}^n - \boldsymbol{U}_i, f_i \right\rangle_\Theta + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \left\langle f_i, f_j \right\rangle_\Theta \\ &= 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{U}_i, f_i \right\rangle_\Theta - 2\Delta t \sum_{i,j=1}^{s} b_i a_{ij} \left\langle f_j, f_i \right\rangle_\Theta + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \left\langle f_i, f_j \right\rangle_\Theta \end{aligned} \tag{3.25}$$

where $f_i := f(\boldsymbol{U}_i, t + c_i \Delta t)$, $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_\Theta = \langle \boldsymbol{u}, \Theta \boldsymbol{v} \rangle$ and $||\boldsymbol{u}||_\Theta^2 = \langle \boldsymbol{u}, \boldsymbol{u} \rangle_\Theta$ for any symmetric positive definite (SPD) matrix Θ. In the fourth line the definition a stage vector (3.16) was used in the second term on the right-hand side. Denoting $B = \text{diag}(b_i)$, $\boldsymbol{b} = [b_i]^T$, $i \in \{1, ..., s\}$ the last

two terms in the last line of (3.25) can be taken together in a bilinear form:

$$-2\Delta t \sum_{i,j=1}^{s} b_i a_{ij} \langle f_j, f_i \rangle_\Theta + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \langle f_i, f_j \rangle_\Theta = -\Delta t^2 \sum_{i,j=1}^{s} m_{ij} \langle f_i, f_j \rangle_\Theta ,$$

where $M \in \mathbb{R}^{s \times s}$, satisfying $(M)_{ij} = m_{ij}$, takes the form:

$$M = BA + A^T B - \boldsymbol{b}\boldsymbol{b}^T .$$

Considering the Ω_h-norm, which is indeed captured in the definition of the Θ-norm since Ω_h is a diagonal matrix with positive entries, (3.23) and (3.20) the Ω_h-norm of the solution $\boldsymbol{u}_h$ changes as:

$$\begin{aligned} \left|\left|\boldsymbol{u}_h^{n+1}\right|\right|_{\Omega_h}^2 - ||\boldsymbol{u}_h^n||_{\Omega_h}^2 &= 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{U}_i, \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1}\right) \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) \right\rangle_{\Omega_h} \\ &- \Delta t^2 \sum_{i,j=1}^{s} m_{ij} \left\langle \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1}\right) \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right), \ldots \right. \\ &\qquad \left. \ldots \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1}\right) \left(-C_h(\boldsymbol{U}_j) + \nu D_h \boldsymbol{U}_j\right) \right\rangle_{\Omega_h} . \end{aligned} \tag{3.26}$$

The following holds for the first term on the right-hand side of (3.26):

$$\begin{aligned} 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{U}_i, \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1}\right) \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) \right\rangle_{\Omega_h} = \\ 2\Delta t \sum_{i=1}^{s} b_i \boldsymbol{U}_i^T \Omega_h \Omega_h^{-1} \left(I - G_h L_h^{-1} M_h \Omega_h^{-1}\right) \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) = \\ 2\Delta t \sum_{i=1}^{s} b_i \left[\boldsymbol{U}_i^T \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) - \boldsymbol{U}_i^T G_h L_h^{-1} M_h \Omega_h^{-1} \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) \right] = \\ 2\Delta t \sum_{i=1}^{s} b_i \left[-\nu \left|\left|Q_h \boldsymbol{U}_i\right|\right|^2 + (M_h \boldsymbol{U}_i)^T L_h^{-1} M_h \Omega_h^{-1} \left(-C_h(\boldsymbol{U}_i) + \nu D_h \boldsymbol{U}_i\right) \right] = \\ -2\Delta t \sum_{i=1}^{s} b_i \nu \left|\left|Q_h \boldsymbol{U}_i\right|\right|^2 . \end{aligned}$$

If additionally it holds that:

$$m_{ij} = 0 \ \forall i, j \in \{1, ..., s\}, \tag{3.27}$$

it can be stated for the discrete total kinetic energy:

$$K_h^{n+1} - K_h^n = -\Delta t \sum_{i=1}^{s} b_i \nu \left|\left|Q_h \boldsymbol{U}_i\right|\right|^2 .$$

Thus, the Runge-Kutta method conserves discrete total kinetic energy in the inviscid limit. Furthermore, if $b_i \geq 0 \ \forall i \in \{1, ..., s\}$ the method is nonlinearly stable in the viscous case and the inviscid limit. Runge-Kutta methods satisfying (3.27) exist. An example is the family of Gauss-Legendre methods. In fact, Gauss-Legendre methods will be the energy-conserving methods of interest to this thesis. The Butcher tableaus for two well-known Gauss-Legendre methods, namely the implicit midpoint method and the Gauss-Legendre 4 method, are provided in Table 3.3 and Table 3.4, respectively.

Strictly, the only consistent energy-conserving Runge-Kutta methods are implicit. However, due to the substantial computational cost of solving (3.24) an interest has recently been generated in energy-conserving explicit Runge-Kutta methods. This then will come at the cost of not having consistency of the time-integration schemes. Such an energy-conserving explicit Runge-Kutta method has been proposed in [51]. Here a factor γ multiplying the time step Δt in (3.17) has been introduced in order to effectively conserve energy in the inviscid limit. These methods are referred to as relaxation Runge-Kutta methods.

1/2	1/2
	1

Table 3.3: The Butcher tableau of the implicit midpoint method.

1/2 - 1/6 $\sqrt{3}$	1/4	1/4 - 1/6 $\sqrt{3}$
1/2 + 1/6 $\sqrt{3}$	1/4 + 1/6 $\sqrt{3}$	1/4
	1/2	1/2

Table 3.4: The Butcher tableau of the Gauss-Legendre 4 method.

3.4. Verification

The structure-preserving discretization methods proposed in the previous sections have been implemented in the C++ programming language using a combination of the Armadillo template-based linear algebra library [89, 88] and the Lis library of iterative solvers for linear systems [67]. To verify that the implementation contains no mistakes several convergence studies will be performed. A spatial convergence study will be done to verify the spatial discretization and a temporal convergence study will be done to verify the different time integration schemes that have been implemented. It is useful to base these studies on an analytical solution of (2.13) as this provides precise notions of the error of the numerical solution. The exact solution of choice is the Taylor-Green vortex.

3.4.1. An Exact Solution: The Taylor-Green Vortex

The Taylor-Green vortex is a well-known exact solution of the two-dimensional incompressible Navier-Stokes equations with periodic boundary conditions. In the CFD community it is often used for the convergence analysis of numerical methods to solve eq. (2.13). Several formulations of the flow are possible. In this thesis the flow will be defined on a periodic domain $\Omega = [0, 2\pi] \times [0, 2\pi]$, which gives rise to the following formulation of the solution:

$$u(x, y, t) = \cos(x) \sin(y) e^{-2\nu t}$$
$$v(x, y, t) = -\sin(x) \cos(y) e^{-2\nu t}.$$

It can easily be verified that this solution satisfies the incompressibility condition (2.5) and the periodic boundary conditions on Ω. The solution is plotted for $\nu = 0.001$ at $t = 1$ in Figure 3.2.

3.4.2. Spatial Convergence Study

The spatial convergence study will be performed by calculating the numerical error between the exact and numerical solutions for increasingly finer numerical grids. Following theoretical results [84], the discussed spatial discretization is of second order in the spatial stepsize $\mathcal{O}(h^2)$, where h is the characteristic step size of the grid. Hence, as the grid is refined by a factor of two the numerical error should decrease by a factor of four. The range of gridsizes $n \times n$ that will be considered is $n \in \{5, 10, 20, 40, 80, 160, 320\}$ and a uniform grid will be used. To observe this scaling behaviour care should be taken such that the numerical error is dominated by errors of spatial origin. This is realized by using a fourth order explicit Runge-Kutta time-integrator with a temporal stepsize of $\Delta t = 0.001$ such that the temporal integration error is practically negligible. The value of the kinematic viscosity considered is $\nu = 0.01$. Finally, the numerical error $\epsilon_x(t) : \mathbb{R}^+ \to \mathbb{R}^+$ will be measured in the $\infty-$norm as:

$$\epsilon_x(t) = ||\boldsymbol{u}_{TG,h}(t) - \boldsymbol{u}_h(t)||_\infty ,$$

where $\boldsymbol{u}_{TG,h}(t) : \mathbb{R}^+ \to \mathbb{R}^N$ is a function that provides the exact Taylor-Green vortex solution at time t as a vector of suitably ordered unknowns on a staggered grid with characteristic spatial stepsize h. Similarly $\boldsymbol{u}_h(t) : \mathbb{R}^+ \to \mathbb{R}^N$ is the numerical solution at time t on a staggered grid with characteristic spatial stepsize h.

The results of the spatial convergence study are provided in Figure 3.3. As predicted by the theory a scaling of $\mathcal{O}(h^2)$ is observed in the error. Based on this it is concluded that the spatial discretization has been implemented correctly.

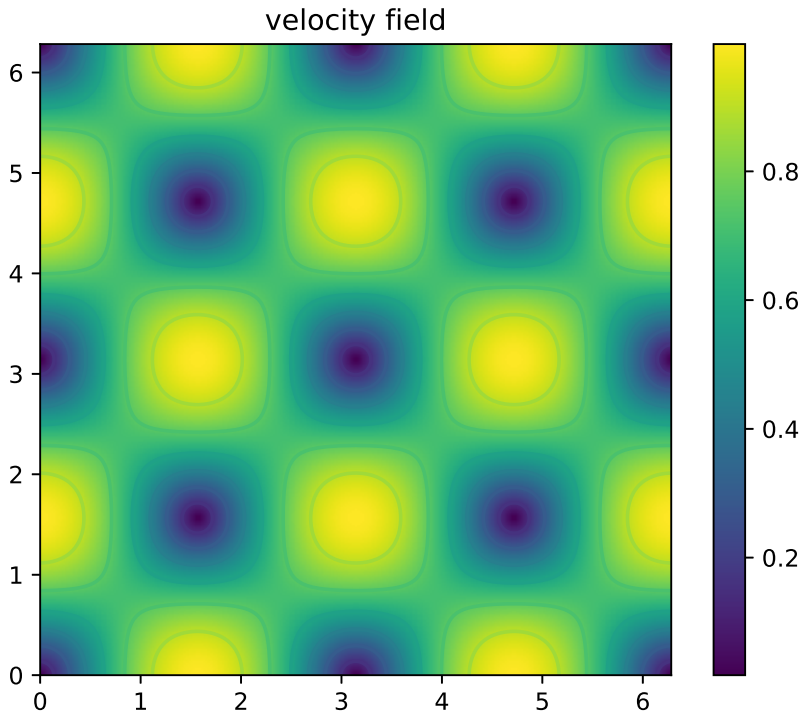


Figure 3.2: The exact absolute velocity field of the Taylor-Green vortex evaluated at $t = 1$ for $\nu = 0.001$.

3.4.3. Temporal Convergence Study

Similarly to the verification of the spatial discretization implementation, the correct implementation of the temporal integration schemes is also verified using convergence analyses. Again, the Taylor-Green vortex will be used an analytical solution. The temporal error will be denoted as $\epsilon_t(t) : \mathbb{R}^+ \to \mathbb{R}^+$ and a range of $\Delta t \in \{0.1, 0.05, 0.025, 0.0125, 0.00625\}$ will be considered for the study. The following temporal integrators will be considered; the implicit midpoint method of theoretical order $\mathcal{O}(\Delta t^2)$, explicit RK4 of theoretical order $\mathcal{O}(\Delta t^4)$ and implicit GL4 of theoretical order $\mathcal{O}(\Delta t^4)$. Due to the high order of accuracy of most of these methods it is expected that the difference between the exact solution and numerical solution at a given instance in time will be dominated by errors of spatial origin. To obtain an accurate picture of the error from temporal sources the numerical solution will not be subtracted from the exact solution but from a numerical solution with a very short timestep size $\Delta t = 0.001$, this has the effect of eliminating contribution from spatial errors in $\epsilon_t(t)$. Thus, $\epsilon_t(t)$ is calculated as:

$$\epsilon_t(t) = ||\boldsymbol{u}_{h,\Delta t=0.001}(t) - \boldsymbol{u}_h(t)||_\infty ,$$

where $\boldsymbol{u}_{h,\Delta t=0.001}(t) : \mathbb{R}^+ \to \mathbb{R}^N$ is the numerical solution found using the fine timestep size $\Delta t = 0.001$. For efficiency a numerical grid of size 20×20 will be considered. Furthermore, the kinematic viscosity will be set at $\nu = 0.1$, the Newton-Raphson method combined with a direct linear solver will be used to solve nonlinear problems arising from implicit methods and time integration will be performed until $t = 0.5$.

The results of the temporal convergence study are provided in Figure 3.4. It can be observed that the predicted theoretical error behaviour is exhibited until machine precision effects take place. Based on this it is concluded that the temporal discretization has been implemented correctly.

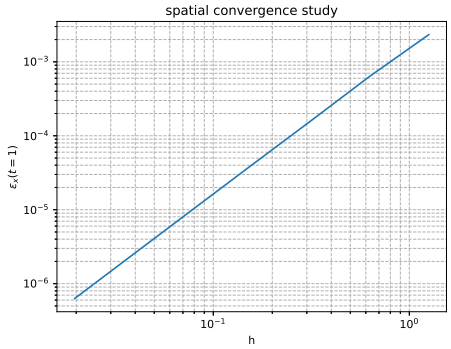


Figure 3.3: The numerical errors in the spatial convergence study using the Taylor-Green vortex measured in the ∞-norm as a function of the numerical grid's characteristic spatial stepsize h.

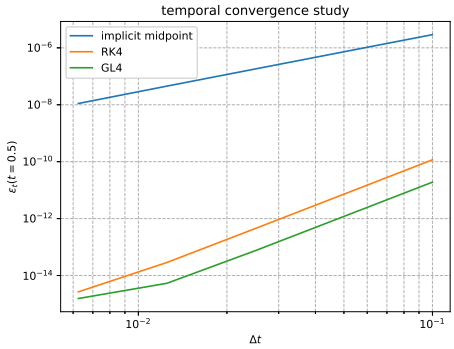


Figure 3.4: The numerical errors in the temporal convergence study using the Taylor-Green vortex measured in the ∞-norm as a function of the numerical grid's characteristic spatial stepsize h.

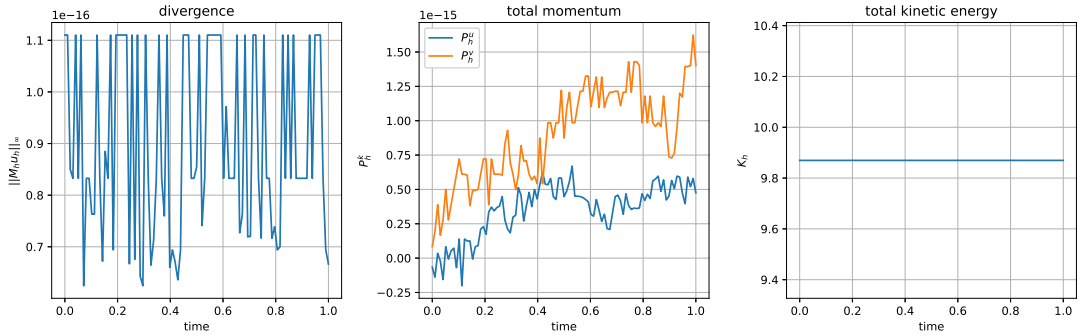


Figure 3.5: Conservation properties of the FOM.

3.4.4. FOM Conservation

The conservation properties of the FOM will also briefly be verified. This will be done by plotting the temporal evolution of both components of the discrete total momentum $\boldsymbol{P}_h(t)$ and the discrete total kinetic energy $K_h(t)$ for a simulation using the Taylor-Green vortex as an initial condition and $\nu = 0$. To verify that the discrete velocity field satisfies condition (3.2) the time evolution of $||M_h \boldsymbol{u}_h(t)||_\infty$ will additionally be analysed. When condition (3.2) is satisfied this quantity should be of the order of machine precision. The simulation will take place on a 20×20 grid for efficiency. The system will be integrated in time using the energy-conserving implicit midpoint method, where integration takes place until $t = 1$ using a timestep of size $\Delta t = 0.01$. The nonlinear system arising from application of the implicit midpoint method is solved using the Newton-Raphson method combined with a direct linear solver.

The results are provided in Figure 3.5. It can be observed that $||M_h \boldsymbol{u}_h(t)||_\infty$ and both components of $\boldsymbol{P}_h(t)$ have values of the order of machine precision and that K_h remains steadily at the value of its initial condition. This verifies that the implementation conserves all theoretically conserved quantities.

4

A Structure-Preserving hROM of the Incompressible Navier-Stokes Equations

In this chapter a structure-preserving hyper-reduced order model (hROM) of the incompressible Navier-Stokes equations will be constructed based on the structure-preserving FOM discussed in the previous chapter. The hROM is constructed by introducing so-called structure-preserving hyper-reduction methods to the structure-preserving reduced order model (ROM) proposed in [86]. The structure-preserving POD-Galerkin ROM proposed in [86] will be described, alongside a more general and brief introduction to the POD-Galerkin method. Subsequently hyper-reduction will be introduced and the need for efficient, structure-preserving hyper-reduction methods will be discussed. Then, three structure-preserving hyper-reduction methods will be proposed and their energy-conserving properties will be analysed. These methods constitute the main novelty of this research.

4.1. Structure-Preserving Model Reduction

In [86] a nonlinearly stable structure-preserving ROM is proposed that is based on the structure-preserving FOM described in the previous chapter. The ROM is constructed using a POD-Galerkin method. Here POD stands for proper orthogonal decomposition and Galerkin refers to Galerkin-projection, which is the method used to arrive at a reduced order (approximate) formulation of the dynamical system in equation (3.1). The ROM is pressure-free, meaning that the effect of the pressure variable $\boldsymbol{p}_h$ in (3.1) is exactly eliminated in the ROM. In the following a brief introduction is provided to POD-Galerkin model reduction and after this the structure-preserving ROM as proposed in [86] will be introduced.

4.1.1. The Reduced Space

Fundamentally, the POD-Galerkin method revolves around restricting the solution of a ROM to evolve strictly in a low-dimensional linear subspace $\mathcal{V} \subset \mathbb{R}^N$, referred to as the reduced space. Here, $\dim(\mathcal{V}) = r \ll N$. Given such a linear subspace $\mathcal{V}$, a linear basis $\Phi \in \mathbb{R}^{N \times r}$ may be found such that:

$$\text{span}(\Phi) = \mathcal{V}.$$

Now, using the POD-Galerkin method, the approximation is made that the FOM solution $\boldsymbol{u}_h \in \mathbb{R}^N$ can be well approximated with a ROM solution $\boldsymbol{u}_r$ as:

$$\boldsymbol{u}_h(t) \approx \boldsymbol{u}_r(t) = \Phi \boldsymbol{a}(t) \in \mathcal{V}, \tag{4.1}$$

where $\boldsymbol{a}(t) : \mathbb{R}^+ \to \mathbb{R}^r$ are generalized coordinates of the subspace $\mathcal{V}$. There are many well-known methods to construct a basis Φ: one may consider e.g. (pseudo-)spectral methods [83]. However, using the POD-Galerkin method an optimal, data-driven basis is constructed using the POD. Here optimality is in the sense that a certain reconstruction error is minimized as will be seen later. The basis Φ is constructed at the hand of a snapshot matrix $X \in \mathbb{R}^{N \times n_s}$

which may be considered as n_s equidistant-in-time samples of the FOM solution manifold $\mathcal{M}_u^h$ defined as:

$$\mathcal{M}_u^h := \{\boldsymbol{u}_h(t) \mid t \in [0,T]\},$$

hence X takes the form:

$$X = \left[\boldsymbol{u}_h(t^0), \boldsymbol{u}_h(t^0+\Delta t), ..., \boldsymbol{u}_h(t^0+(n_s-1)\Delta t)\right].$$

Note that this notion can easily be extended to a case where the model parameters (like ν) are varying by adding dimensions to the solution manifold $\mathcal{M}_u^h$ corresponding to the parameter values. Nonetheless, this chapter will not consider such extensions. To obtain a good approximation (4.1), $\mathcal{V}$ should contain as much as possible of $\mathcal{M}_u^h$. Namely, in this case the ROM is most capable of reproducing the FOM solutions, as most elements of $\mathcal{M}_u^h$ then lie in the space $\mathcal{V}$ to which $\boldsymbol{u}_r$ is restricted. The POD finds a basis Φ, referred to as the POD basis, in accordance with this objective by posing the following optimization problem [101]:

$$\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_r\} = \underset{\widetilde{\boldsymbol{\phi}}_1,...,\widetilde{\boldsymbol{\phi}}_r \in \mathbb{R}^N}{\arg\max} \sum_{i=1}^{n_s}\sum_{j=1}^{r} \left|\left\langle X_{,i}, \widetilde{\boldsymbol{\phi}}_j \right\rangle\right|^2 \quad \text{s.t.} \quad \left\langle \widetilde{\boldsymbol{\phi}}_k, \widetilde{\boldsymbol{\phi}}_l \right\rangle = \delta_{kl} \;\forall\; k,l \in \{1,...,r\}, \tag{4.2}$$

where $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_r\}$ form the columns of Φ, $X_{,i}$ denotes the i^{th} column of X and δ_{ij} denotes the Kronecker delta function. The orthogonality constraint is introduced to obtain a unique minimizer to the optimization, as the solution to the unconstrained optimization may be shown to be non-unique [101]. Intuitively, this optimization provides a set of vectors, referred to as POD modes, which can optimally reconstruct the data in X in an average sense. This is a result of the fact that the projection of the columns of X onto these POD modes is maximal and thus that the POD modes form a good orthogonal basis to X. The solution of optimization problem (4.2) is given by the first r left singular vectors of the singular value decomposition (SVD) of X:

$$X = \hat{\Phi}\Sigma\Psi^T.$$

A proof of this statement can be found in [101]. Here, $\hat{\Phi} \in \mathbb{R}^{N\times N}$ is a matrix containing the eigenvectors of the correlation matrix XX^T as columns, referred to as left singular vectors, $\Psi \in \mathbb{R}^{n_s\times n_s}$ is a matrix containing the eigenvectors of the correlation matrix X^TX as columns, referred to as right singular vectors and $\Sigma \in \mathbb{R}^{N\times n_s}$ is a matrix with the following block structure:

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix},$$

where $D = \text{diag}(\sigma_1, ..., \sigma_{d_r}) \in \mathbb{R}^{d_r\times d_r}$ is a diagonal matrix with diagonal entries satisfying $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{d_r} > 0$, referred to as singular values. Furthermore d_r is the rank of X. Thus, to obtain an r-dimensional POD basis the first r columns of $\hat{\Psi}$ are used, where $r \leq d_r$. In addition, it may be shown ([101]) that the POD basis can also be found from the following minimization:

$$\Phi = \underset{\widetilde{\Phi}\in\mathbb{R}^{N\times r}}{\arg\min} \left|\left|X - \widetilde{\Phi}\widetilde{\Phi}^T X\right|\right|_F^2 \quad \text{s.t.} \quad \widetilde{\Phi}^T\widetilde{\Phi} = I. \tag{4.3}$$

The solution of minimization problem (4.3) is also the SVD of X and is captured in the Schmidt-Eckart-Young-Mirsky theorem [7]. In minimization problem (4.3), $||A||_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |(A)_{ij}|^2$, $A \in \mathbb{R}^{m\times n}$ denotes the Frobenius matrix norm. It is clear that minimization problem (4.3) minimizes the difference between X and the reconstruction of the columns of X in the POD basis measured using the Frobenius norm.

4.1.2. Galerkin Projection

After finding the POD basis using the SVD of the snapshot matrix X, the approximation $\boldsymbol{u}(t) \approx \boldsymbol{u}_r(t) = \Phi\boldsymbol{a}(t)$ can be substituted into the dynamical system of interest. The general case of equation (3.15) will be considered for the purpose of this introduction. After substitution of the approximation $\boldsymbol{u}_r(t)$ into equation (3.15), the general dynamical system takes the form:

$$\frac{d\Phi\boldsymbol{a}}{dt} = \Phi\frac{d\boldsymbol{a}}{dt} = f(\Phi\boldsymbol{a}(t)) + \boldsymbol{r}(t), \tag{4.4}$$

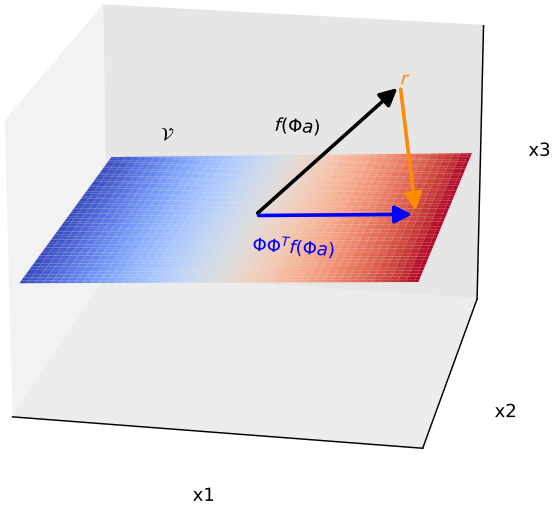


Figure 4.1: An orthogonal projection of $f(\Phi \boldsymbol{a})$ onto $\mathcal{V}$ resulting in an approximation $\Phi\Phi^T f(\Phi \boldsymbol{a})$, depicted for $n = 3$ and $r = 2$.

note the presence of the residual vector $\boldsymbol{r}(t) : \mathbb{R}^+ \to \mathbb{R}^n$. This is a result of the left-hand side of the second equality being an element of the reduced space, $\Phi \frac{d\boldsymbol{a}}{dt} \in \mathcal{V}$, whereas the output space of the function f on the right-hand side is not necessarily an element of $\mathcal{V} \subset \mathbb{R}^n$. Rather, the function f on the right-hand side evaluated for $\boldsymbol{u}_r$ is an element of the larger state space $\mathbb{R}^n$, since $f(\Phi \boldsymbol{a}(t)) : \mathbb{R}^+ \to \mathbb{R}^n$. Hence, to assure that the right-hand side of (4.4) is also an element of $\mathcal{V}$, the residual vector $\boldsymbol{r}(t)$ is introduced. However, as it stands there is no explicit formula for this residual and the system (4.4) is highly over-determined. To resolve this, the system (4.4) is projected onto $\mathcal{V}$ through the Galerkin projection. The Galerkin projection can be interpreted as an orthogonal projection onto $\mathcal{V}$. An illuminating example for $n = 3$ and $r = 2$ is provided in Figure 4.1. By projecting orthogonally onto $\mathcal{V}$ the residual $\boldsymbol{r}(t)$ as in (4.4) is minimized. Mathematically, the Galerkin projection is performed by taking the inner product with the columns of Φ:

$$\Phi^T \Phi \frac{d\boldsymbol{a}}{dt} = \frac{d\boldsymbol{a}}{dt} = \Phi^T f(\Phi \boldsymbol{a}(t)). \tag{4.5}$$

Note that $\Phi^T \boldsymbol{r}(t) = 0$, since the difference between a vector and its orthogonal projection onto a linear subspace is orthogonal to the respective linear subspace, a defining feature of Galerkin projections. Note also that in the first equality it was used that the columns of Φ are orthogonal. Having projected (4.4) onto $\mathcal{V}$ to obtain (4.5) an r-dimensional dynamical system is found, constituting a ROM of (3.15). The r-dimensional system (4.5) can now be evolved in time to some time $t = t^n$ to obtain the generalized coordinates $\boldsymbol{a}(t^n)$ of a reduced order approximation $\boldsymbol{u}_r(t^n)$ to the FOM solution $\boldsymbol{u}_h(t^n)$, for an initial condition $\boldsymbol{u}_0 \in \mathcal{V}$.

4.1.3. A Structure-Preserving ROM of the incompressible Navier-Stokes equations

Having introduced the POD-Galerkin method, the structure-preserving ROM of the incompressible Navier-Stokes equations as proposed in [86] can be introduced. The ROM has three key-properties to preserve the structure of (3.1) and (2.13). Namely: d extra modes are inserted into the r-dimensional POD basis that are specifically constructed such that the reduced analogue to total momentum is conserved; the altered POD basis is discretely divergence-free; the skew-symmetry of the operator $\widetilde{C}_h(\boldsymbol{u})$, where $C_h(\boldsymbol{u}) = \widetilde{C}_h(\boldsymbol{u})\boldsymbol{u}$, and the negative-definiteness and symmetry of the diffusion operator D_h are preserved on the reduced level.

Without specifying the exact construction of the altered POD basis $\Phi \in \mathbb{R}^{N\times r}$ yet, the approximation $\boldsymbol{u}_r(t) = \Phi\boldsymbol{a}(t)$ is inserted into (3.1)-(3.2), resulting into:

$$\Omega_h\Phi\frac{d\boldsymbol{a}}{dt} + C_h(\Phi\boldsymbol{a}) = -G_h\boldsymbol{p}_h + \nu D_h\Phi\boldsymbol{a} + \boldsymbol{r} \tag{4.6}$$

$$M_h\Phi\boldsymbol{a} = 0. \tag{4.7}$$

Where $\boldsymbol{r}(t) : \mathbb{R}^+ \to \mathbb{R}^N$ is a residual vector such that both sides of the equality in (4.6) are elements of the reduced space $\mathcal{V}$. [86] now proposes to generalize the orthogonality conditions in the construction of the POD basis from orthogonality in the Euclidean inner product to orthogonality in the Ω_h-inner product. This has as a result that the Galerkin projection of (4.6) onto Φ can be taken immediately to obtain an r-dimensional evolution equation for $\boldsymbol{a}$:

$$\Phi^T\Omega_h\Phi\frac{d\boldsymbol{a}}{dt} = \frac{d\boldsymbol{a}}{dt} = -\Phi^T C_h(\Phi\boldsymbol{a}) - \Phi^T G_h\boldsymbol{p}_h + \nu\Phi^T D_h\Phi\boldsymbol{a}. \tag{4.8}$$

The residual as defined by the difference between (4.6) and (4.8) multiplied by Φ is then orthogonal to $\mathcal{V}$, i.e. $\Phi^T\boldsymbol{r} = 0$. Furthermore, if the individual modes of the altered POD basis are discretely divergence-free, $\boldsymbol{u}_r$ satisfies (4.7). This is the case due to linearity of both the discrete divergence operator M_h and the approximation $\boldsymbol{u}_r = \Phi\boldsymbol{a}$. In addition to satisfying (4.7), the ROM as in (4.8) will also become pressure-free. Following [86], this may be shown by analysing the projected pressure term in (4.8):

$$-\Phi^T G_h\boldsymbol{p}_h = \Phi^T M_h^T\boldsymbol{p}_h = (M_h\Phi)^T\boldsymbol{p}_h = 0,$$

where the duality of the discrete gradient and divergence of the FOM were leveraged. Under these conditions (Ω_h-orthogonality and mode-wise divergence-freeness) the ROM will take the form:

$$\frac{d\boldsymbol{a}}{dt} = -\Phi^T C_h(\Phi\boldsymbol{a}) + \nu\Phi^T D_h\Phi\boldsymbol{a}, \tag{4.9}$$

where (4.7) is satisfied for all times t by construction.

The following operators are now defined:

$$C_r(\boldsymbol{a}) := \Phi^T C_h(\Phi\boldsymbol{a}) \tag{4.10}$$

$$\widetilde{C}_r(\boldsymbol{a}_1)\boldsymbol{a}_2 := \Phi^T\widetilde{C}_h(\Phi\boldsymbol{a}_1)\Phi\boldsymbol{a}_2 \tag{4.11}$$

$$D_r\boldsymbol{a} := \Phi^T D_h\Phi\boldsymbol{a}, \tag{4.12}$$

where $C_r(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^r$ is referred to as the reduced convection operator, $\widetilde{C}_r(\boldsymbol{a}_1)\boldsymbol{a}_2 : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}^r$ is referred to as the reduced quasi-linear convection operator and $D_r \in \mathbb{R}^{r\times r}$ is referred to as the reduced diffusion operator. Additionally, it holds that:

$$C_r(\boldsymbol{a}) = \widetilde{C}_r(\boldsymbol{a})\boldsymbol{a},$$

for the reduced convection and quasi-linear convection operators. Relation (4.9) may now be written in terms of these reduced operators as follows:

$$\frac{d\boldsymbol{a}}{dt} = -C_r(\boldsymbol{a}) + \nu D_r\boldsymbol{a} = -\widetilde{C}_r(\boldsymbol{a})\boldsymbol{a} + \nu D_r\boldsymbol{a}. \tag{4.13}$$

When time-integration of (4.13) is performed using implicit methods, like the energy-conserving Gauss-Legendre family of Runge-Kutta methods discussed in the previous chapter, it will be required to evaluate the Jacobian of the reduced convection operator. This Jacobian can be expressed in terms of the Jacobian of the FOM operator $C_h(\boldsymbol{u})$, a derivation of this is given in section B.1. Having performed the derivation, the following can be written for the Jacobian of the reduced convection operator:

$$J_r(\boldsymbol{a}) = \frac{\partial C_r}{\partial \boldsymbol{a}}(\boldsymbol{a}) = \Phi^T J_h(\Phi \boldsymbol{a})\Phi.$$

Using this expression the Newton-Raphson algorithm can be performed at the reduced level to solve nonlinear systems of equations resulting from the application of implicit time-integration schemes.

Momentum

The first conservation property that will be considered is the conservation of total momentum at the reduced level. Analogously to the discrete total momentum $\boldsymbol{P}_h(t)$, the reduced total momentum $\boldsymbol{P}_r(t) : \mathbb{R}^+ \to \mathbb{R}^d$ will be defined based on the reduced velocity vector $\boldsymbol{u}_r$:

$$P_r^u(t) := \boldsymbol{e}_u^T \Omega_h \boldsymbol{u}_r = \boldsymbol{e}_u^T \Omega_h \Phi \boldsymbol{a} \tag{4.14}$$

$$P_r^v(t) := \boldsymbol{e}_v^T \Omega_h \boldsymbol{u}_r = \boldsymbol{e}_v^T \Omega_h \Phi \boldsymbol{a}, \tag{4.15}$$

where the vectors $\boldsymbol{e}_u, \boldsymbol{e}_v$ are those used in (4.39)-(4.40) and $d = 2$. The temporal evolution equations of the reduced momentum components P_r^u, P_r^v are found in similar fashion to (4.41)-(3.13), i.e. by temporal differentiation. Doing so, results in the following evolution equations:

$$\frac{dP_r^u}{dt} = \boldsymbol{e}_u^T \Omega_h \Phi \frac{d\boldsymbol{a}}{dt} = \boldsymbol{e}_u^T \Omega_h \Phi \left[-C_r(\boldsymbol{a}) + \nu D_r \boldsymbol{a}\right] \tag{4.16}$$

$$\frac{dP_r^v}{dt} = \boldsymbol{e}_v^T \Omega_h \Phi \frac{d\boldsymbol{a}}{dt} = \boldsymbol{e}_v^T \Omega_h \Phi \left[-C_r(\boldsymbol{a}) + \nu D_r \boldsymbol{a}\right]. \tag{4.17}$$

Here, (4.13) was used in the right-hand side of the second equality. To preserve structure at the reduced level, both (4.16) and (4.17) have to equal zero for periodic boundary conditions, analogously to (2.9). In [86] this property is attained by making sure the telescoping property of the discrete FOM operators can still be invoked. In detail, rewriting (4.16)-(4.17) to be expressed in FOM operators as:

$$\frac{dP_r^u}{dt} = \boldsymbol{e}_u^T \Omega_h \Phi \Phi^T \left[-C_h(\Phi \boldsymbol{a}) + \nu D_h \Phi \boldsymbol{a}\right] \tag{4.18}$$

$$\frac{dP_r^v}{dt} = \boldsymbol{e}_v^T \Omega_h \Phi \Phi^T \left[-C_h(\Phi \boldsymbol{a}) + \nu D_h \Phi \boldsymbol{a}\right], \tag{4.19}$$

the telescoping property can be invoked to set the right-hand side of (4.18)-(4.19) to zero if:

$$\boldsymbol{e}_u^T \Omega_h \Phi \Phi^T = \boldsymbol{e}_u^T \tag{4.20}$$

$$\boldsymbol{e}_v^T \Omega_h \Phi \Phi^T = \boldsymbol{e}_v^T. \tag{4.21}$$

These conditions imply that the Ω_h-orthogonal altered POD basis Φ has to exactly embed the vectors $\boldsymbol{e}_u, \boldsymbol{e}_v$. One may see this by considering the action of $\Phi\Phi^T\Omega_h$ on an element of $\mathcal{V}$ denoted by $\boldsymbol{v} = \Phi \boldsymbol{a}_v \in \text{span}(\Phi)$ and taking the transpose:

$$\boldsymbol{v}^T \Omega_h \Phi \Phi^T = (\Phi \Phi^T \Omega_h \boldsymbol{v})^T = (\Phi \Phi^T \Omega_h \Phi \boldsymbol{a}_v)^T = (\Phi \boldsymbol{a}_v)^T = \boldsymbol{v}^T.$$

Indeed, if $\boldsymbol{e}_u, \boldsymbol{e}_v \in \text{span}(\Phi)$, conditions (4.20) and (4.21) are satisfied. In [86] this is achieved by altering the procedure to construct Φ to explicitly include $\boldsymbol{e}_u, \boldsymbol{e}_v$ in the columns of Φ. Note that the Ω_h-orthogonality of the POD basis can be satisfied since $\boldsymbol{e}_u^T \Omega_h \boldsymbol{e}_v = 0$ holds. Furthermore, because the vectors $\boldsymbol{e}_u, \boldsymbol{e}_v$ have ones at indices corresponding to u and v-unknowns respectively and zeros otherwise, they can be thought of as constant velocity fields. Therefore they are discretely divergence-free. Thus to obtain (4.13) and conserve $\boldsymbol{P}_r$, the conventional

POD procedure should be altered to produce both Ω_h-orthogonal and discretely divergence-free modes. Additionally, they should span $\boldsymbol{e}_u, \boldsymbol{e}_v$ by e.g. explicitly containing the Ω_h-orthogonal and divergence-free vectors $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$ in the set of modes like [86].

The procedure to construct such a basis is described in [86] and uses the work in [104]. Defining the matrix $E \in \mathbb{R}^{N\times d}$ as:

$$E := \left[\frac{\boldsymbol{e}_u}{||\boldsymbol{e}_u||_{\Omega_h}}, \frac{\boldsymbol{e}_v}{||\boldsymbol{e}_v||_{\Omega_h}}\right],$$

the procedure can be understood quite intuitively. Firstly, the components of the snapshot matrix X that are non-Ω_h-orthogonal to the Ω_h-orthogonal basis modes in E are subtracted from the data set X. This results in a remainder $\widetilde{X}$ that is Ω_h-orthogonal to the modes in E. Secondly, the POD is applied to this remainder $\widetilde{X}$ where the orthogonality condition is generalized to Ω_h-orthogonality. The solution to this generalized POD procedure is well-known and described in e.g. [101]. It still has a sense of optimality as it solves the minimization ([86, 101]):

$$\widetilde{\Phi} = \underset{\Phi^*\in\mathbb{R}^{N\times m}}{\arg\min} \left|\left|\widetilde{X} - \Phi^*\Phi^{*T}\Omega_h\widetilde{X}\right|\right|_F^2 \quad \text{s.t.} \quad \Phi^{*T}\Omega_h\Phi^* = I.$$

Finally, concatenating the Ω_h-orthogonal modes in E and the first $r-2$ POD basis modes found from the generalized POD procedure, a complete basis is obtained. Indeed, E forms a basis for the data set $X - \widetilde{X}$ and the other $r-2$ POD basis modes form an approximate basis set for $\widetilde{X}$. Together they can approximately span X, given $r < d_r$. The complete algorithm is given in Algorithm 1.

Algorithm 1 Algorithm to determine Ω_h-orthogonal and divergence-free POD basis explicitly containing the columns of matrix E.

1: $\widetilde{X} = X - EE^T\Omega_h X$ ▷ Remove Ω_h-projection of X on span(E) from data
2: $\hat{X} = \Omega_h^{1/2}\widetilde{X}$ ▷ Transformation for Ω_h-orthogonality
3: $\hat{X} = \hat{\Phi}\hat{\Sigma}\hat{\Psi}^T$ ▷ SVD truncated at $r-2$
4: $\widetilde{\Phi} = \Omega_h^{-1/2}\hat{\Phi}$ ▷ Transformation for Ω_h-orthogonality
5: $\Phi = \left[E, \widetilde{\Phi}\right]$ ▷ Concatenate E and $\widetilde{\Phi}$

The proof that the resulting POD basis is divergence-free is given in [86]. It requires considering the eigenvector problem underlying the SVD in step 3 of Algorithm 1, transforming that to $\widetilde{\Phi}$ and showing the discrete divergence of this expression equals zero everywhere on the grid. For completeness a proof is provided in Appendix C.

Summarizing, to state relation (4.13) requires that the POD-basis Φ is divergence-free for every individual mode and that the modes are mutually Ω_h-orthogonal. If it is furthermore desired to conserve reduced total momentum $\boldsymbol{P}_r(t)$ conditions (4.20)-(4.21) must hold. The previous conditions are satisfied if $\boldsymbol{e}_u, \boldsymbol{e}_v \in$ span(Φ), a condition which is in turn satisfied by including them explicitly in the columns of Φ. Algorithm 1 is a method to construct such an Ω_h-orthogonal POD basis that explicitly contains $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$. Considering the eigenvector problem underlying the SVD in Algorithm 1, it can be shown that the resulting altered POD basis is divergence-free. Since, the set of the columns of E combined with the POD modes constructed in Algorithm 1 are mutually Ω_h-orthogonal, Φ is Ω_h-orthogonal. Finally, it can therefore be stated for relation (4.13) that:

$$\frac{dP_r^u}{dt} = 0 \tag{4.22}$$

$$\frac{dP_r^v}{dt} = 0, \tag{4.23}$$

meaning that total reduced momentum is conserved. In addition, (4.7) is satisfied by construction.

Kinetic energy

The last conserved quantity to be addressed is the total kinetic energy at the reduced level K_r. The reduced total kinetic energy will be defined by inserting the reduced velocity vector $\boldsymbol{u}_r$ in the definition of discrete total kinetic energy K_h as follows:

$$K_r := \frac{1}{2}\left|\left|\boldsymbol{u}_r\right|\right|^2_{\Omega_h} = \frac{1}{2}\left\langle \boldsymbol{u}_r, \Omega_h \boldsymbol{u}_r \right\rangle = \frac{1}{2}\left\langle \Phi\boldsymbol{a}, \Omega_h \Phi\boldsymbol{a} \right\rangle = \frac{1}{2}\boldsymbol{a}^T\Phi^T\Omega_h\Phi\boldsymbol{a} = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} = \frac{1}{2}\left|\left|\boldsymbol{a}\right|\right|^2. \quad (4.24)$$

Note that K_r is proportional to the squared Euclidean norm of $\boldsymbol{a}$. This has as a significant result that when K_r is bounded, the ROM (4.13) is nonlinearly stable in the generalized coordinates $\boldsymbol{a}$. Indeed this would be the case if the evolution of K_r satisfied an analogue to (2.20) i.e. by preserving structure at the reduced level. An evolution equation can be derived by taking temporal derivatives of (4.24):

$$\begin{aligned} \frac{dK_r}{dt} = \frac{1}{2}\frac{d}{dt}\left\langle \boldsymbol{a}, \boldsymbol{a} \right\rangle &= \left\langle \boldsymbol{a}, \frac{d\boldsymbol{a}}{dt} \right\rangle \\ &= \left\langle \boldsymbol{a}, -\widetilde{C}_r(\boldsymbol{a})\boldsymbol{a} + \nu D_r \boldsymbol{a} \right\rangle \\ &= -\boldsymbol{a}^T\widetilde{C}_r(\boldsymbol{a})\boldsymbol{a} + \nu\boldsymbol{a}^T D_r\boldsymbol{a}. \end{aligned} \quad (4.25)$$

The last line in (4.25) may be analysed term by term to check if K_r is conserved in the inviscid limit like in (2.20). Firstly, the leftmost term will be analysed. Rewriting the reduced quasi-linear convection operator in terms of FOM operators it holds that:

$$\widetilde{C}_r(\boldsymbol{a}_c)^T = (\Phi^T\widetilde{C}_h(\Phi\boldsymbol{a}_c)\Phi)^T = \Phi^T\widetilde{C}_h(\Phi\boldsymbol{a}_c)^T\Phi = -\Phi^T\widetilde{C}_h(\Phi\boldsymbol{a}_c)\Phi = -\widetilde{C}_r(\boldsymbol{a}_c).$$

Here $\boldsymbol{a}_c \in \mathbb{R}^r$ are generalized coordinates of the convecting reduced velocity vector. Hence, the reduced quasi-linear convection operator is also skew-symmetric. Therefore the first term in the last line of (4.25) satisfies:

$$\boldsymbol{a}^T\widetilde{C}_r(\boldsymbol{a})\boldsymbol{a} = 0,$$

due to skew-symmetry. The reduced diffusion operator in the second term of the last line of (4.25) can also be rewritten in terms of FOM operators to obtain:

$$D_r = \Phi^T D_h \Phi = -\Phi^T Q_h^T Q_h \Phi = -(Q_h\Phi)^T(Q_h\Phi) = -Q_r^T Q_r,$$

where $Q_r := Q_h\Phi$. The second term in the last line of (4.25) now satisfies:

$$\boldsymbol{a}^T D_r \boldsymbol{a} = -\boldsymbol{a}^T Q_r^T Q_r \boldsymbol{a} = -\left|\left|Q_r\boldsymbol{a}\right|\right|^2 \leq 0.$$

Substituting the results for both terms into (4.25) results in the following evolution equation for the reduced total kinetic energy:

$$\frac{dK_r}{dt} = -\nu\left|\left|Q_r\boldsymbol{a}\right|\right|^2 \leq 0, \quad (4.26)$$

which is a reduced analogue to (2.20). Indeed, K_r is conserved in the inviscid limit and in the presence of nonzero ν it remains bounded following:

$$K_r(t) \leq K_r(0),$$

as from (4.26) it is clear the K_r is monotonically decreasing. Equivalently, the norm of the generalized coordinates satisfies:

$$\left|\left|\boldsymbol{a}(t)\right|\right| \leq \left|\left|\boldsymbol{a}_0\right|\right|,$$

where $\boldsymbol{a}_0$ are generalized coordinates of the ROM approximation of the initial condition. For this reason the ROM proposed by [86] as in (4.13) is nonlinearly stable and conserves mass, reduced total momentum and reduced total kinetic energy (in the inviscid limit).

4.2. Hyper-Reduction

The ROM (4.13) as constructed in the previous section is r-dimensional as it is a system of r variables. However, as it stands, evaluating (4.13) strictly as it is represented in the equation requires a computational effort that scales with the FOM dimensions N. This will be the case for every newly calculated value of $\boldsymbol{a}$ in a simulation of (4.13). This is a result of:

- having to compute $\boldsymbol{u}_r = \Phi\boldsymbol{a}$ explicitly, which scales computationally as $\mathcal{O}(rN)$;
- having to evaluate $C_h(\boldsymbol{u}_r)$, which scales computationally as $\mathcal{O}(N)$;
- having to perform the Galerkin projection $\Phi^T C_h(\boldsymbol{u}_R)$, which scales computationally as $\mathcal{O}(rN)$.

Therefore, the ROM is not truly low-dimensional. It is the objective of so-called hyper-reduction methods to eliminate the dependence of the computational effort on the FOM dimensions N. This may be achieved by either finding exact formulations of the reduced convection operator such that there is no computational scaling with N or by making suitable approximations that are low-dimensional. Two such methods will be discussed in this section. Namely, the exact tensor decomposition, a method that results in an exact low-dimensional representation of the nonlinearity and the discrete empirical interpolation method (DEIM) [29], a method constructing a low-dimensional approximation to the nonlinearity.

Both methods will require some preliminary operations having a computational effort proportional to a power of N. However, it is the aim of hyper-reduction methods to construct algorithms where these operations can be performed on a one-time basis before the simulation of (4.13) starts. The phase of the simulation of a ROM where these preliminary computations are performed is known in the reduced order modelling community as the offline phase of the simulation. In turn, the low-dimensional computations to find new values of the generalized coordinates $\boldsymbol{a}$ by time integrating (4.13) is referred to as the online phase. In fact, this so-called offline-online paradigm is not just associated to hyper-reduction but is generally applied to any ROM. Indeed, computation of the reduced diffusion operator $D_r = \Phi^T D_h \Phi$ is a calculation performed in the offline phase of the simulation referred to as a pre-computation. The calculation of the POD basis Φ is also a part of the offline-phase of the simulation and collecting FOM data in X may be considered part of that as well.

4.2.1. The Exact Tensor Decomposition

The exact tensor decomposition leverages the simple quadratic nonlinearity of the convection operator. The method can exactly represent the nonlinearity after a Galerkin projection on Φ with a computational effort that solely scales as a function of the reduced space dimensions r. It is derived by considering the entries of $\widetilde{C}_h(\Phi\boldsymbol{a})$ as in (3.8) (the terms between brackets). It is enough to consider one component of the matrix $\widetilde{C}_h(\Phi\boldsymbol{a})$, e.g. the component multiplying u_N in (3.8), to see how the decomposition works. Substituting the reduced velocity vector into the matrix component multiplying u_N in (3.8) results into:

$$\frac{1}{4}\left((\boldsymbol{u}_r)_{i_{NW}} + (\boldsymbol{u}_r)_{i_{NE}}\right) = \frac{1}{4}\left(\sum_{j=1}^{r} a_j \overline{\Phi}_{i_{NE},j} + \sum_{j=1}^{r} a_j \overline{\Phi}_{i_{NW},j}\right).$$

Here the bar over $\overline{\Phi}$ represents integration over appropriate pressure cell surfaces of the components of Φ in identical fashion to (3.8). The integers i_{NW} and i_{NE} are the indices of the velocity unknowns v_{NE} and v_{NW} in (3.8) in the reduced velocity vector respectively. The generalized coordinates can simply be taken outside of the brackets as follows:

$$\frac{1}{4}\left(\sum_{j=1}^{r} a_j \overline{\Phi}_{i_{NE},j} + \sum_{j=1}^{r} a_j \overline{\Phi}_{i_{NW},j}\right) = \sum_{j=1}^{r} a_j \frac{1}{4}\left(\overline{\Phi}_{i_{NE},j} + \overline{\Phi}_{i_{NW},j}\right). \tag{4.27}$$

Similar expressions can be found for the other nonzero components of $\widetilde{C}_h(\Phi\boldsymbol{a})$. Hence, the generalized coordinates can be taken outside of the full matrix, resulting in a sum of matrices

evaluated for individual modes multiplied by corresponding generalized coordinates. Thus, instead of calculating $\widetilde{C}_h(\Phi\boldsymbol{a})$ directly the following can also be computed:

$$\widetilde{C}_h(\Phi\boldsymbol{a}) = \sum_{j=1}^{r} a_j \widetilde{C}_h(\Phi_{,j}).$$

Indeed, the matrices $\widetilde{C}_h(\Phi_{,j})$ are computable during the offline phase of the simulation. Using the exact tensor decomposition the reduced quasi-linear convection operator can also be calculated as follows:

$$\widetilde{C}_r(\boldsymbol{a})\boldsymbol{a} = \sum_{j=1}^{r} a_j \Phi^T \widetilde{C}_h(\Phi_{,j})\Phi\boldsymbol{a}, \tag{4.28}$$

where the terms $\Phi^T\widetilde{C}_h(\Phi_{,j})\Phi \in \mathbb{R}^{r\times r}$ are computable during the offline phase. The scaling of the computational effort to evaluate the expression on the right-hand side of (4.28) is $\mathcal{O}(r^3)$. This is the case because r matrix-vector products between an $r\times r$ matrix, $\Phi^T\widetilde{C}_h(\Phi_{,j})\Phi$, and an r-dimensional vector, $\boldsymbol{a}$, are necessary to be computed. The order of precomputing all necessary $\Phi^T\widetilde{C}_h(\Phi_{,j})\Phi$ is $\mathcal{O}(N^2r^2)$, a cost noted in [86] to become prohibitively for large r. Especially when N is large due to high spatial resolutions of the FOM. For notational simplicity, the following formulation for the reduced convection operator in exact tensor decomposition form will be introduced:

$$C_r(\boldsymbol{a}\otimes\boldsymbol{a}) := \sum_{j=1}^{r} a_j \Phi^T \widetilde{C}_h(\Phi_{,j})\Phi\boldsymbol{a}. \tag{4.29}$$

The formulation $C_r(\boldsymbol{T})$ should be thought of as a tensor contraction over the indices k and j between the third order tensor $(\Phi^T\widetilde{C}_h(\Phi_{,k})\Phi)_{ij}$ and the second order tensor T_{kj}. The ROM as in (4.13) then takes the form:

$$\frac{d\boldsymbol{a}}{dt} = -C_r(\boldsymbol{a}\otimes\boldsymbol{a}) + \nu D_r\boldsymbol{a}. \tag{4.30}$$

This form is still identical to the formulations using the other two reduced convection operators, however, evaluating the right-hand side requires only $\mathcal{O}(r^3)$ operations instead of $\mathcal{O}(r\cdot N)$.

As the exact tensor decomposition is an exact representation of the reduced convection operator it inherits all its conservation properties. Thus, it is structure-preserving. However, using energy-conserving Runge-Kutta methods to temporally discretize (4.30) it will be necessary to find an expression for the Jacobian of $C_r(\boldsymbol{a}\otimes\boldsymbol{a})$. Considering (4.29) the Jacobian can be calculated as follows:

$$\left(\frac{\partial C_r(\boldsymbol{a}\otimes\boldsymbol{a})}{\partial\boldsymbol{a}}\right)_{ij} = \sum_{k,l=1}^{r} (\Phi^T\widetilde{C}_h(\Phi_{,k})\Phi)_{il}\left(a_l\delta_{kj} + a_k\delta_{lj}\right), \tag{4.31}$$

which will be denoted as $J_r(I\otimes\boldsymbol{a} + \boldsymbol{a}\otimes I)$, here I is the $r\times r$ identity tensor. The derivation of this expression is provided in section B.2. Evaluating (4.31) requires a computational effort of $\mathcal{O}(r^4)$. The relatively high computational cost of working with the exact tensor decomposition can be prohibitive for real-time and multi-query situations with slow Kolmogorov N-width decay and therefore high r. For this reason it may be of interest to use inexact hyper-reduction methods like the DEIM. In what follows an introduction to the DEIM will be provided.

4.2.2. The DEIM

Generally, precomputation is efficient when models are affine or linear. Namely, in this situation the Galerkin projection of the operator is known for any reduced velocity vector in advance, i.e. a linear operator $A\in\mathbb{R}^{N\times N}$ acting on the reduced velocity $\boldsymbol{u}_r$ has the Galerkin projection $\Phi^T A$ for any value of $\boldsymbol{u}_r$. Indeed, the Galerkin projection of a nonlinear operator $N(\boldsymbol{u}):\mathbb{R}^N\to\mathbb{R}^N$ acting on $\boldsymbol{u}_r$ also has the form $\Phi^T N(\boldsymbol{u}_r)$, however the explicit Galerkin projected operator can not be formed in a precomputation. It is the DEIM's objective to find an affine approximation of a nonlinear operator $N(\boldsymbol{u})$ [29, 30, 28]. The approximation is based on evaluating $N(\boldsymbol{u})$ in only a low-dimensional subset $\mathcal{P}_h\subset\mathcal{M}_h$ of nodes on the computational mesh $\mathcal{M}_h$. This subset

$\mathcal{P}_h$ is referred to as the measurement space and nodes that are elements of $\mathcal{P}_h$ are referred to as measurement points. The affinity of the approximation allows the Galerkin projection of the approximation to be precomputed.

For the discrete convection operator the DEIM approximation takes the form:

$$C_h(\boldsymbol{u}) \approx M\boldsymbol{c}(\boldsymbol{u}). \tag{4.32}$$

Here $M \in \mathbb{R}^{N\times m}$ is the affine basis for the DEIM approximation and $\boldsymbol{c}(\boldsymbol{u}) : \mathbb{R}^N \to \mathbb{R}^m$ are the generalized coordinates of the approximation to $C_h(\boldsymbol{u})$ in the DEIM basis M, referred to as the DEIM coordinates. Note the difference between M and the discrete divergence operator M_h. For low-dimensional approximations it holds that $\dim(\mathcal{P}_h) = m \ll N$. The DEIM may be considered as an algorithm that constructs a separate m-dimensional reduced space $\mathcal{M}_d \subset \mathbb{R}^N$ for approximation of the convection operator such that:

$$\text{span}(M) = \mathcal{M}_d.$$

Since $m \ll N$, determining $\boldsymbol{c}(\boldsymbol{u})$ as in (4.32) is a highly over-determined problem. Furthermore, if $C_h(\boldsymbol{u}) \notin \mathcal{M}_d$ exact equality in (4.32) cannot be attained. To solve this, the DEIM coordinates $\boldsymbol{c}(\boldsymbol{u})$ are calculated by solving equation (4.32) only in the measurement space. To this end a basis to the measurement space is introduced and denoted by $P \in \mathbb{R}^{N\times m}$, such that:

$$\text{span}(P) = \mathcal{P}_h.$$

The matrix P will be referred to as the measurement matrix and takes the form:

$$P^T = \begin{bmatrix} 0 & 0 & 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & 1 & 0 & \dots & \dots & \dots & 0 \end{bmatrix}.$$

The measurement matrix P consists of selected columns of the $N \times N$ identity matrix corresponding to the vector indices of the measurement points. This has a result that multiplication of a vector or matrix from the left with P^T is equivalent to evaluating the components or rows of the vector or matrix, respectively, in the measurement points exclusively. Calculating the nonlinearity in only the measurement points can then be written as $P^T C_h(\boldsymbol{u})$. In turn, the DEIM coordinates are found by solving:

$$P^T C_h(\boldsymbol{u}) = P^T M\boldsymbol{c}. \tag{4.33}$$

Note the exact equality, unlike (4.32). Indeed, the DEIM coordinates are chosen such that the DEIM approximation $M\boldsymbol{c}$ exactly corresponds to the nonlinear convection operator $C_h(\boldsymbol{u})$ evaluated in the measurement space $\mathcal{P}_h$. Solving for $\boldsymbol{c}$ gives the formula:

$$\boldsymbol{c}(\boldsymbol{u}) = (P^T M)^{-1} P^T C_h(\boldsymbol{u}). \tag{4.34}$$

As the DEIM's efficiency partly stems from evaluating $C_h(\boldsymbol{u})$ only in the measurement points in $\mathcal{P}_h$, the computational stencil underlying $C_h(\boldsymbol{u})$ should be cheap to evaluate. If this was not the case, then evaluating the nonlinearity in only a small set of measurement points would still be expensive. For the discrete convection operator that is used in this thesis (see equation (3.8)) the computational stencil is sufficiently small but this may not hold in more general cases (e.g. for integro-differential equations [19]).

Indeed, the DEIM constitutes an interpolation procedure where the measurement space forms the set of interpolation points and the DEIM basis M are interpolation modes. The Galerkin projection $\Phi^T C_h(\boldsymbol{u}_r)$ can now be approximated efficiently as:

$$\Phi^T C_h(\boldsymbol{u}_r) \approx \Phi^T M(P^T M)^{-1} P^T C_h(\boldsymbol{u}_r), \tag{4.35}$$

where $\Phi^T M \in \mathbb{R}^{r\times m}$ can be precomputed, $(P^T M)^{-1} \in \mathbb{R}^{m\times m}$ can be performed efficiently using e.g. a precomputed LU-decomposition and $P^T C_h(\boldsymbol{u}_r)$ requires to only evaluate the convection

operator in m nodes. Here, (4.34) is substituted into the DEIM approximation $M\boldsymbol{c}$ followed by a Galerkin projection. Alternatively, the DEIM can be thought of as an oblique projection of the nonlinearity $C_h(\Phi\boldsymbol{a})$ on $\mathcal{M}_d$ orthogonally to $\mathcal{P}_h$. The computational cost of evaluating (4.35) is then of order $\mathcal{O}(\max(rm, m^2))$, depending on if matrix-vector multiplication with $\Phi^T M$ or solving $(P^T M)^{-1} P^T C_h(\boldsymbol{u}_r)$ using an LU-solve procedure is most expensive.

The quality of the approximation $M\boldsymbol{c}$ for nodes that are not elements of $\mathcal{P}_h$ depends on how well both the basis M and the measurement space $\mathcal{P}_h$ are chosen. The choice of $\mathcal{P}_h$ and M are well-established in the literature [29]. Firstly, the choice of M will be considered. The DEIM basis is, similarly to the POD basis Φ, constructed from an SVD of a snapshot data set. However, instead of solution snapshots, these snapshots consist of operator evaluations at different equidistant time instances. The DEIM snapshot matrix $\Xi \in \mathbb{R}^{N \times n_s}$ is given by:

$$\Xi = \left[C_h(\boldsymbol{u}_h(t^0)), C_h(\boldsymbol{u}_h(t^0 + \Delta t)), ..., C_h(\boldsymbol{u}_h(t^0 + (n_s - 1)\Delta t))\right], \tag{4.36}$$

which can be obtained during the same simulation as the snapshots in the snapshots matrix X. The DEIM basis M is now chosen as the first m left singular vector of an SVD of Ξ. Equivalently to Φ, this choice of M is an optimal data-driven choice for a basis M in the sense that it solves the following minimization problem:

$$M = \underset{\tilde{M} \in \mathbb{R}^{N \times m}}{\arg\min} \left|\left|\Xi - \tilde{M}\tilde{M}^T \Xi\right|\right|_F^2 \quad \text{s.t.} \quad \tilde{M}^T \tilde{M} = I.$$

This choice is quite natural in the current data-driven setting and, when combined with a principled choice of measurement points, may nearly solve the above minimization problem [29, 28] based on evaluating $C_h(\boldsymbol{u})$ only in $\mathcal{P}_h$.

The choice of measurement space following the DEIM algorithm was first proposed in [29]. It is based on a greedy algorithm that iteratively builds P one column at a time. Following [29] the measurement points are found from Algorithm 2. In this algorithm, $\boldsymbol{\xi}_i$ is the i^{th} left singular vector of the SVD of Ξ, p_i is the vector index of the i^{th} measurement point and $\boldsymbol{e}_i$ is a vector of all zeros with a one at component i. This algorithm may be interpreted as placing new measurement points at locations where the old measurement space is least capable of representing large correlations in the snapshot data. This may be observed from the fact that it places measurement points where the residual $\boldsymbol{r} = \boldsymbol{\xi}_i - M\boldsymbol{c}$ is maximal in absolute value. [29] remarks that since the DEIM modes are linearly independent the residuals will never be exactly zero. This motivates a lemma which is used to proof that $P^T M$ is always non-singular, a statement that is important for (4.34). The DEIM may be considered to be consistent in the sense that when $P = I$ and $m = N$ the exact operator output $C_h(\boldsymbol{u})$ is recovered. This is due to the fact that when $m = N$ the DEIM modes span the full state space $\mathbb{R}^N$. This makes M non-singular allowing the following to be stated for (4.32):

$$C_h(\boldsymbol{u}) = I^T C_h(\boldsymbol{u}) = I^T M\boldsymbol{c} = M\boldsymbol{c}, \quad \text{where} \quad m = N.$$

The condition $m = N$ in the notion of consistency can even be relaxed further to the condition $C_h(\boldsymbol{u}) \in \mathcal{M}_d$, a proof of this is provided in Appendix A.

Algorithm 2 DEIM interpolation indices

1: $p_1 = \arg\max(|\boldsymbol{\xi}_i|)$
2: $M = [\boldsymbol{\xi}_i], \quad P = [\boldsymbol{e}_{p_1}]$
3: **for** $i = 2$ to m **do**
4: Solve $P^T M\boldsymbol{c} = P^T \boldsymbol{\xi}_i$ for $\boldsymbol{c}$
5: $\boldsymbol{r} = \boldsymbol{\xi}_i - M\boldsymbol{c}$
6: $p_i = \arg\max(|\boldsymbol{r}|)$
7: $M \leftarrow [M, \boldsymbol{\xi}_i], \quad P \leftarrow [P, \boldsymbol{e}_{p_i}]$

Having constructed the DEIM approximation to the nonlinear convection operator, (4.13) is approximated as follows:

$$\frac{d\boldsymbol{a}}{dt} = -\Phi^T M\boldsymbol{c}(\boldsymbol{a}) + \nu D_r \boldsymbol{a}, \tag{4.37}$$

where $\boldsymbol{c}(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^m$ are the DEIM coordinates calculated for $C_h(\Phi\boldsymbol{a})$[1]. In turn the DEIM coordinates are determined from:

$$\boldsymbol{c}(\boldsymbol{a}) = (P^T M)^{-1} P^T C_h(\Phi\boldsymbol{a}). \tag{4.38}$$

In evaluating $P^T C_h(\Phi\boldsymbol{a})$ it will be necessary to reconstruct components of the reduced velocity vector $\boldsymbol{u}_r$ corresponding to the computational stencil of the discrete convection operator. Component i of the reduced velocity vector may be found in $\mathcal{O}(r)$ operations by the Euclidean inner product between row i of Φ and the generalized coordinates $\boldsymbol{a}$.

Momentum
It will now be shown that, using the altered POD basis of [86], the ODE in (4.37) conserves total momentum for periodic boundary conditions. To the author's knowledge this construction to conserve total momentum has not been considered before. Conservation of reduced total momentum can be shown by considering the eigenvector problem underlying the SVD used in the construction of M. The reduced total momentum evolution due to (4.37) is found by substituting (4.37) into the temporally differentiated definition of reduced total momentum:

$$\frac{dP_r^u}{dt} = \boldsymbol{e}_u^T \Omega_h \Phi \Phi^T \left[-M\boldsymbol{c}(\boldsymbol{a}) + \nu D_h \Phi \boldsymbol{a} \right], \tag{4.39}$$

here Φ^T was taken outside of the brackets. Since the POD basis Φ is constructed to exactly embed $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$ it can be stated that:

$$\frac{dP_r^u}{dt} = \boldsymbol{e}_u^T \left[-M\boldsymbol{c}(\boldsymbol{a}) + \nu D_h \Phi \boldsymbol{a} \right]. \tag{4.40}$$

Using the telescoping property of the FOM operators this results into:

$$\frac{dP_r^u}{dt} = -\boldsymbol{e}_u^T M \boldsymbol{c}(\boldsymbol{a}). \tag{4.41}$$

Now considering the eigenvector problem underlying the construction of M and taking the Euclidean inner product with $\boldsymbol{e}_u$ results into:

$$\boldsymbol{e}_u^T \Xi \Xi^T \boldsymbol{\xi}_j = \lambda_j^2 \boldsymbol{e}_u^T \boldsymbol{\xi}_j,$$

where λ_j is the singular value associated to the left singular vector $\boldsymbol{\xi}_j$ which is the j^{th} column of M. It can now be stated that, because every column of Ξ satisfies the telescoping property, $\boldsymbol{e}_u^T \Xi = 0$ holds. In turn, because $\lambda_j > 0$ holds for the singular values given j is less than the rank of Ξ it must mean that $\boldsymbol{e}_u^T \boldsymbol{\xi}_j = 0$. Therefore, the inner products $-\boldsymbol{e}_u^T M \boldsymbol{c}(\boldsymbol{a})$ in (4.41) is zero for periodic boundary conditions and total reduced momentum is conserved by the hyper-reduced order model (hROM) (4.37):

$$\boxed{\frac{dP_r^u}{dt} = 0}.$$

Equivalent results hold for $\frac{dP_r^v}{dt}$.

Kinetic energy
Using the DEIM, the hROM (4.37) *does not conserve reduced total kinetic energy.* Going through the derivation (4.25) again using the DEIM approximation (4.32) results into:

$$\begin{aligned} \frac{dK_r}{dt} = \frac{1}{2}\frac{d}{dt}\langle \boldsymbol{a}, \boldsymbol{a} \rangle &= \left\langle \boldsymbol{a}, \frac{d\boldsymbol{a}}{dt} \right\rangle \\ &= \left\langle \boldsymbol{a}, -\Phi^T M \boldsymbol{c}(\boldsymbol{a}) + \nu D_r \boldsymbol{a} \right\rangle \\ &= -\boldsymbol{a}^T \Phi^T M \boldsymbol{c}(\boldsymbol{a}) - \nu \left|\left| Q_r \boldsymbol{a} \right|\right|^2. \end{aligned} \tag{4.42}$$

[1] This is a slight change in notation from the notation introduced in (4.34)

Formulating the term $\boldsymbol{a}^T\Phi^T M\boldsymbol{c}(\boldsymbol{a})$ explicitly in $\boldsymbol{a}$, using the quasi-linear form of the discrete convection operator gives the following:

$$\boldsymbol{a}^T\Phi^T M\boldsymbol{c}(\boldsymbol{a}) = \boldsymbol{a}^T\Phi^T M(P^T M)^{-1}P^T\widetilde{C}_h(\Phi\boldsymbol{a})\Phi\boldsymbol{a}.$$

The matrix:

$$\Phi^T M(P^T M)^{-1}P^T\widetilde{C}_h(\Phi\boldsymbol{a})\Phi,$$

is not skew-symmetric and simple empirical testing shows that generally $\boldsymbol{a}^T\Phi^T M\boldsymbol{c}(\boldsymbol{a}) \neq 0$. Thus, in the case of (4.37) and (4.42) structure is not preserved by the hROM. This can lead to instability as the norm of the generalized coordinates $\boldsymbol{a}$ is not bounded from above. It is for this reason that the present thesis concerns itself with developing structure-preserving DEIM algorithms, as for structure-preserving DEIM algorithms the norm of $\boldsymbol{a}$ *is* bounded.

4.3. The LSDEIM

It has been shown in the previous section that the conventional DEIM conserves reduced total momentum, but it does not conserve reduced total kinetic energy (for inviscid flow). However, energy conservation is a property that is crucial for nonlinear stability. Therefore it is of interest to design a DEIM-like hyper-reduction technique that conserves both reduced total momentum and reduced total kinetic energy. Such an energy and momentum-conserving DEIM-like algorithm will be referred to as a structure-preserving DEIM algorithm. The first of three structure-preserving methods proposed in this thesis will be referred to as the least-squares discrete empirical interpolation method (LSDEIM) and will be discussed in this section.

A sufficient condition for reduced total kinetic energy to be conserved as stated in (4.42), is that the interpolated quasi-linear convection operator $M(P^T M)^{-1}P^T\widetilde{C}_h(\Phi\boldsymbol{a})$ be skew-symmetric, yet this is generally not the case. However, considering the last line of (4.42), skew-symmetry of the interpolated quasi-linear convection operator is not a necessary condition for energy-conservation. Instead, a necessary condition to conserve reduced total kinetic energy in the inviscid limit is:

$$\boldsymbol{a}^T\Phi^T M\boldsymbol{c}(\boldsymbol{a}) = 0. \tag{4.43}$$

Condition (4.43) can be satisfied even if the operator $\Phi^T M\boldsymbol{c}(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^r$ is not skew-adjoint and, when satisfied, results in the correct reduced kinetic energy evolution equation (4.26). The idea of the LSDEIM is to enforce this condition by posing the DEIM as a constrained minimization problem.

4.3.1. The Method: Constrained Minimization

The conventional DEIM finds the DEIM coordinates $\boldsymbol{c}$ by minimizing the Euclidean norm between the nonlinearity and the DEIM approximation in the measurement space. The Euclidean norm can be considered as minimized since the difference in $\mathcal{P}_h$ i.e. $P^T C_h(\Phi\boldsymbol{a}) - P^T M\boldsymbol{c}$, is zero. The new idea of the LSDEIM is to constrain this minimization problem to take place over the set $\mathcal{F}(\boldsymbol{a})$ of DEIM approximations satisfying condition (4.43) defined using the DEIM coordinates as:

$$\mathcal{F}(\boldsymbol{a}) := \{\boldsymbol{c} \in \mathbb{R}^m \mid \boldsymbol{a}^T\Phi^T M\boldsymbol{c} = 0\}.$$

The set $\mathcal{F}(\boldsymbol{a})$ is referred to as the feasible set. As DEIM approximations with $\boldsymbol{c}(\boldsymbol{a}) \in \mathcal{F}(\boldsymbol{a})$ satisfy condition (4.43), the LSDEIM produces approximations that conserve reduced total kinetic energy in the inviscid limit. The constrained minimization problem to find the DEIM coordinates $\boldsymbol{c}$ underlying the LSDEIM will be posed as the following linearly constrained least-squares problem:

$$\boldsymbol{c}(\boldsymbol{a}) = \underset{\widetilde{\boldsymbol{c}}\in\mathbb{R}^m}{\arg\min} \left|\left|P^T C_h(\Phi\boldsymbol{a}) - P^T M\widetilde{\boldsymbol{c}}\right|\right|^2 \quad \text{s.t.} \quad \boldsymbol{a}^T\Phi^T M\widetilde{\boldsymbol{c}} = 0. \tag{4.44}$$

This means the LSDEIM relaxes condition (4.33) of exact correspondence between the FOM's convection operator and the DEIM approximation in the measurement space imposed by the conventional DEIM. Rather, the LSDEIM minimizes the differences between the DEIM approximation and the FOM operator in the measurement space, simultaneously constraining the

approximation to be energy-conserving. Note that the LSDEIM has a sense of consistency in the same way as the conventional DEIM when the underlying FOM operator is skew-symmetric. This is the case as the constraint does not prevent equality between the DEIM approximation and the FOM operator when $m = N$ or more generally when $C_h(\Phi \boldsymbol{a}) \in \mathcal{M}_d$. A proof of this statement is provided in a later section (subsection 4.5.3).

Considering the objective function and the geometric interpretation of the conventional DEIM, the LSDEIM can be interpreted geometrically as an oblique projection of $C_h(\Phi \boldsymbol{a})$ onto the subspace $\mathcal{M}_{\mathcal{F}} \subset \mathbb{R}^N$ of all DEIM approximations $M\boldsymbol{c}$ with $\boldsymbol{c} \in \mathcal{F}(\boldsymbol{a})$. The subspace $\mathcal{M}_{\mathcal{F}}$ is defined as:

$$\mathcal{M}_{\mathcal{F}} = \mathcal{M}_d \cap \ker(\boldsymbol{a}^T \Phi^T),$$

as these are all vectors in $\mathbb{R}^N$ that can be written as a linear combination of the columns of M and satisfy condition (4.43). Since $\mathcal{M}_{\mathcal{F}}$ is the intersection between two linear subspaces of $\mathbb{R}^N$ it is also a linear subspace of $\mathbb{R}^N$. Contrary to the conventional DEIM, the LSDEIM also projects obliquely through the measurement space $\mathcal{P}_h$. Hence, the orthogonal projection of the DEIM residual $\boldsymbol{r}(t) \in \mathbb{R}^N$ onto $\mathcal{P}_h$, as given by:

$$P^T \boldsymbol{r}(t) = P^T \left[C_h(\Phi \boldsymbol{a}(t)) - M\boldsymbol{c}(t) \right],$$

is generally not zero. However, the LSDEIM projector $\Pi_{\mathcal{M}_{\mathcal{F}}} : \mathbb{R}^N \to \mathcal{M}_{\mathcal{F}}$ does project $C_h(\Phi \boldsymbol{a})$ onto $\mathcal{M}_{\mathcal{F}}$ such that the orthogonally projected residual $P^T \boldsymbol{r}(t)$ is minimal in the Euclidean norm. This statement follows exactly from the formulation of the LSDEIM minimization problem (4.44).

Furthermore, the DEIM basis M and the measurement space $\mathcal{P}_h$ can simply be found following the procedures of the conventional DEIM algorithm. Indeed, using the conventional DEIM basis the reduced total momentum will remain a conserved quantity for the LSDEIM.

The constrained minimization problem can be solved using the method of Lagrange multipliers. The Lagrangian $\mathcal{L}(\boldsymbol{c}, \lambda) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ of this minimization is defined as:

$$\mathcal{L}(\boldsymbol{c}, \lambda) = \left|\left| P^T C_h(\Phi \boldsymbol{a}) - P^T M \boldsymbol{c} \right|\right|^2 + \lambda \boldsymbol{a}^T \Phi^T M \boldsymbol{c},$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. Taking partial derivatives of the Lagrangian results in:

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{c}, \lambda)}{\partial \boldsymbol{c}} &= 2\boldsymbol{c}^T M^T P P^T M - 2C_h(\Phi \boldsymbol{a})^T P P^T M + \lambda \boldsymbol{a}^T \Phi^T M \\ \frac{\partial \mathcal{L}(\boldsymbol{c}, \lambda)}{\partial \lambda} &= \boldsymbol{a}^T \Phi^T M \boldsymbol{c}. \end{aligned}$$

As optimality condition it is required that:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{c}}(\boldsymbol{c}_o, \lambda_o) = 0, \quad \frac{\partial \mathcal{L}}{\partial \lambda}(\boldsymbol{c}_o, \lambda_o) = 0,$$

where $(\boldsymbol{c}_o, \lambda_o)$ constitutes an optimum. It will be verified that the optimum is a local minimum in subsection 4.3.3 by proving the Lagrangian's Hessian is positive definite. A set of equations for the optima can be found as:

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{c}_o, \lambda_o)}{\partial \boldsymbol{c}} &= 2\boldsymbol{c}_o^T M^T P P^T M - 2C_h(\Phi \boldsymbol{a})^T P P^T M + \lambda_o \boldsymbol{a}^T \Phi^T M = \boldsymbol{0}^T \\ 2\boldsymbol{c}_o^T M^T P P^T M + \lambda_o \boldsymbol{a}^T \Phi^T M &= 2C_h(\Phi \boldsymbol{a})^T P P^T M \\ 2(M^T P P^T M)^T \boldsymbol{c}_o + (\boldsymbol{a}^T \Phi^T M)^T \lambda_o &= 2(P^T M)^T P^T C_h(\Phi \boldsymbol{a}) \\ 2(P^T M)^T P^T M \boldsymbol{c}_o + (\boldsymbol{a}^T \Phi^T M)^T \lambda_o &= 2(P^T M)^T P^T C_h(\Phi \boldsymbol{a}) \\ \frac{\partial \mathcal{L}(\boldsymbol{c}_o, \lambda_o)}{\partial \lambda} &= \boldsymbol{a}^T \Phi^T M \boldsymbol{c}_o = 0, \end{aligned}$$

where $\boldsymbol{0} \in \mathbb{R}^m$ is a column vector of zeros. This system can be written in block matrix form as:

$$\begin{bmatrix} 2(P^T M)^T P^T M & (\boldsymbol{a}^T \Phi^T M)^T \\ \boldsymbol{a}^T \Phi^T M & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_o \\ \lambda_o \end{bmatrix} = \begin{bmatrix} 2(P^T M)^T P^T C_h(\Phi \boldsymbol{a}) \\ 0 \end{bmatrix}. \tag{4.45}$$

Note that this matrix is symmetric and has constant coefficients with the exception of the last row and column which depend on $\boldsymbol{a}$. Solving (4.45) can be done explicitly. Namely, [57] provides the inverse of a symmetric 2×2 block matrix as:

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - B^TA^{-1}B)^{-1}B^TA^{-1} & -A^{-1}B(D - B^TA^{-1}B)^{-1} \\ -(D - B^TA^{-1}B)^{-1}B^TA^{-1} & (D - B^TA^{-1}B)^{-1} \end{bmatrix},$$

where A and D are symmetric. Taking:

$$\begin{aligned} A &:= 2(P^TM)^TP^TM && \in \mathbb{R}^{m\times m} \\ B &:= (\boldsymbol{a}^T\Phi^TM)^T && \in \mathbb{R}^m \\ D &:= 0, && \in \mathbb{R} \end{aligned}$$

the DEIM coordinates $\boldsymbol{c}_o$ solving (4.44) are determined by:

$$\boldsymbol{c}(\boldsymbol{a}) = \left(A^{-1} + A^{-1}B(D - B^TA^{-1}B)^{-1}B^TA^{-1}\right) 2(P^TM)^TP^TC_h(\Phi\boldsymbol{a}).$$

It will be insightful to denote explicitly the dependence of B on $\boldsymbol{a}$ and take into account that it is a vector, hence the notation $B := \boldsymbol{b}(\boldsymbol{a})$ will be used from here on. Now using that $D = 0$ and that $B^TA^{-1}B = \boldsymbol{b}(\boldsymbol{a})^TA^{-1}\boldsymbol{b}(\boldsymbol{a})$ and $B^TA^{-1}2(P^TM)^TP^TC_h(\Phi\boldsymbol{a}) = \boldsymbol{b}(\boldsymbol{a})^T(P^TM)^\dagger P^TC_h(\Phi\boldsymbol{a})$ are simply scalars, a final expression for $\boldsymbol{c}_o$ can be obtained:

$$\boxed{\boldsymbol{c}(\boldsymbol{a}) = (P^TM)^\dagger P^TC_h(\Phi\boldsymbol{a}) - \frac{\boldsymbol{b}(\boldsymbol{a})^T(P^TM)^\dagger P^TC_h(\Phi\boldsymbol{a})}{(\boldsymbol{b}(\boldsymbol{a})^TA^{-1}\boldsymbol{b}(\boldsymbol{a}))}A^{-1}\boldsymbol{b}(\boldsymbol{a})}. \tag{4.46}$$

Here $(P^TM)^\dagger$ denotes the Moore-Penrose pseudoinverse of P^TM and is simply the normal inverse for square and non-singular P^TM. The value of the DEIM coordinates found from equation (4.46) is used for the LSDEIM and results in a DEIM approximation $M\boldsymbol{c}$ satisfying condition (4.43).

4.3.2. Practical Implementation of the Algorithm

Using the DEIM coordinates as determined by the LSDEIM in (4.46) should result in competitive performance in comparison with the conventional DEIM. However, there are many different orders in which a practical implementation to evaluate (4.46) can compute the different terms involved. Therefore, a practical implementation will be provided that scales computationally in the same way as the LU-solve step in the conventional DEIM algorithm. This keeps the computing work minimal and remains equivalent to the conventional DEIM in terms of speed. The offline stage of the practical implementation is provided in Algorithm 3 and the online stage is provided in Algorithm 4.

Algorithm 3 Evaluating DEIM coordinates using the LSDEIM (Offline phase)

1: $L, U = \text{LU-Decompose}(2(P^TM)^TP^TM)$
2: $M_1 = \text{LU-Solve}(L, U, 2(P^TM)^T)$

Algorithm 4 Evaluating DEIM coordinates using the LSDEIM (Online phase)

1: $\boldsymbol{v}_1 = P^TC_h(\Phi\boldsymbol{a})$
2: $\boldsymbol{v}_2 = (\boldsymbol{a}^T\Phi^TM)^T$
3: $\boldsymbol{v}_3 = \text{LU-Solve}(L, U, \boldsymbol{v}_2)$
4: $\boldsymbol{v}_4 = M_1\boldsymbol{v}_1$
5: $\boldsymbol{c} = \boldsymbol{v}_4 - \frac{\langle \boldsymbol{v}_2, \boldsymbol{v}_4\rangle}{\langle \boldsymbol{v}_2, \boldsymbol{v}_2\rangle}\boldsymbol{v}_3$

The most expensive steps in Algorithm 4 in terms of scaling are step 3 where an LU-solve is performed and step 4 where a matrix-vector product is performed. The computational complexity of both the LU-solve step and the matrix-vector product is $\mathcal{O}(m^2)$. Hence the algorithm's computational cost scales with $\mathcal{O}(m^2)$. The evaluation of (4.46) can thus be performed in a number of computations that scales equivalently to the determination of the DEIM coordinates using the conventional DEIM.

4.3.3. Existence and Uniqueness

Existence of a solution to the LSDEIM minimization (4.44) can be proved by analysing the expression in (4.46). There are two conditions that must hold for a solution to (4.46) to be defined namely:

1. A is non-singular,
2. $\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a}) \neq 0, \quad \forall\, \boldsymbol{a} \in \{\boldsymbol{a} \in \mathbb{R}^r \mid \boldsymbol{a} \notin \ker\left(M^T\Phi\right)\}$.

It can be shown easily that A is non-singular as it is the product of two non-singular matrices P^TM and $(P^TM)^T$, where invertibility of P^TM, and hence $(P^TM)^T$, is shown in [29]. Using that:

$$\boldsymbol{b}(\boldsymbol{a}) = 0 \iff \boldsymbol{a} \in \ker\left(M^T\Phi\right),$$

condition 2 holds at least for all $\boldsymbol{a} \in \{\boldsymbol{a} \in \mathbb{R}^r \mid \boldsymbol{a} \notin \ker\left(M^T\Phi\right)\}$ if A^{-1} is positive-definite, i.e.:

$$\boldsymbol{x}^T A^{-1}\boldsymbol{x} > 0 \quad \forall\, \boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{x} \neq 0\}. \tag{4.47}$$

Condition (4.47), holds if A is positive-definite, since:

$$\boldsymbol{x}^T A^{-1}\boldsymbol{x} = \boldsymbol{y}^T A^T A^{-1} A\boldsymbol{y} = \boldsymbol{y}^T A\boldsymbol{y},$$

and $\mathrm{col}(A) = \mathbb{R}^m$. Positive-definiteness of A may then be shown by:

$$\begin{aligned}\boldsymbol{y}^T A\boldsymbol{y} &= 2\boldsymbol{y}^T(P^TM)^T P^TM\boldsymbol{y}\\ &= 2(P^TM\boldsymbol{y})^T(P^TM\boldsymbol{y})\\ &= 2||P^TM\boldsymbol{y}||^2,\end{aligned}$$

where, as P^TM has linearly independent columns, it may thus be stated that:

$$\boldsymbol{y}^T A\boldsymbol{y} = 2||P^TM\boldsymbol{y}||^2 > 0 \quad \forall\, \boldsymbol{y} \in \{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{y} \neq 0\},$$

proving condition 2 holds for all $\boldsymbol{a} \in \{\boldsymbol{a} \in \mathbb{R}^r \mid \boldsymbol{a} \notin \ker\left(M^T\Phi\right)\}$. Clearly, using the definition of $\boldsymbol{b}(\boldsymbol{a})$, condition 2 does not hold if $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$. Thus (4.46) is not defined when $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$. However, this is no problem. Namely, if $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$, it holds for the feasible set $\mathcal{F}(\boldsymbol{a})$ that:

$$\mathcal{F}(\boldsymbol{a}) = \mathbb{R}^m,$$

since condition (4.43) holds for any $\boldsymbol{c} \in \mathbb{R}^m$. Therefore, minimization (4.44) is essentially unconstrained and the unconstrained least-squares version of (4.46) can be solved while still conserving energy, i.e. the conventional DEIM for which there exists a unique solution by virtue of P^TM being invertible.

Uniqueness of the optimum of the constrained minimization (4.44) may be shown through the well-known theory of convex optimization [20]. In what follows the special, well-posed case of $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$ will be ignored. It may be stated that for a constrained minimization problem:

$$\min f(\boldsymbol{x}) \quad \text{s.t.} \quad C(\boldsymbol{x}) = 0$$

there exists no more than one optimum if the objective function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is strictly convex:

$$f(\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y}) < \theta f(\boldsymbol{x}) + (1-\theta)f(\boldsymbol{y}), \quad \forall\, (\boldsymbol{x},\boldsymbol{y}) \in \{\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n \mid \boldsymbol{x} \neq \boldsymbol{y}\}, \quad \forall\, \theta \in (0,1),$$

and the feasible set $\mathcal{F}_C$ associated to $C(\boldsymbol{x}) = 0$ is convex:

$$\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y} \in \mathcal{F}_C, \quad \forall\, \boldsymbol{x},\boldsymbol{y} \in \mathcal{F}_C, \quad \forall\, \theta \in [0,1],$$

and closed. Using that the constraint in (4.44) is affine, convexity of the feasible set $\mathcal{F}(\boldsymbol{a})$ is easily derived:

$$\begin{aligned}C(\theta\boldsymbol{c}_1 + (1-\theta)\boldsymbol{c}_2) &= \boldsymbol{a}^T\Phi^T M(\theta\boldsymbol{c}_1 + (1-\theta)\boldsymbol{c}_2)\\ &= \theta\boldsymbol{a}^T\Phi^T M\boldsymbol{c}_1 + (1-\theta)\boldsymbol{a}^T\Phi^T M\boldsymbol{c}_2\\ &= 0,\end{aligned}$$

for any arbitrary $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{F}(\boldsymbol{a})$ and $\theta \in [0,1]$, hence $\theta\boldsymbol{c}_1 + (1-\theta)\boldsymbol{c}_2 \in \mathcal{F}(\boldsymbol{a})$, thus proving convexity of $\mathcal{F}(\boldsymbol{a})$. Furthermore, since $\mathcal{F}(\boldsymbol{a})$ is a linear subspace of $\mathbb{R}^m$ for all $\boldsymbol{a} \notin \ker\left(M^T\Phi\right)$ it is a closed set. A sufficient condition for strict convexity of the objective function is if its Hessian is positive-definite [20]. It is straightforward to show that this is the case. The Hessian matrix of the objective function is:

$$\begin{aligned}
&\frac{\partial^2}{\partial \boldsymbol{c}\partial \boldsymbol{c}^T}\left(\left|\left|P^T C_h(\Phi\boldsymbol{a}) - P^T M\boldsymbol{c}\right|\right|\right) \\
&= \frac{\partial^2}{\partial \boldsymbol{c}\partial \boldsymbol{c}^T}\left(\boldsymbol{c}^T M^T PP^T M\boldsymbol{c} - 2C_h(\Phi\boldsymbol{a})^T PP^T M\boldsymbol{c} + C_h(\Phi\boldsymbol{a})^T PP^T C_h(\Phi\boldsymbol{a})\right) \\
&= \frac{\partial}{\partial \boldsymbol{c}^T}\left(2\boldsymbol{c}^T M^T PP^T M - 2C_h(\Phi\boldsymbol{a})^T PP^T M + \lambda \boldsymbol{a}^T\Phi^T M\right) \\
&= 2M^T PP^T M.
\end{aligned}$$

Indeed, the matrix $2M^T PP^T M = 2(P^T M)^T P^T M$ has already been shown to be positive definite. The objective function is therefore strictly convex. Since a solution to the LSDEIM minimization (4.44) *exists* following the previous paragraph and can be found using the method of Lagrange multipliers, it can be concluded from the strict convexity of the minimization over a closed, convex set that this solution is *unique* and therefore the only solution.

4.3.4. The LSDEIM Jacobian

As the hROM (4.37) using the LSDEIM satisfies condition (4.43), it conserves reduced total kinetic energy. For this reason it can be desirable to integrate the hROM in time using an energy-conserving implicit Runge-Kutta method. However, as will be seen in section 4.6, this requires determination of the Jacobian $J_m(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^{r\times r}$ of the Galerkin projected LSDEIM approximation of the convection operator. This Jacobian can be calculated as follows:

$$J_m(\boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{a}}\left(\Phi^T M\boldsymbol{c}(\boldsymbol{a})\right) = \Phi^T M\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}.$$

Where the partial derivative $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}$ is given by:

$$\begin{aligned}
\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} =& 2A^{-1}(P^T M)^T P^T J_h(\Phi\boldsymbol{a})\Phi - 2\gamma(\boldsymbol{a})A^{-1}M^T\Phi \\
&- 2A^{-1}\boldsymbol{b}(\boldsymbol{a}) \otimes \left[\frac{1}{\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})}\left(\Phi^T MA^{-1}\boldsymbol{c}(\boldsymbol{a}) + \left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1}(P^T M)^T P^T J_h(\Phi\boldsymbol{a})\Phi\right]^T\right)\right] \\
&- 2A^{-1}\boldsymbol{b}(\boldsymbol{a}) \otimes \left[\frac{\gamma(\boldsymbol{a})}{\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})}\left(\Phi^T MA^{-1}M^T\Phi\boldsymbol{a} + \left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1}M^T\Phi\right]^T\right)\right].
\end{aligned}$$

A derivation of this expression is provided in section B.4.

To calculate the LSDEIM Jacobian in practise the following computing algorithms are proposed. In Algorithm 5 the offline phase of the algorithm is provided and in Algorithm 6 the online phase is provided. The most expensive step in Algorithm 6 is the matrix-matrix multiplication of $M_1 \in R^{m\times m}$ and $M_5 \in \mathbb{R}^{m\times r}$ which is of order $\mathcal{O}(m^2 r)$. The calculation of the LSDEIM Jacobian requires an additional matrix-matrix multiplication of $\Phi^T M \in \mathbb{R}^{r\times m}$ and $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} \in \mathbb{R}^{m\times r}$ of order $\mathcal{O}(r^2 m)$. Hence, the full procedure of calculating the LSDEIM Jacobian has computational scaling of $\mathcal{O}(\max(m^2 r, r^2 m))$, equivalently to the conventional DEIM.

Algorithm 5 Evaluating LSDEIM Jacobian (Offline phase)

1: $L, U =$ LU-Decompose$(2(P^T M)^T P^T M)$
2: $M_1 :=$ LU-Solve$(L, U, 2(P^T M)^T)$
3: $M_2 := \Phi^T M \cdot M_1$
4: $M_3 := \Phi^T M \cdot M_4$
5: $M_4 :=$ LU-Solve$(L, U, M^T\Phi)$

Algorithm 6 Evaluating LSDEIM Jacobian (Online phase)

1: $M_5 := P^T J_h(\Phi \boldsymbol{a})\Phi$
2: $\boldsymbol{v}_1 := P^T C_h(\Phi \boldsymbol{a})$
3: $\alpha := \boldsymbol{a}^T M_3 \boldsymbol{a}$
4: $\beta := \boldsymbol{a}^T M_2 \boldsymbol{v}_1$
5: $\gamma := \beta/\alpha$
6: $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} := M_1 M_5 - \gamma M_4 - M_4 \boldsymbol{a} \otimes \left[\frac{1}{\alpha}\left(M_2 \boldsymbol{v}_1 + M_5^T M_2^T \boldsymbol{a}\right) - \frac{\gamma}{\beta}\left(M_3 \boldsymbol{a} + M_3^T \boldsymbol{a}\right)\right]$

4.4. The SMDEIM

The LSDEIM introduced in the previous section produces results that do not exactly match the nonlinearity $C_h(\Phi \boldsymbol{a})$ in $\mathcal{P}_h$. In an unfortunate case these differences could become substantial. It may therefore be desirable to have exact correspondence between the DEIM approximation $M\boldsymbol{c}$ and $C_h(\Phi \boldsymbol{a})$ in $\mathcal{P}_h$. A method that can do this will be proposed in what follows and will be referred to as the Sherman-Morrison discrete empirical interpolation method (SMDEIM).

4.4.1. The Method: Rank One Correction

Fundamentally, the problem of satisfying condition (4.43) and exact correspondence between the nonlinearity and the DEIM approximation in the measurement space is over-determined. Given m DEIM modes and DEIM coordinates and thus m degrees of freedom there are $m+1$ equations to be satisfied. Namely, m equations are required for the interpolation and another equation is necessary to satisfy (4.43). To find a system with an equal number of equations as unknowns an additional degree of freedom is therefore required. The SMDEIM provides this degree of freedom by taking into consideration an additional $(m+1)^{\text{th}}$ DEIM mode and coordinate but leaving the measurement space $\mathcal{P}_h$ m-dimensional. This allows the following system to be written:

$$\begin{bmatrix} P^T M \\ \boldsymbol{a}^T \Phi^T M \end{bmatrix} \boldsymbol{c} = \begin{bmatrix} P^T C_h(\Phi \boldsymbol{a}) \\ 0 \end{bmatrix}. \tag{4.48}$$

A solution to this system simultaneously results in the nonlinearity being matched by the $m+1$ DEIM modes in the m DEIM interpolation points and additionally satisfies the extra condition for energy conservation (4.43). In some sense this approach is similar to the manner in which momentum conservation is enforced in the altered POD basis. Namely, both methods sacrifice accuracy for structure-preservation. Whereas the SMDEIM sacrifices the accuracy offered by an $(m+1)^{\text{th}}$ measurement point to impose an energy-conservation condition, the altered POD basis sacrifices d descriptive POD modes for d less descriptive POD modes that provide momentum conservation.

The linear system in (4.48) will now be solved. Although, the following procedure can be performed for any of the $m+1$ components of $\boldsymbol{c}$ it will be chosen to perform it on the $(m+1)^{\text{th}}$ component as a principled choice is lacking and the $(m+1)^{\text{th}}$ DEIM mode carries the least energy. Moreover, if the linear system in (4.48) is non-singular all choices should lead to the same solution. The $m+1$ DEIM coordinates $\boldsymbol{c}$ can be solved for as follows:

$$\boldsymbol{a}^T \Phi^T M \boldsymbol{c} = \sum_{k=1}^{m+1} (\boldsymbol{a}^T \Phi^T M)_k c_k = 0 \rightarrow c_{m+1} = -\sum_{k=1}^{m} \frac{(\boldsymbol{a}^T \Phi^T M)_k}{(\boldsymbol{a}^T \Phi^T M)_{m+1}} c_k. \tag{4.49}$$

Note:

$$P^T M \boldsymbol{c} = \sum_{k=1}^{m+1} (P^T M)_{,k} c_k = P^T C_h(\Phi \boldsymbol{a}). \tag{4.50}$$

Substituting the previous relation for c_{m+1} (4.49) in this equation (4.50) gives:

$$P^T M \boldsymbol{c} = \sum_{k=1}^{m} \left[(P^T M)_{,k} - \frac{(\boldsymbol{a}^T \Phi^T M)_k}{(\boldsymbol{a}^T \Phi^T M)_{m+1}} (P^T M)_{,m+1}\right] c_k = P^T C_h(\Phi \boldsymbol{a}).$$

For simplicity the following notation is defined, the m-dimensional vector $\boldsymbol{r}(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^m$ satisfying:

$$(\boldsymbol{r}(\boldsymbol{a}))_i = -\frac{(\boldsymbol{a}^T \Phi^T M)_i}{(\boldsymbol{a}^T \Phi^T M)_{m+1}},$$

and the non-singular ([29]) matrix $M_p \in \mathbf{R}^{m\times m}$ satisfying:

$$(M_p)_{,i} = (P^T M)_{,i}, \quad i \in \{1, 2, ..., m\}.$$

The vector of the first m components of $\boldsymbol{c}$ is written as $\boldsymbol{c}_m \in \mathbb{R}^m$. The previous definitions can be introduced to find:

$$P^T M \boldsymbol{c} = \left[M_p + (P^T M)_{,m+1} \otimes \boldsymbol{r}(\boldsymbol{a})\right] \boldsymbol{c}_m = P^T C_h(\Phi a).$$

This is a rank one correction of the non-singular matrix M_p which would solve the conventional DEIM using m DEIM modes and measurement points. The inverse of the corrected matrix can be found using the Sherman-Morrison formula [93]:

$$\boxed{\boldsymbol{c}_m(\boldsymbol{a}) = \left[M_p^{-1} - \frac{M_p^{-1}(P^T M)_{,m+1}\boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}}{1 + \boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}(P^T M)_{,m+1}}\right] P^T C_h(\Phi \boldsymbol{a})}. \tag{4.51}$$

The last DEIM scaling factor is found using (4.49) by:

$$\boxed{c_{m+1}(\boldsymbol{a}) = \boldsymbol{r}(\boldsymbol{a})^T \boldsymbol{c}_m(\boldsymbol{a})}. \tag{4.52}$$

The hROM (4.37) is now given by the dynamical system:

$$\frac{d\boldsymbol{a}}{dt} = -\Phi^T M \begin{bmatrix} \boldsymbol{c}_m \\ c_{m+1} \end{bmatrix} + \nu D_r \boldsymbol{a},$$

and satisfies (4.43).

4.4.2. Practical Implementation of the Algorithm

Like the LSDEIM, the SMDEIM should result in competitive performance in comparison with the conventional DEIM in terms of speed. As with the LSDEIM there are many different orders in which the terms in (4.51) can be evaluated. Hence, an algorithm to evaluate (4.51) and (4.52) in practise is provided in Algorithm 7 and Algorithm 8. Evaluating the terms in (4.51) in this order results in a computational effort scaling with the LU-Solve step in the conventional DEIM. The most expensive step in Algorithm 8 is step 3, which scales as $\mathcal{O}(m^2)$.

Algorithm 7 Evaluating DEIM coordinates using the SMDEIM (Offline phase)

1: $L, U =$ LU-Decompose(M_p)
2: $\boldsymbol{v}_1 :=$ LU-Solve$(L, U, ((P^T M)_{,m+1}))$

Algorithm 8 Evaluating DEIM coordinates using the SMDEIM (Online phase)

1: $\boldsymbol{v}_2 = P^T C_h(\Phi \boldsymbol{a})$
2: $\boldsymbol{v}_3 = \boldsymbol{r}(\boldsymbol{a})$
3: $\boldsymbol{v}_4 =$ LU-Solve$(L, U, \boldsymbol{v}_2)$
4: $\boldsymbol{c}_m = \boldsymbol{v}_4 - \frac{\langle \boldsymbol{v}_3, \boldsymbol{v}_4 \rangle}{\langle \boldsymbol{v}_3, \boldsymbol{v}_1 \rangle} \boldsymbol{v}_1$
5: $c_{m+1} = \langle \boldsymbol{v}_3, \boldsymbol{c}_m \rangle$

4.4.3. The SMDEIM Jacobian

As the DEIM approximations using the SMDEIM satisfy condition (4.43), implicit energy-conserving Runge-Kutta methods can be used to integrate (4.37) in time. Indeed, it is necessary to solve nonlinear systems of equations for such methods. These can be solved iteratively using the Newton-Raphson method. However, this does require the Jacobian of the DEIM coordinates to be known. For this reason a derivation of the SMDEIM Jacobian is provided here. The SMDEIM Jacobian $J_m(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^{r\times r}$ is found by taking partial derivatives to $\boldsymbol{a}$ of the Galerkin projected DEIM approximation as:

$$J_m(\boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{a}}\left(\Phi^T M \begin{bmatrix} \boldsymbol{c}_m(\boldsymbol{a}) \\ c_{m+1}(\boldsymbol{a}) \end{bmatrix}\right) = \Phi^T M \frac{\partial}{\partial \boldsymbol{a}}\left(\begin{bmatrix} \boldsymbol{c}_m(\boldsymbol{a}) \\ c_{m+1}(\boldsymbol{a}) \end{bmatrix}\right) = \Phi^T M \begin{bmatrix} \frac{\partial \boldsymbol{c}_m}{\partial \boldsymbol{a}} \\ (\nabla_a c_{m+1})^T \end{bmatrix},$$

here ∇_a denotes the gradient with respect to $\boldsymbol{a}$. The Jacobian is worked out further in section B.5.

The expression for the SMDEIM Jacobian is quite long and may be evaluated in many different orders. It is therefore of importance to choose the correct order to minimize the computational effort required to set up the Jacobian. An evaluation algorithm to keep the number of operations minimal is proposed here. In Algorithm 9 the offline phase has been provided and in Algorithm 10 the online phase has been provided.

Algorithm 9 Evaluating SMDEIM Jacobian (Offline phase)

1: $L, U = \text{LU-Decompose}(M_p)$
2: $\boldsymbol{v}_0 = \Phi^T M_{,m+1}$
3: $\boldsymbol{v}_1 = \text{LU-Solve}(L, U, ((P^T M)_{,m+1}))$
4: $M_1 = \Phi^T M I_m$

Algorithm 10 Evaluating SMDEIM Jacobian (Online phase)

1: $\boldsymbol{v}_2 = P^T C_h(\Phi \boldsymbol{a})$
2: $\boldsymbol{v}_3 = \boldsymbol{r}(\boldsymbol{a})$
3: $\boldsymbol{v}_4 = (\boldsymbol{a}^T \Phi^T M)^T$
4: $\boldsymbol{v}_5 = \text{LU-Solve}(L, U, \boldsymbol{v}_2)$
5: $\alpha = 1.0 + \langle \boldsymbol{v}_3, \boldsymbol{v}_1 \rangle$
6: $M_2 = \frac{1}{\text{last-component}(\boldsymbol{v}_4)}\left[-M_1^T + \boldsymbol{v}_4 \boldsymbol{v}_0^T\right]$
7: $M_3 = P^T J_h(\Phi \boldsymbol{a})\Phi$
8: $M_4 = \text{LU-Solve}(L, U, M_3)$
9: $\boldsymbol{v}_6 = \frac{1}{\alpha}\left[M_2^T \boldsymbol{v}_5 + M_4^T \boldsymbol{v}_3\right] - \frac{\langle \boldsymbol{v}_3, \boldsymbol{v}_5 \rangle}{\alpha^2} M_2^T \boldsymbol{v}_0$
10: $\frac{\partial \boldsymbol{c}_m}{\partial \boldsymbol{a}} = M_3 - \boldsymbol{v}_1 \boldsymbol{v}_6^T$
11: $\nabla_a c_{m+1} = M_2^T\left(\boldsymbol{v}_5 - \frac{\langle \boldsymbol{v}_3, \boldsymbol{v}_2 \rangle}{\alpha} \boldsymbol{v}_0\right) + \frac{\partial \boldsymbol{c}_m}{\partial \boldsymbol{a}}^T \boldsymbol{v}_3$

The most expensive step in the online phase denoted in Algorithm 10 is step 8. Here, an LU-Solve step is performed for every column of a $\mathbb{R}^{m\times r}$ matrix. Hence, the algorithm to compute the partial derivatives of the DEIM coordinates calculated using the SMDEIM has order of complexity $\mathcal{O}(m^2 r)$. The calculation of the SMDEIM Jacobian requires an additional matrix-matrix multiplication of $\Phi^T M \in \mathbb{R}^{r\times m}$ and $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} \in \mathbb{R}^{m\times r}$ of order $\mathcal{O}(r^2 m)$. Hence, the full procedure of calculating the SMDEIM Jacobian has computational scaling of $\mathcal{O}(\max(m^2 r, r^2 m))$, equivalently to the conventional DEIM and LSDEIM.

4.5. The Decoupled LSDEIM

When the flows that are modelled become more convection-dominated (ν becomes smaller) the Kolmogorov N-width decay of both the POD and the DEIM approximations becomes slower. Furthermore, for the incompressible Navier-Stokes equation (2.13) increasingly finer spatial features start appearing, making velocity fields highly fluctuating in space. It is known that for such conditions the DEIM can have stability issues [72]. Moreover, it can be shown [72,

9] that the approximation error increases as a function the DEIM space dimension according to $\mathcal{O}(\sqrt{m})$. In the case of the proposed structure-preserving DEIM methods this cannot lead to unbounded solution norms due to the nonlinear stability, but it can lead to a deterioration of approximation accuracy and non-convergence of any iterative algorithms to solve implicit timesteps (this will be observed in chapter 5). A solution to this problem applied in this thesis is oversampling. The concept of oversampling is not new and has been analysed in [72, 82], other references regarding oversampling are [107, 59]. The idea of oversampling is to increase $\dim(\mathcal{P}_h)$ while keeping the number of DEIM modes m constant, essentially decoupling the dimension of the measurement space and the DEIM space $\mathcal{M}_d$. Thus more measurement points are added to the DEIM solution algorithm.

Although it is difficult to employ this concept in the SMDEIM setting, the LSDEIM can easily be generalized to encapsulate this. The generalization of the LSDEIM to decoupled measurement and DEIM space dimensions will be referred to as the decoupled least-squares discrete empirical interpolation method (DLSDEIM). Denoting the dimension of the measurement space $m_p := \dim(\mathcal{P}_h)$ with $m_p \geq m$, the measurement matrix now becomes a matrix $P \in \mathbb{R}^{N\times m_p}$. The constrained least squares problem (4.44) underlying the LSDEIM is still equally valid for the DLSDEIM when $m \leq m_p$ and has the same solution (4.46). Note that no more measurement points can be determined following Algorithm 2 than the number of operator snapshots n_s as this is the number of DEIM modes available. However, situations might be encountered where more measurement points than n_s are required. In such situations it is recommended to apply any of the results in [59, 107] to find further measurement points in addition to those provided by Algorithm 2. Yet for simplicity, this thesis will use randomly selected extra measurement points in such situations.

4.5.1. Practical Implementation of the Algorithm

Since the measurement and DEIM space dimensions are now decoupled the algorithm to construct the measurement space changes slightly. The loop in Algorithm 2 is now carried out until $i = m_p$. Additionally, not all DEIM basis modes used to determine the measurement space are retained in the DEIM basis, only the first m modes are kept. The resulting algorithm is given in Algorithm 11.

The algorithms to determine the DEIM coordinates using the DLSDEIM approach are the same as Algorithm 3-4, where only the dimensions of some objects change due to the dimensions of the measurement matrix P changing. This does influence the scaling of the computational effort required to evaluate the DEIM approximation using the DLSDEIM. Namely, the matrix M_1 in Algorithm 3 is now $M_1 \in \mathbb{R}^{m\times m_p}$, therefore the cost of step 4 in Algorithm 4 will now be of order $\mathcal{O}(m_p m)$. When $m_p > m$ the computational scaling of step 4 will thus dominate the computational scaling of Algorithm 4. Equivalently, the algorithms to compute the DLSDEIM Jacobian are the same as Algorithm 5-6. Furthermore, the scaling of the computational effort of the most expensive step in Algorithm 6 now changes due to the dimensions of P changing. As $M_1 \in \mathbb{R}^{m\times m_p}$ and $M_5 \in \mathbb{R}^{m_p\times r}$, step 6 in Algorithm 6 will now scale computationally as $\mathcal{O}(m_p m r)$.

Algorithm 11 DLSDEIM interpolation indices

1: $p_1 = \arg\max(|\boldsymbol{\xi}_i|)$
2: $\widetilde{M} = [\boldsymbol{\xi}_i], \quad M = [\boldsymbol{\xi}_i], \quad P = [\boldsymbol{e}_{p_1}]$
3: **for** $i = 2$ to m_p **do**
4: Solve $P^T\widetilde{M}\boldsymbol{c} = P^T\boldsymbol{\xi}_i$ for $\boldsymbol{c}$
5: $\boldsymbol{r} = \boldsymbol{\xi}_i - \widetilde{M}\boldsymbol{c}$
6: $p_i = \arg\max(|\boldsymbol{r}|)$
7: $\widetilde{M} \leftarrow [\widetilde{M}, \boldsymbol{\xi}_i], \quad P \leftarrow [P, \boldsymbol{e}_{p_i}]$
8: **if** $i <= m$ **then**
9: $M \leftarrow [M, \boldsymbol{\xi}_i]$
10:

4.5.2. Existence and Uniqueness

Proving existence and uniqueness of the minimizer to the DLSDEIM is similar to proving existence and uniqueness of the LSDEIM. Conditions 1 and 2 in subsection 4.3.3 still must be satisfied for the DEIM coordinates using the DLSDEIM to be defined. However, proving that A is non-singular can now not be done directly on the basis of P^TM being non-singular as P^TM is no longer a square matrix. However, the symmetric matrix A is non-singular if it is positive-definite. A proof that A is positive-definite is as follows. Consider the matrix $M_1 := P^T[M, \widetilde{M}] \in \mathbb{R}^{m_p \times m_p}$ where the columns of $\widetilde{M} \in \mathbb{R}^{N \times (m_p - m)}$ consist of the $m_p - m$ DEIM modes succeeding the m^{th} DEIM mode. Following [29] the matrix M_1 is non-singular, hence it has linearly independent columns. Since the set of columns of P^TM is a subset of the set of columns of M_1, P^TM also has linearly independent columns. Now considering the condition for A to be positive-definite results into:

$$\begin{aligned}\boldsymbol{x}^T A \boldsymbol{x} &= 2\boldsymbol{x}^T (P^T M)^T P^T M \boldsymbol{x} \\ &= 2(P^T M \boldsymbol{x})^T (P^T M \boldsymbol{x}) \\ &= 2||P^T M \boldsymbol{x}||^2 > 0, \quad \forall\ \boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{x} \neq 0\},\end{aligned}$$

since the columns of P^TM are linearly independent. Hence, matrix A is positive-definite and thus non-singular, meaning condition 1 in subsection 4.3.3 is satisfied. Furthermore, by virtue of the same reasoning as in subsection 4.3.3, A^{-1} is positive-definite and thus in turn satisfies condition 2. As a result a solution to the constrained least-squares problem (4.44) where the dimensions of the measurement and DEIM space are decoupled can always be found using the method of Lagrange multipliers assuming $m_p \geq m$.

As the constraint did not change, the feasible set $\mathcal{F}(\boldsymbol{a})$ is still closed and convex for all $\boldsymbol{a} \notin \ker\left(M^T\Phi\right)$ and coincides with $\mathbb{R}^m$ in case $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$. The Hessian of the objective function in (4.44) is still $2(P^TM)^TP^TM$, and using that this matrix is positive-definite, it can be stated that the objective function is strictly convex. Thus, the constrained least-squares problem (4.44) has a unique minimizer and it can be found using the method of Lagrange multipliers. The unconstrained least-squares problem in cases $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$ and thus $\mathcal{F}(\boldsymbol{a}) = \mathbb{R}^m$ is also strictly convex and therefore also has no more than one minimizer. Furthermore, the solution to the unconstrained problem is well-known [20] and is given by the Moore-Penrose pseudoinverse of P^TM multiplied by $P^TC_h(\Phi\boldsymbol{a})$:

$$\boldsymbol{c} = (P^TM)^\dagger P^T C_h(\Phi\boldsymbol{a}) = \left((P^TM)^T(P^TM)\right)^{-1}(P^TM)^T P^T C_h(\Phi\boldsymbol{a})$$

which, due to $(P^TM)^TP^TM$ being invertible, is defined. Hence, a unique solution to the DLSDEIM exists for both cases $\boldsymbol{a} \notin \ker\left(M^T\Phi\right)$ and $\boldsymbol{a} \in \ker\left(M^T\Phi\right)$.

4.5.3. Consistency

The DLSDEIM and the LSDEIM both possess a sense of consistency when *the underlying FOM discretization is energy-conserving* and when $m = N$ or, more generally, when:

$$C_h(\Phi\boldsymbol{a}) \in \mathcal{M}_d. \tag{4.53}$$

Here consistency is in the sense of subsection 4.2.2. A proof of this statement is given in the following. If condition (4.53) is satisfied, it may be written that:

$$C_h(\Phi\boldsymbol{a}) = M\boldsymbol{c}_a, \tag{4.54}$$

since M is a basis of $\mathcal{M}_d$. Now considering equation (4.46) it may be written that:

$$\begin{aligned}\boldsymbol{c}(\boldsymbol{a}) &= (P^TM)^\dagger P^T C_h(\Phi\boldsymbol{a}) - \frac{\boldsymbol{b}(\boldsymbol{a})^T(P^TM)^\dagger P^T C_h(\Phi\boldsymbol{a})}{(\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a}))} A^{-1}\boldsymbol{b}(\boldsymbol{a}) \\ &= (P^TM)^\dagger P^T M\boldsymbol{c}_a - \frac{\boldsymbol{b}(\boldsymbol{a})^T(P^TM)^\dagger P^T M\boldsymbol{c}_a}{(\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a}))} A^{-1}\boldsymbol{b}(\boldsymbol{a}),\end{aligned}$$

when condition (4.53) is satisfied. Using the definition of the Moore-Penrose pseudoinverse it holds that:

$$(P^TM)^\dagger P^TM\boldsymbol{c}_a = ((P^TM)^T(P^TM))^{-1}(P^TM)^TP^TM\boldsymbol{c}_a = \boldsymbol{c}_a.$$

Using the previous result, the following may be written for the DEIM coordinates found using the LSDEIM or DLSDEIM:

$$\begin{aligned}\boldsymbol{c}(\boldsymbol{a}) &= (P^TM)^\dagger P^TM\boldsymbol{c}_a - \frac{\boldsymbol{b}(\boldsymbol{a})^T(P^TM)^\dagger P^TM\boldsymbol{c}_a}{(\boldsymbol{b}(\boldsymbol{a})^TA^{-1}\boldsymbol{b}(\boldsymbol{a}))}A^{-1}\boldsymbol{b}(\boldsymbol{a}) \\ &= \boldsymbol{c}_a - \frac{\boldsymbol{a}^T\Phi^TM\boldsymbol{c}_a}{(\boldsymbol{b}(\boldsymbol{a})^TA^{-1}\boldsymbol{b}(\boldsymbol{a}))}A^{-1}\boldsymbol{b}(\boldsymbol{a}).\end{aligned}$$

Finally, using (4.54) and that the FOM operator is energy-conserving it may be stated:

$$\boldsymbol{a}^T\Phi^TM\boldsymbol{c}_a = \boldsymbol{a}^T\Phi^TC_h(\Phi\boldsymbol{a}) = 0,$$

and thus that:

$$\boxed{\boldsymbol{c}(\boldsymbol{a}) = \boldsymbol{c}_a \quad \text{when} \quad C_h(\Phi\boldsymbol{a}) \in \mathcal{M}_d \text{ and } \boldsymbol{a}^T\Phi^TC_h(\Phi\boldsymbol{a}) = 0},$$

proving consistency of the LSDEIM and DLSDEIM in the sense of the conventional DEIM (see subsection 4.2.2).

4.6. Temporal Discretization of Structure-Preserving hROMs

Having found three different structure-preserving DEIM algorithms, the dynamical system (4.37) can now be integrated in time. In what follows, the time-integration of (4.37) will be discussed and the conservative properties of the structure-preserving DEIM algorithms will be analysed.

4.6.1. Time Integration

Like the FOM, the family of time-integration schemes to integrate the hROM of interest to this thesis are Runge-Kutta methods. Both implicit and explicit Runge-Kutta methods will be considered. Firstly, the implementation of explicit Runge-Kutta methods with Butcher tableaus characterized by Table 3.2 will be discussed. By virtue of the duality between G_h and M_h and the FOM being projected on a divergence-free subspace and expanded in a divergence-free basis to the aforementioned subspace the hROM is pressure-free and (4.7) is implicitly satisfied by any hROM solution. This makes it relatively straightforward to implement explicit Runge-Kutta integrators to integrate the hROM (4.37) in time, as the incompressibility constraint does not have to be explicitly enforced any longer. Using an s-stage explicit Runge-Kutta method the time-integration of the hROM (4.37) is performed as:

$$\boldsymbol{A}_i = \boldsymbol{a}^n + \Delta t\sum_{j=1}^{i-1} a_{ij}\left(-\Phi^TM\boldsymbol{c}(\boldsymbol{A}_j) + \nu D_r\boldsymbol{A}_j\right) \tag{4.55}$$

$$\boldsymbol{a}^{n+1} = \boldsymbol{a}^n + \Delta t\sum_{i=1}^{s} b_i\left(-\Phi^TM\boldsymbol{c}(\boldsymbol{A}_i) + \nu D_r\boldsymbol{A}_i\right), \tag{4.56}$$

where $\boldsymbol{A}_k$ is the stage vector of generalized coordinates $\boldsymbol{a}$ at stage k of the time-integration. The DEIM coordinates at the different stages $\boldsymbol{c}(\boldsymbol{A}_k)$ $k \in \{1, 2, ..., s\}$ are then evaluated using any of the previously discussed methods i.e. DEIM, LSDEIM, SMDEIM or DLSDEIM.

Time-integration using implicit Runge-Kutta methods might present more difficulties, even though the hROM is clearly still pressure-free and divergence-free by construction. Namely, implicit Runge-Kutta methods are characterized by the general Butcher tableau Table 3.1. Due to this implicit nature these methods require at least the current or later stage vectors in

evaluation of the slope at some stage k, thus:

$$\boldsymbol{A}_i = \boldsymbol{a}^n + \Delta t \sum_{j=1}^{s} a_{ij} \left(-\Phi^T M \boldsymbol{c}(\boldsymbol{A}_j) + \nu D_r \boldsymbol{A}_j \right) \tag{4.57}$$

$$\boldsymbol{a}^{n+1} = \boldsymbol{a}^n + \Delta t \sum_{i=1}^{s} b_i \left(-\Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) + \nu D_r \boldsymbol{A}_i \right). \tag{4.58}$$

Hence, in its generality, using implicit methods, any stage vector may depend on any other unknown stage vector and therefore (4.57) must be solved simultaneously for all stages. However, as the DEIM coordinates are determined from nonlinear functions, solving (4.57) requires solving large systems of nonlinear equations. To cope with the nonlinearity of the DEIM coordinates, (4.57) is solved for iteratively using the Newton-Raphson algorithm. The terms $\boldsymbol{c}(\boldsymbol{A}_j)$ in (4.57) are therefore linearized to obtain the iterative procedure:

$$\boldsymbol{A}_i^{k+1} = \boldsymbol{a}^n + \Delta t \sum_{j=1}^{s} a_{ij} \left(- \left[\Phi^T M \boldsymbol{c}(\boldsymbol{A}_j^k) + J_m(\boldsymbol{A}_j^k) \left(\boldsymbol{A}_j^{k+1} - \boldsymbol{A}_j^k \right) \right] + \nu D_r \boldsymbol{A}_j^{k+1} \right), \quad \forall\, i \in \{1, ..., s\}. \tag{4.59}$$

Equation (4.59) can be written in compact form for all $i \in \{1, ..., s\}$ as:

$$\left(I - \Delta t (A \otimes I_r) F_1(\boldsymbol{A}^k) \right) \boldsymbol{A}^{k+1} = \boldsymbol{e}_s \otimes \boldsymbol{a}^n + \Delta t (A \otimes I_r) F_2(\boldsymbol{A}^k) \tag{4.60}$$

where $\boldsymbol{A}^k \in \mathbb{R}^{s \cdot r}$ is a block-vector containing the stage vectors of the generalized coordinates in chronological order, I is the $(s \cdot r) \times (s \cdot r)$ identity matrix, I_r is the $r \times r$ identity matrix, $\boldsymbol{e}_s \in \mathbb{R}^s$ is a vector of all ones, $A \in \mathbb{R}^{s \times s}$ is a matrix such that $(A)_{ij} = a_{ij}$, $(\cdot) \otimes (\cdot)$ here denotes the Kronecker product, $F_1(\boldsymbol{A}) : \mathbb{R}^{s \cdot r} \to \mathbb{R}^{(s \cdot r) \times (s \cdot r)}$ is a function producing an $s \times s$ block diagonal matrix with nonzero blocks satisfying:

$$(F_1(\boldsymbol{A}))_{ii} = (-J_m(\boldsymbol{A}_i) + \nu D_r) \quad \forall i \in \{1, ..., s\},$$

$F_2(\boldsymbol{A}) : \mathbb{R}^{s \cdot r} \to \mathbb{R}^{s \cdot r}$ is function producing an s-dimensional block vector with blocks satisfying:

$$(F_2(\boldsymbol{U}))_i = - \left(\Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) - J_r(\boldsymbol{A}_i) \boldsymbol{A}_i \right).$$

The system in (4.60) is now solved repeatedly until a predetermined convergence criterion is reached, resulting in a set of stage vectors collected in a block vector $\boldsymbol{A}$. The converged stages are then used in (4.58) to determine the generalized coordinates $\boldsymbol{a}^{n+1}$ at the new time-step, completing the computation process of a new time-step.

4.6.2. Energy-Conserving Runge-Kutta Methods

At the semi-discrete level the hROM (4.37), using a structure-preserving DEIM algorithm, conserves both reduced total momentum and reduced total kinetic energy in the inviscid limit. To conserve reduced total kinetic energy K_r in the inviscid limit at the fully discrete level however, the time-integration scheme has to conserve the norm of the generalized coordinates $||\boldsymbol{a}||$ while integrating (4.37) for $\nu = 0$. In the following a short analysis is provided to show that for energy-conserving Runge-Kutta methods like the family of Gauss-Legendre integrators this is possible. Following the analysis provided in (3.25) it may be stated for the squared norm of the generalized coordinates found by integration using Runge-Kutta methods that:

$$\left|\left|\boldsymbol{a}^{n+1}\right|\right|_\Theta^2 - ||\boldsymbol{a}^n||_\Theta^2 = 2\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, f_i \right\rangle_\Theta - \Delta t^2 \sum_{i,j=1}^{s} m_{ij} \left\langle f_i, f_j \right\rangle_\Theta, \tag{4.61}$$

where $f_i = -\Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) + \nu D_r \boldsymbol{A}_i$ $i \in \{1, ..., s\}$. Taking $\Theta = I$ in (4.61), where clearly I is SPD such that it is captured by the definition of the Θ-inner product and induced norm, the reduced total kinetic energy changes as:

$$K_r^{n+1} - K_r^n = \Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, f_i \right\rangle - \frac{\Delta t^2}{2} \sum_{i,j=1}^{s} m_{ij} \left\langle f_i, f_j \right\rangle. \tag{4.62}$$

The first term in (4.62) can be written as:

$$\begin{aligned}\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, f_i \right\rangle &= \Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, -\Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) + \nu D_r \boldsymbol{A}_i \right\rangle \\ &= -\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, \Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) \right\rangle + \Delta t \sum_{i=1}^{s} b_i \nu \left\langle \boldsymbol{A}_i, D_r \boldsymbol{A}_i \right\rangle \\ &= -\Delta t \sum_{i=1}^{s} b_i \boldsymbol{A}_i^T \Phi^T M \boldsymbol{c}(\boldsymbol{A}_i) + \Delta t \sum_{i=1}^{s} b_i \nu \boldsymbol{A}_i^T D_r \boldsymbol{A}_i.\end{aligned}$$

Using a structure-preserving DEIM algorithm the first term on the right-hand side of the third line will be zero as condition (4.43) is satisfied for these methods. The second term on the right-hand side of the third line can be rewritten using the properties of the reduced diffusion operator to obtain:

$$\Delta t \sum_{i=1}^{s} b_i \left\langle \boldsymbol{A}_i, f_i \right\rangle = -\Delta t \sum_{i=1}^{s} b_i \nu \left|\left| Q_r \boldsymbol{A}_i \right|\right|^2.$$

As (3.27) holds for energy-conserving Runge-Kutta methods, the change in reduced total kinetic energy K_r using these methods is given as:

$$\boxed{K_r^{n+1} - K_r^n = -\Delta t \sum_{i=1}^{s} b_i \nu \left|\left| Q_r \boldsymbol{A}_i \right|\right|^2}. \tag{4.63}$$

Thus in the inviscid limit it can be seen from (4.63) that the hROM (4.37) using the LSDEIM, SMDEIM or DLSDEIM integrated using energy-conserving Runge-Kutta methods conserves reduced kinetic energy. Additionally, in the viscous case the resulting hROM will be nonlinearly stable when $b_i \geq 0 \; \forall \; i \in \{1, ..., s\}$, a condition that holds for the implicit midpoint rule and Gauss-Legendre 4 method.

4.7. Bypassing the Kolmogorov Barrier

Most fluid flows that are of interest to engineering applications are characterized by high Reynolds numbers, meaning they are dominated by convection phenomena. This has significant effects on the performance of linear projection-based ROMs like those obtained using the POD-Galerkin method. Namely, convection-dominated flows are well-known to be difficult to capture in a low-dimensional linear subspace of the state space $\mathbb{R}^N$, where the discrete flow field's solution manifold $\mathcal{M}_u^h$ resides. This also holds for the POD basis found from a snapshot matrix X_{T_s} associated to a FOM simulation of a significantly longer duration T_s than a characteristic timescale $\mathcal{T} \ll T_s$ of e.g. a large scale turbulent eddy. That is to say of a duration such that substantial spatial transport of any coherent structures in the flow can take place.

The reason that transport phenomena are difficult to capture for linear and projection-based ROMs is that the associated solution manifold $\mathcal{M}_u^h$ typically is a trajectory traversing a considerable portion of the phase space $\mathbb{R}^N$. For example, consider the solution manifold of a thin traveling wave under linear advection [18]. Many of its states are not correlated as their inner-product equals zero. Geometrically, this implies that the solution indeed visits a large portion of the phases space and that it cannot be accurately embedded on a low-dimensional linear manifold. Analytically, it shows that the correlation matrix $X^T X$ of its associated snapshot matrix nearly equals some scalar multiple of the identity matrix I. As the eigenvalues of the identity matrix, and thus approximately the singular values of X, do not decay there are no POD modes that carry significantly more energy than others. Consequently, the solution manifold cannot be well-expressed in a low-dimensional basis of POD modes. Furthermore, the short timescales present in turbulent flows are difficult to resolve by global linear methods, like the POD, that attempt to approximate the entire snapshots matrix. Instead, all short timescales are smeared out over the full duration of the snapshot set such that the POD basis is only accurate in an average sense.

A measure for how well a problem can be reduced in dimensionality is given by the decay of the Kolmogov N-width $d(\mathcal{M}_u^h) : \mathbb{M}_u^N \to \mathbb{R}^+$:

$$d(\mathcal{M}_u^h) := \inf_{\mathcal{V}\subset\mathbf{R}^N;\dim(\mathcal{V})=r} \sup_{\boldsymbol{u}\in\mathcal{M}_u^h} \inf_{\boldsymbol{v}\in\mathcal{V}} ||\boldsymbol{u}-\boldsymbol{v}||_2,$$

where $\mathbb{M}_u^N$ is a set of all possible solution manifolds in $\mathbb{R}^N$. Essentially, the Kolmogorov N-width is a measure of the largest error that occurs between $\mathcal{M}_u^h$ and the best possible solution trajectory approximating $\mathcal{M}_u^h$ that is restricted to the best possible r-dimensional subspace of $\mathbb{R}^N$. The decay $d(\mathcal{M}_h^h) \to 0$ as $r \to N$ indicates how well the problem can be represented in a low-dimensional linear subspace of $\mathbb{R}^N$. Indeed, this decay is typically slow for convection-dominated flows. However, the Kolmogorov N-width decay is often difficult to determine, instead the decay of the singular values of a respective snapshot matrix is mostly used as a proxy for the Kolmogorov N-width.

Finding ways around the slow Kolomogorov N-width decay, also referred to as the Kolmogorov barrier, has been the topic of considerable research. To deal with the Kolmogorov barrier in this thesis the application of the principal interval decomposition (PID) [49] for both the DEIM and reduced space construction will be proposed. The use of the PID or localization in other forms for the construction of reduced spaces is not new. In [18, 4] the PID was used to construct reduced spaces specifically to overcome the difficulties associated to convection-dominated flows, in [71] the construction of DEIM spaces was localized in parameter and state space and in [41] the ECSW hyper-reduction method was constructed for parametric and temporal intervals on the solution manifold. In [32] temporal localization has been applied for the DEIM, but not in a structure-preserving framework.

4.7.1. Temporal Localization of Reduced and DEIM Spaces Using PID

The premise of the PID is to decompose the snapshot set over n_p intervals in time $[t_i, t_{i+1}]$ and apply the POD algorithm to the individual intervals. The hope is then that by calculating modes tailored to specific intervals, the local timescales within the respective interval are captured significantly better than by a set of modes calculated from the full set of snapshots. Based on snapshot sets:

$$X = [X_0, X_1, ..., X_{n_p-1}], \quad \Xi = [\Xi_0, \Xi_1, ..., \Xi_{n_p-1}],$$

the PID provides sets of POD modes:

$$\Phi_i, M_i, \ i \in \{0, 1, ..., n_p - 1\}$$

applicable to use at times $t \in [t_i, t_{i+1}]$ within their respective intervals. It is not necessary to use the same number of intervals for the calculation of either bases, however for simplicity this will be the case in this thesis. Using the PID, the POD modes now solve:

$$\widetilde{\Phi}_i = \underset{\Phi^*\in\mathbb{R}^{N\times m}}{\arg\min} \left|\left|\widetilde{X}_i - \Phi^*\Phi^{*T}\Omega_h\widetilde{X}_i\right|\right|_F^2 \quad \text{s.t.} \quad \Phi^{*T}\Omega_h\Phi^* = I$$

$$M_i = \underset{\tilde{M}\in\mathbb{R}^{N\times m}}{\arg\min} \left|\left|\Xi_i - \tilde{M}\tilde{M}^T\Xi_i\right|\right|_F^2 \quad \text{s.t.} \quad \tilde{M}^T\tilde{M} = I.$$

where Φ_i is calculated from $\widetilde{\Phi}_i$ using steps 4 and 5 from Algorithm 1 and $\widetilde{X}_i = \Omega_h^{1/2}\left[X_i - EE^T\Omega_h X_i\right]$. The temporally localized DEIM measurement space $\mathcal{P}_h^i$ for the i^{th} interval should now also be determined solely based on the operator snapshots in Ξ_i using Algorithm 2 specifically for Ξ_i.

Setting up the hROM using the PID, the dynamical system (4.37) now takes the form:

$$\frac{d\boldsymbol{a}_i}{dt} = -\Phi_i^T M_i \boldsymbol{c}_i(\boldsymbol{a}_i) + \nu D_r^i \boldsymbol{a}_i \quad t \in [t_i, t_{i+1}], \tag{4.64}$$

the notation $D_r^i = \Phi_i^T D_h \Phi_i$ has been introduced to not clutter the subscripts. Note that it is also necessary to introduce subscripts for the generalized and DEIM coordinates $\boldsymbol{a}_i$ and $\boldsymbol{c}_i$ respectively, as they are only valid during the interval $[t_i, t_{i+1}]$. The DEIM coordinates are calculated using either the DEIM, LSDEIM, SMDEIM or DLSDEIM algorithms using the

appropriate measurement spaces and DEIM and POD bases. It should be noted that all the conservation properties of the previously discussed structure-preserving DEIM methods hold within intervals. This is somewhat trivial as nothing changes in the ways the DEIM coordinates are calculated. In what follows, a hROM using the PID and a certain structure-preserving DEIM algorithm will be referred to as PID-DLSDEIM hROM when, for example, the DLSDEIM is used.

The reason why the PID might give improved solutions over ordinary POD is that it has the potential to more accurately capture nonlinearity in the FOM solution manifold $\mathcal{M}_u^h$. Specifically, where the POD provides a low-dimensional linear subspace approximation of the full solution manifold, the PID provides linear subspaces for individual intervals of the solution manifold. Hence, a piecewise linear subspace is provided to approximate the full solution manifold, allowing a sense of nonlinearity of the full reduced space. However using the PID comes at a price, precomputed operators have to be calculated for every individual interval. This may have negative effects on the computational costs of the offline phase of the calculation. When the offline phase becomes restrictive due to this method it is suggested to consider online adaptive methods [74].

4.7.2. Transitioning between Intervals

After integrating (4.64) over an interval using a suitable time-integration scheme, the reduced velocity vector $\boldsymbol{u}_r(t_{i+1})$ expressed in the generalized coordinates $\boldsymbol{a}_i \in \mathbb{R}^{r_i}$ has to be mapped to an expression in terms of the generalized coordinates $\boldsymbol{a}_{i+1} \in \mathbb{R}^{r_{i+1}}$, where $r_i \neq r_{i+1}$ in general. In the setting of Galerkin projection a natural choice for this transition mapping is the following:

$$\text{find} \quad \boldsymbol{a}_{i+1} \in \mathbb{R}^{r_{i+1}} \quad \text{s.t.} \quad \left\langle \boldsymbol{u}_r^-(t_{i+1}) - \boldsymbol{u}_r^+(t_{i+1}), (\Phi_{i+1})_{j,} \right\rangle_{\Omega_h} = 0 \quad \forall j \in \{1, 2, ..., r_{i+1}\}, \tag{4.65}$$

where $\boldsymbol{u}_r^-(t_{i+1}) = \Phi_i \boldsymbol{a}_i$ and $\boldsymbol{u}_r^+(t_{i+1}) = \Phi_{i+1}\boldsymbol{a}_{i+1}$. Condition (4.65) selects the new generalized coordinates $\boldsymbol{a}_{i+1}$ such that the residual of the approximation is orthogonal to the new reduced space $\text{span}(\Phi_{i+1})$ in the Ω_h-inner product. Solving condition (4.65) results in a precomputable and low-dimensional mapping for $\boldsymbol{a}_{i+1}$ in terms of $\boldsymbol{a}_i$:

$$\boldsymbol{a}_{i+1} = \Phi_{i+1}^T \Omega_h \Phi_i \boldsymbol{a}_i.$$

Unfortunately, this transition mapping does not preserve-structure. The reduced total kinetic energy and reduced total momentum are the following for either side of the interval boundary:

$$(P_r^k)^i = \boldsymbol{e}_k^T \Omega_h \Phi_i \boldsymbol{a}_i, \qquad (P_r^k)^{i+1} = \boldsymbol{e}_k^T \Omega_h \Phi_{i+1} \boldsymbol{a}_{i+1} = \boldsymbol{e}_k^T \Omega_h \Phi_{i+1} \Phi_{i+1}^T \Omega_h \Phi_i \boldsymbol{a}_i,$$
$$K_r^i = \frac{1}{2} \boldsymbol{a}_i^T \boldsymbol{a}_i, \qquad K_r^{i+1} = \frac{1}{2} \boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1} = \frac{1}{2} \boldsymbol{a}_i^T \Phi_i^T \Omega_h \Phi_{i+1} \Phi_{i+1}^T \Omega_h \Phi_i \boldsymbol{a}_i.$$

For $(P_r^k)^{i+1} = (P_r^k)^i$ to hold, the following must hold $\Phi_{i+1}\Phi_{i+1}^T \Omega_h = I$, note that this is equivalent to $\Omega_h \Phi_{i+1} \Phi_{i+1}^T = I$. However generally the matrix $\Omega_h \Phi_{i+1} \Phi_{i+1}^T$ is not equal to the identity matrix. For $K_r^{i+1} = K_r^i$ to hold, it must hold that $\Phi_i^T \Omega_h \Phi_{i+1} \Phi_{i+1}^T \Omega_h \Phi_i = I$. A condition that is in general also not satisfied. Thus, although structure is preserved within the interval, upon transitioning between intervals the preservation of structure is lost.

Structure-preserving transition mappings could be derived by considering that the solution to condition (4.65) solves the following least squares problem:

$$\boldsymbol{a}_{i+1} = \underset{\boldsymbol{a} \in \mathbb{R}^{r_{i+1}}}{\arg\min} \left|\left| \Phi_i \boldsymbol{a}_i - \Phi_{i+1} \boldsymbol{a} \right|\right|_{\Omega_h}. \tag{4.66}$$

To preserve structure one may consider adding constraints to this minimization problem that enforce reduced total kinetic energy and reduced total momentum conservation. Hence a new structure-preserving interface condition is proposed in this thesis in the form of a constrained minimization problem:

$$\boxed{\boldsymbol{a}_{i+1} = \underset{\boldsymbol{a} \in \mathbb{R}^{r_{i+1}}}{\arg\min} \left|\left| \Phi_i \boldsymbol{a}_i - \Phi_{i+1} \boldsymbol{a} \right|\right|_{\Omega_h} \quad \text{s.t.} \quad \frac{1}{2} \boldsymbol{a}^T \boldsymbol{a} = K_r^i, \quad \boldsymbol{e}_k^T \Omega_h \Phi_{i+1} \boldsymbol{a} = (P_r^k)^i}.$$

If it is only desired to guarantee nonlinear stability a less constrained minimization problem may be solved as transition mapping:

$$\boxed{\boldsymbol{a}_{i+1} = \underset{\boldsymbol{a}\in\mathbb{R}^{r_{i+1}}}{\arg\min} ||\Phi_i \boldsymbol{a}_i - \Phi_{i+1}\boldsymbol{a}||_{\Omega_h} \quad \text{s.t.} \quad \frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} \leq K_r^i}.$$

For the rest of this thesis only the transition mapping obtained from (4.65) will be considered [2]. It is expected that, due tot the accuracy of the reduced bases and the optimality of the transition mapping in the sense that it minimizes (4.66), severe deviations from exact structure-preservation will not take place in many practical scenarios.

[2] Due to time constraints.

Pros and Cons

A brief summary in the form of Table 4.1 is provided to gather all expected pros and cons of the proposed structure-preserving hyper-reduction methods. In short, both the LSDEIM and DLSDEIM provide guaranteed unique solutions, this is a strong point in favor of their robustness. Furthermore, as the DLSDEIM employs oversampling, it can be expected that its approximations may be more accurate. This does come at the cost of more function evaluations making the DLSDEIM a more expensive method. The nature of the optimization problem underlying the the LSDEIM and DLSDEIM also forms a con as it is difficult to say a priori how accurate the methods will be. This is the strong point of the SMDEIM. Namely, it does provide exact correspondence in the measurement space. However, for the SMDEIM there are no guarantees on the underlying linear system of the SMDEIM being non-singular. Finally, the LSDEIM and DLSDEIM provided consistency.

	LSDEIM	SMDEIM	DLSDEIM
Pros	• Existence and uniqueness • Consistency	• Exact correspondence in $\mathcal{P}_h$	• Existence and uniqueness • Consistency • Larger measurement space
Cons	• Not clear a priori how accurate minimizer is	• Invertibility not guaranteed	• Not clear a priori how accurate minimizer is • Expensive

Table 4.1: Table summarizing pros and cons of the proposed methods.

5

Results and Discussion

In this chapter the structure-preserving DEIM algorithms developed in the previous chapter will be tested to determine their robustness, accuracy and efficiency in modelling convection-dominated flows. Two test cases, both on periodic domains, will be considered to this end. The first test case is the roll-up of a shear layer which will be used to determine which hyper-reduction algorithm is the most accurate and provides the most computational gain with respect to the FOM. Robustness will also be checked with this test case. The hyper-reduction method with the overall best performance will be used in the second test case, which is a two-dimensional isotropic and freely decaying turbulent flow. Using this test case the capability of the hROM to capture the relevant physical phenomena of this type of turbulence in the reduced spaces $\mathcal{V}$ and $\mathcal{M}_d$ is analysed. As will be discussed shortly there are significant differences between two-dimensional turbulence and three-dimensional turbulence, however being able to accurately model two-dimensional turbulence is considered as a minimum requirement for the feasibility of simulating three-dimensional turbulence.

5.1. The Test Suite

In what follows the previously mentioned test cases will be described. The acquired understanding will aid in the construction and interpretation of numerical experiments to test and compare the proposed hROMs.

5.1.1. Shear Layer Roll-Up

The shear layer roll-up [62, 52] is a flow on a double-periodic domain $[0, 2\pi] \times [0, 2\pi]$ with a central band of flow in a positive coordinate direction and neighbouring bands of flow in the opposite direction. The bands are joined with a thin region of very strong velocity gradients and hence strong shear-forces. The flow has the following initial conditions:

$$u(x, y, 0) = \begin{cases} \tanh\left(\frac{y-\pi/2}{\delta}\right), & y \leq \pi \\ \tanh\left(\frac{3\pi/2-y}{\delta}\right), & y > \pi \end{cases}, \quad v(x, y, 0) = \epsilon \sin(x).$$

The parameter $\delta \in \mathbb{R}$ determines the initial thickness of the shear layers and the parameter $\epsilon \in \mathbb{R}$ determines the initial amplitude of an unstable perturbation in the second coordinate direction to trigger the so-called roll-up. The value for these parameters will be kept constantly at $\delta = \frac{\pi}{15}$ and $\epsilon = 0.05$ in this thesis. The roll-up refers to the behaviour of the flow after the perturbation has grown to significant size. Namely, this is when the thin layers of strong shear spiral into a vortex like pattern until local gradients become sufficiently strong for the energy to be dissipated. The flow has become a useful benchmark to test numerical methods for high Re as high Reynolds numbers will postpone the dissipation process and induce very rapidly fluctuating spatial velocity gradients, which are challenging to capture numerically.

The presence of strong spatial gradients motivates the use of the shear layer roll-up in this thesis to study the effect of different reduced and DEIM space dimensions on the numerical

error between the hROM and the FOM for different values of Re. Furthermore, before the roll-up stage the flow is sufficiently simple to do inviscid calculations without any spurious oscillations using the previously introduced FOM. This means that some high-fidelity data can also be obtained to construct fully inviscid hROMs to test the energy conservation properties of the structure-preserving DEIM algorithms proposed in the previous chapter.

5.1.2. Freely Decaying Two-Dimensional Turbulence

Freely decaying two-dimensional turbulence [35, 54, 15, 83, 4] (2DT) is a complex flow to model with strong vortex dynamics and behaviour according to some distinct physics. This test case will serve to test if such physics can be captured in reduced spaces using a hROM and to determine if the Kolmogorov N-width decay of the convection-dominated vortex dynamics can be increased using the PID. A brief discussion of the phenomenology of freely decaying two-dimensional turbulence will be provided such that the experiments in what follows can be interpreted. In this thesis the initial conditions for the 2DT will be formed from an adapted version of those proposed in [91]. A 16×16 lattice of pairwise oppositely rotating vortices will be placed in the centre of a periodic domain $\Omega = [0, 1] \times [0, 1]$. To trigger the complex motion the centres of all vortices will be perturbed according to a normal distribution $\mathcal{N}(\mu, \sigma)$ with mean $\mu = 0$ and standard deviation $\sigma = 0.01$. The initial flow field is obtained from numerical differentiation of the streamfunction $\psi_0(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}$:

$$\psi_0(\boldsymbol{x}) = \sum_{i,j=2}^{17} 0.05 \cdot (-1)^{i+j} \cdot e^{-2000\left[\left(x+\mathcal{N}_{ij}^x(0,0.01)-\frac{j}{19}\right)^2+\left(y+\mathcal{N}_{ij}^y(0,0.01)-\frac{i}{19}\right)^2\right]},$$

evaluated in the vertices of pressure cells on the numerical grid. Here:

$$u_0(\boldsymbol{x}) = \frac{\partial \psi_0}{\partial y}, \quad v_0(\boldsymbol{x}) = -\frac{\partial \psi_0}{\partial x}.$$

The numerical differentiation operation is such that its resulting discrete velocity field is an element of $\ker(M_h)$ and hence is discretely divergence-free [43].

Phenomenology

The dynamics of two-dimensional turbulence are strongly associated to the behaviour in two spatial dimensions of a quantity referred to as vorticity. Vorticity can be interpreted as measuring the rotation of a differential fluid element and is defined (in full-three dimensional space) as:

$$\boldsymbol{\omega} := \nabla \times \boldsymbol{u} \quad \xrightarrow{2D} \quad \omega := \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}.$$

Note that vorticity essentially becomes a scalar under the assumption of two-dimensional flow. An evolution equation for this quantity can be found by taking the curl of (2.13):

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{\omega} = \underbrace{(\boldsymbol{\omega} \cdot \nabla)\boldsymbol{u}}_{\text{vortex stretching}} + \nu \Delta \boldsymbol{\omega} \quad \xrightarrow{2D} \quad \frac{\partial \omega}{\partial t} + (\boldsymbol{u} \cdot \nabla)\omega = \nu \Delta \omega. \tag{5.1}$$

In two-dimensional flow the mechanism of vortex stretching can no longer take place, consequently the physics of 2DT exhibit notable differences to three-dimensional turbulence. Another set of equations in two spatial dimensions that, together with (5.1), are central to classical phenomenological discussions of 2DT is the set:

$$\frac{d\mathcal{K}}{dt} = -2\nu\mathcal{E} \tag{5.2}$$

$$\frac{d\mathcal{E}}{dt} = -2\nu\mathcal{P}_a \tag{5.3}$$

$$\frac{d\mathcal{P}_a}{dt} = \Pi_a - \mathcal{S}_a, \tag{5.4}$$

where $\mathcal{E} := \frac{1}{2}||\omega||^2_{L^2(\Omega)} \geq 0$ is referred to as total enstrophy and $\mathcal{P}_a = \frac{1}{2}||\nabla\omega||^2_{L^2(\Omega)} \geq 0$ is referred to as total palinstrophy. The terms Π_a and $\mathcal{S}_a$ are terms that behave as a source and a

sink of $\mathcal{P}_a$, respectively. From equations (5.1) - (5.4) several distinct phenomenological features can be extracted. In this thesis it will be tested if the proposed hROMs can reproduce these features and in what follows the features will be introduced briefly.

Kinetic energy
First the behaviour of the total kinetic energy $\frac{d\mathcal{K}}{dt}$ in the limit $\nu \to 0$ will be considered. As (5.3) shows that $\mathcal{E}$ is bounded from above by its initial value it can be stated that:

$$\lim_{\nu \to 0} \frac{d\mathcal{K}}{dt} = 0. \tag{5.5}$$

This result is in stark contrast with three-dimensional turbulence where a source term in (5.3) due to vortex-stretching allows $\mathcal{E}$ to attain arbitrarily large values. In turn the limit in (5.5) for three-dimensional turbulence becomes nonzero [35]. This reflects the so-called energy cascade where kinetic energy is passed from large to progressively smaller turbulent spatial structures referred to as eddies. Finally arriving at sufficiently small spatial structures, the energy is dissipated to heat through viscosity. As $\nu \to 0$ the three-dimensional turbulence will simply give rise to smaller eddies such that energy can always be dissipated. This does not happen for 2DT and hence it is very long-lived for high Re.

The enstrophy cascade
Equation (5.1) shows that in the limit $\nu \to 0$, isovortical lines (lines of constant ω) start evolving like material lines (a curve following distinct sets of fluid particles [31]). Material lines are commonly believed to be continually stretched in chaotic vortex flows. Some simple heuristic arguments confirming this belief are provided in [15]. Hence, isovortical lines for $\nu \to 0$ will also be stretched. Consequently, neighbouring isovortical lines will come closer together increasing local vorticity gradients and thus local values of $\mathcal{P}_a$. This process is referred to as vorticity filamentation. A graphical representation of vorticity filamentation is provided in [35].

Note that due to neighbouring isovortical lines coming progressively closer through vorticity filamentation, the enstrophy is increasingly associated to the smaller length scales of the flow. This continues until length scales become sufficiently small such that viscosity can dissipate local values of $\mathcal{E}$. As the rate of filamentation is determined by large scale vortices and the filamentation process transports enstrophy to smaller length scales, it is implied that enstrophy does follow a cascade process in 2DT. This is supported by the fact that $\mathcal{P}_a$ can take arbitrarily large values such that:

$$\lim_{\nu \to 0} \frac{d\mathcal{E}}{dt} > 0,$$

analogously to the energy cascade in three-dimensional turbulence. These ideas are well-established in classical literature on 2DT [15, 54].

The energy spectrum
Consider eddies of length scale $\mathcal{O}(l)$ in the so-called inertial range where $L \gg l \gg \eta$ and where L are the largest and η the smallest turbulent length scales in the flow. If there is an enstrophy cascade process present in the flow these eddies should only be aware of eddies of slightly larger and slightly smaller length scales and not of the length scales at the top and bottom of the cascade. Hence, spectral properties associated to the 2DT for these length scales can only scale with the local wavenumber k (or wavevector magnitude), time t and $\beta := \nu\mathcal{P}_a$. Indeed, β determines the rate of enstrophy dissipation and thus the rate at which enstrophy must pass through the cascade. Using (5.3) it may be shown that β has dimension $\beta \sim t^{-3}$ [35]. Following dimensional analysis and arguments of self-similarity (see [35]) a scaling may then be found for the spectral distribution of kinetic energy $E(k)$ over the inertial range:

$$E(k) \sim \beta^{2/3} k^{-3} \sim t^{-2} k^{-3},$$

a result first described specifically for the freely evolving version of 2DT considered in this thesis in [15].

The inverse energy cascade
In 2DT the energy does not cascade to small scale structures but accumulates in large spatial scales. Hence, there is an inverse cascade of energy in 2DT. The inverse cascade will only be briefly touched upon as there is an abundance of different explanations for this phenomenon [35]. The simplest explanation is that like-signed vortices tend to merge. Therefore, as time proceeds progressively larger vortices start to appear. As a consequence, kinetic energy is seen to redistribute to larger length scales.

5.2. Results: Shear Layer Roll-Up

In this section results will be described of numerical experiments carried out on the proposed structure-preserving hyper-reduction methods using the SLR flow. Here the methods will be tested to determine their structure-preserving capabilities, accuracy and computational performance. Structure-preservation will be tested by considering the temporal evolution of reduced conserved quantities. Accuracy will be tested by considering the error behaviour as a function of time, Re and reduced space sizes. Finally, the computational performance will be tested by measuring execution times associated to the off- and online phases of the hROMs and comparing these the FOM. The best performing hyper-reduction method will then be used in section 5.3 for the challenging two-dimensional turbulence test case.

5.2.1. FOM Convergence Study

However, before this will be done, a convergence study is performed on the SLR flow to find out when the numerical FOM solution is sufficiently grid converged. When computationally feasible, numerical experiments will be held on these fine grid sizes. Whenever this is not possible due to a large amount of repeated precomputations being necessary to perform an experiment, less fine grids will be used. Using a Reynolds number $Re = 1000$ and a timestep size of $\Delta t = 0.01$, the grid sizes $\{128 \times 128,\ 256 \times 256,\ 512 \times 512,\ 1024 \times 1024\}$ are tested in order of increasing size. Due to the staggered grid used in the FOM actual numerical solution values of u- and v-unknowns are not available in the same spatial location. Therefore, the u-velocity components will be analysed on the line $(x = \pi, y)$ and the v-velocity components will be tested on the line $(x, y = \frac{\pi}{2})$. Note that due to the choice to change the grid size in factors of two, there are always unknowns present on these lines in the grid. Furthermore, these specific lines were chosen as the numerical solution values demonstrated the most spatially and temporally varying behaviour in these locations.

The results of the grid convergence study are displayed in Figure 5.5. It is clear that at a grid size of 512×512 the numerical solution largely overlaps with the numerical solution on the finer grid 1024×1024 and that 256×256 is also still close to the solution using a 1024×1024 grid. As 1024×1024 is a quite demanding grid size, both in terms of computational effort to integrate the solution in time and memory requirements to save snapshot matrices, the choice will be made to use either 512×512 or 256×256 grids. Some snapshots of the flow are provided in Figure 5.1 and Figure 5.2 in terms of absolute velocity:

$$U(\boldsymbol{x}, t) = \sqrt{u(\boldsymbol{x}, t)^2 + v(\boldsymbol{x}, t)^2},$$

calculated by interpolating the respective unknowns to pressure cell centres. In Figure 5.3 and Figure 5.4 the corresponding 2D vorticity values are provided which are calculated by taking second order central differences centred on the vertices of pressure cells. In comparison to the results of [83] the contour lines of the vorticity seem equally smooth in the rolled up shear layer, providing some additional confidence in the convergence of the solution.

5.2.2. hROM Conservation

Attention is now turned to the hROMs. The proposed structure-preserving hyper-reduction methods will be tested for their conservation properties. Both the reduced total momentum P_r^k (here k is used to indicate either u and v) and reduced total kinetic energy K_r will be calculated for all hROMs as a function of time and Reynolds number using both energy-conserving and standard Runge-Kutta time-integration methods. Where P_r^k should be conserved for any Reynolds number on a periodic domain and any time time integration method, K_r should

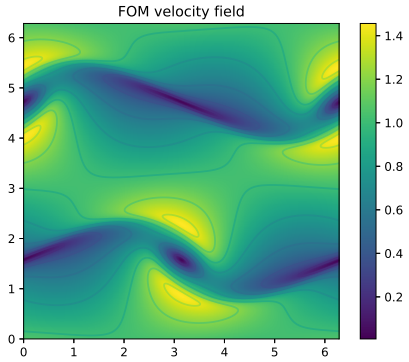


Figure 5.1: SLR velocity field at $t = 5, Re = 1000$ on a 512×512 numerical grid.

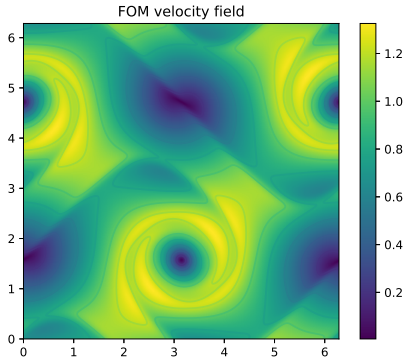


Figure 5.2: SLR velocity field at $t = 8, Re = 1000$ on a 512×512 numerical grid.

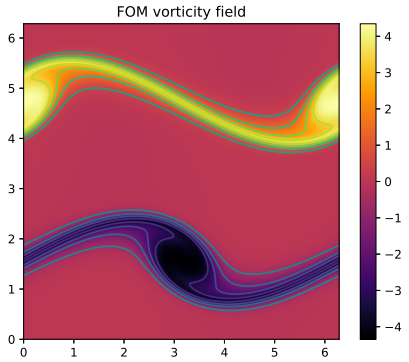


Figure 5.3: SLR vorticity field at $t = 5, Re = 1000$ on a 512×512 numerical grid.

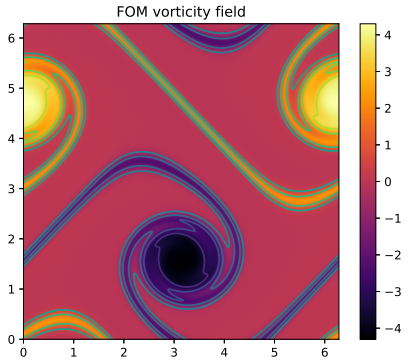
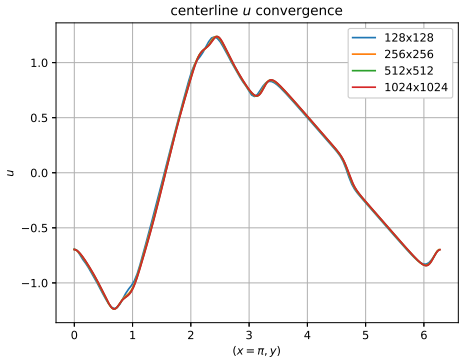


Figure 5.4: SLR vorticity field at $t = 8, Re = 1000$ on a 512×512 numerical grid.

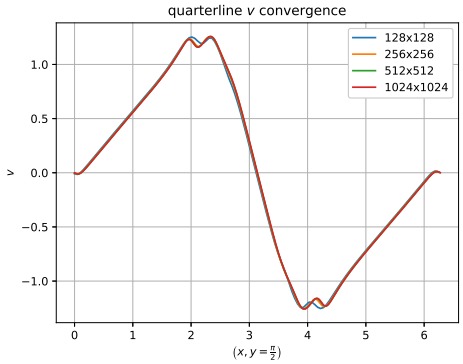
only be conserved for the fully inviscid case using energy-conserving Runge-Kutta methods. Due to the nonlinear stability of the hROMs, K_r should monotonically decrease for nonzero Re, where the rate of energy dissipation increases as Re is decreased. The DEIM, tensor decomposition, SMDEIM, LSDEIM and DLSDEIM will be tested for the Reynolds numbers $Re \in \{100, 1000, \text{inviscid}\}$ using a timestep size $\Delta t = 0.01$, similarly to the FOM, and both the GL4 and RK4 Runge-Kutta methods on a 256×256 grid. Time integration will take place until $t = 4$. The reason for this is that the inviscid case will develop numerical oscillations shortly after this time. It should be noted that these numerical oscillations will generally not destabilize the FOM solution due to the nonlinear stability property, but they render the simulation results inaccurate (and useless). Consequently, they will corrupt the snapshot data set and result in oscillatory POD modes. In turn the DEIM algorithms will place measurement points around the location of these oscillations and produce highly sub-optimal approximations. As the flow until $t = 4$ does not exhibit any quickly varying spatio-temporal behaviour the FOM solution until this point has a quickly decaying Kolmogorov N-width. Therefore usage of only 8 POD and 8 DEIM modes in the following conservation experiments suffices to accurately reduce the FOM dimensionality. The POD and DEIM modes are obtained from snapshots taken at every individual timestep of the FOM simulation.

Momentum

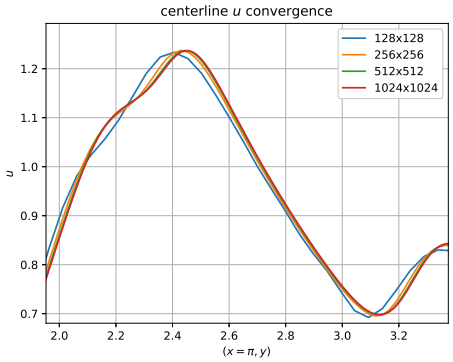
Figure 5.6 shows the reduced total momentum as a function of time and Re for all discussed hyper-reduction methods using both the GL4 and RK4 time-integrators. The discrete total momentum for the FOM is also shown. Here the RK4 time-integration scheme was used and the Reynolds number had a value $Re = 1000$. The reduced total momentum evolution profiles



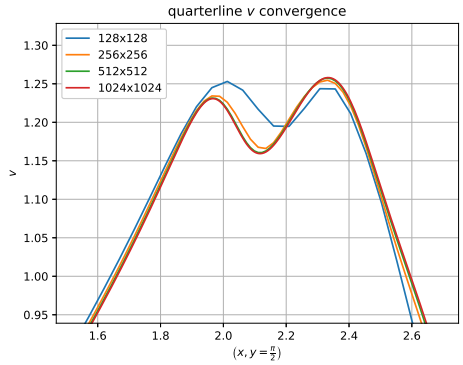
(a) Convergence of u velocity components.



(b) Convergence of v velocity components.



(c) Convergence of u velocity components (zoomed).



(d) Convergence of v velocity components (zoomed).

Figure 5.5: Convergence study of u and v velocity unknowns along the centreline and quarterline of the numerical domain at $t = 8, Re = 1000$.

are very similar among all hyper-reduction methods. They vary smoothly around zero[1] in the machine-precision range $\mathcal{O}(10^{-14})$. Similar accuracy in momentum conservation seems to be attained in [86], which uses only the tensor decomposition (abbreviated as 'tens. dec.'). As the momentum evolution for the tensor decomposition and DEIM algorithms are nearly identical and the smooth excursion from the more noisy evolution of the FOM momentum are of order $\mathcal{O}(10^{-14})$, the momentum will be considered conserved. Note that the conventional DEIM equally conserves P_r^k as the theory in chapter 4 has stated. Furthermore, the smooth behaviour of the momentum evolution for the DEIM algorithms and the tensor decomposition, in contrast to the noisy results for the FOM, will be attributed to both the implementation details as well as the precision of Armadillo's SVD implementation.

Energy

Figure 5.7 shows the reduced total kinetic energy as a function of time and Re for all discussed hyper-reduction methods using both the GL4 and RK4 time-integrators. A panel with only the inviscid case for all methods is also provided. It can clearly be observed for all methods that as the Reynolds number is decreased the rate of energy dissipation increases, as expected. Furthermore, looking at the panel containing only the inviscid cases, it can be seen that all hyper-reduction methods except for the conventional DEIM conserve reduced total kinetic energy. Indeed, where the structure-preserving DEIM algorithms seem to exactly conserve the norm of the generalized coordinates, the norm of the DEIM's generalized coordinates appears to be oscillating around its initial value. Moreover, the amplitude of the oscillation is increasing which could possibly result in instability of the hROM. The presence of sufficient diffusion seems

[1] Note that $P^k(t = 0) = 0$ and $\frac{dP}{dt} = 0$.

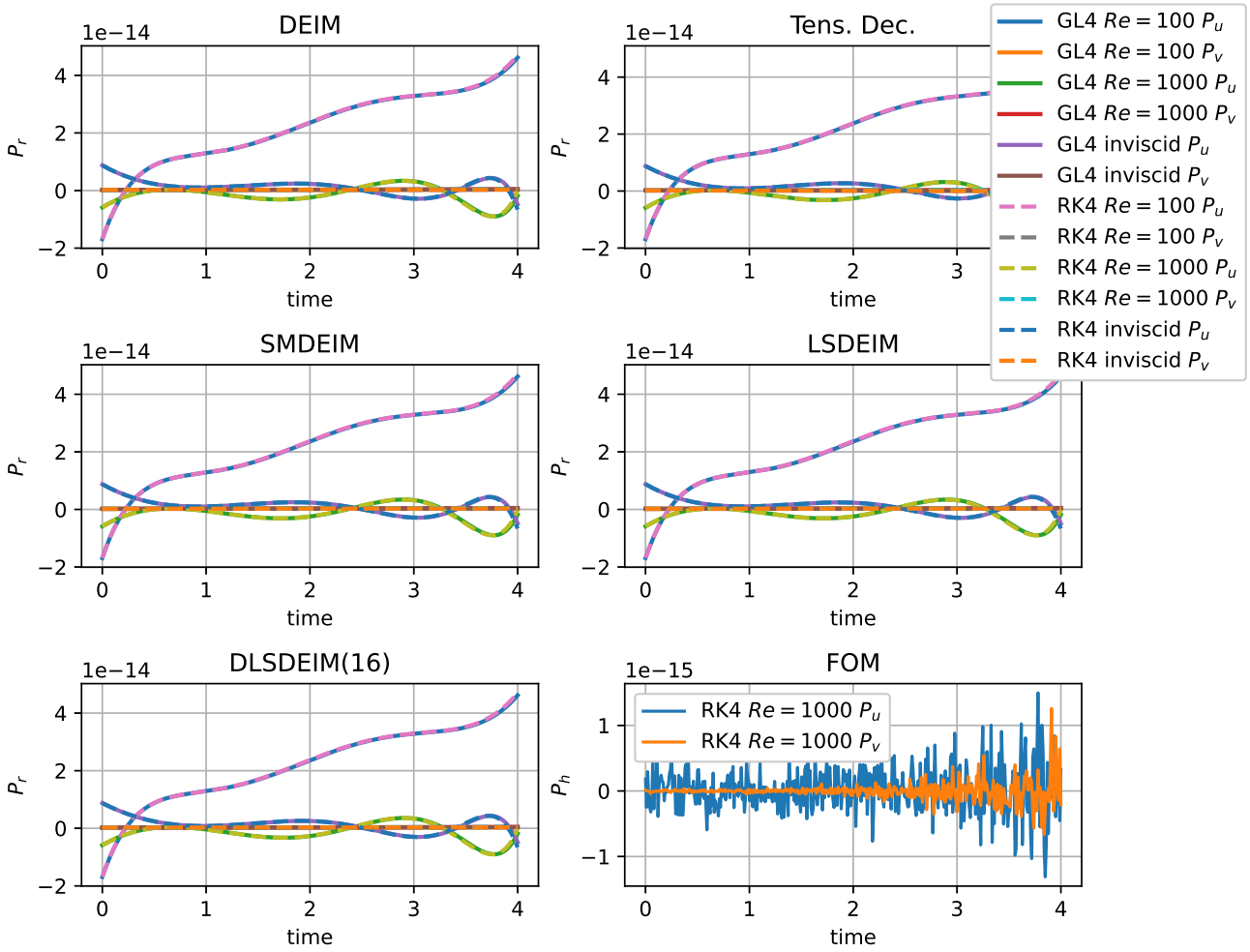


Figure 5.6: Momentum conservation properties for different hyper-reduction methods and Reynolds numbers using 8 POD and 8 DEIM modes.

stabilize the evolution of the kinetic energy of the conventional DEIM. However, for highly turbulent simulations with large values of Re this may not be enough. Another noteworthy observation that can be made from the panel containing the inviscid case is that the use of the RK4 time-integrator does not seem to induce significant errors in the kinetic energy evolution for the structure-preserving DEIM variants, even though it is not an enegy-conserving time-integrator. In [86] equivalent observations are made on the influence of high-order explicit time-integrators on the reduced total kinetic energy (given an energy-conserving semi-discretization of the hROM).

5.2.3. Error Comparison

Having concluded that the proposed DEIM algorithms are capable of preserving the structure of the underlying model at the ROM level, the next question to consider is what method is the most accurate. Before this can be assessed some error metrics must be introduced. The first error metric is the error between the hROM and the FOM; this error will be measured in Ω_h-norm:

$$\epsilon_u(t) = ||\boldsymbol{u}_h(t) - \boldsymbol{u}_r(t)||_{\Omega_h} ,$$

clearly this metric provides a measurement of how far the predicted hROM solution is from the high-fidelity data calculated by the FOM at a given instance in time. Using this metric it is assumed that both a reduced and FOM velocity vector $\boldsymbol{u}_r$ and $\boldsymbol{u}_h$ are available at equal time instances, which is indeed the case when equal timesteps are used in integrating the FOM and

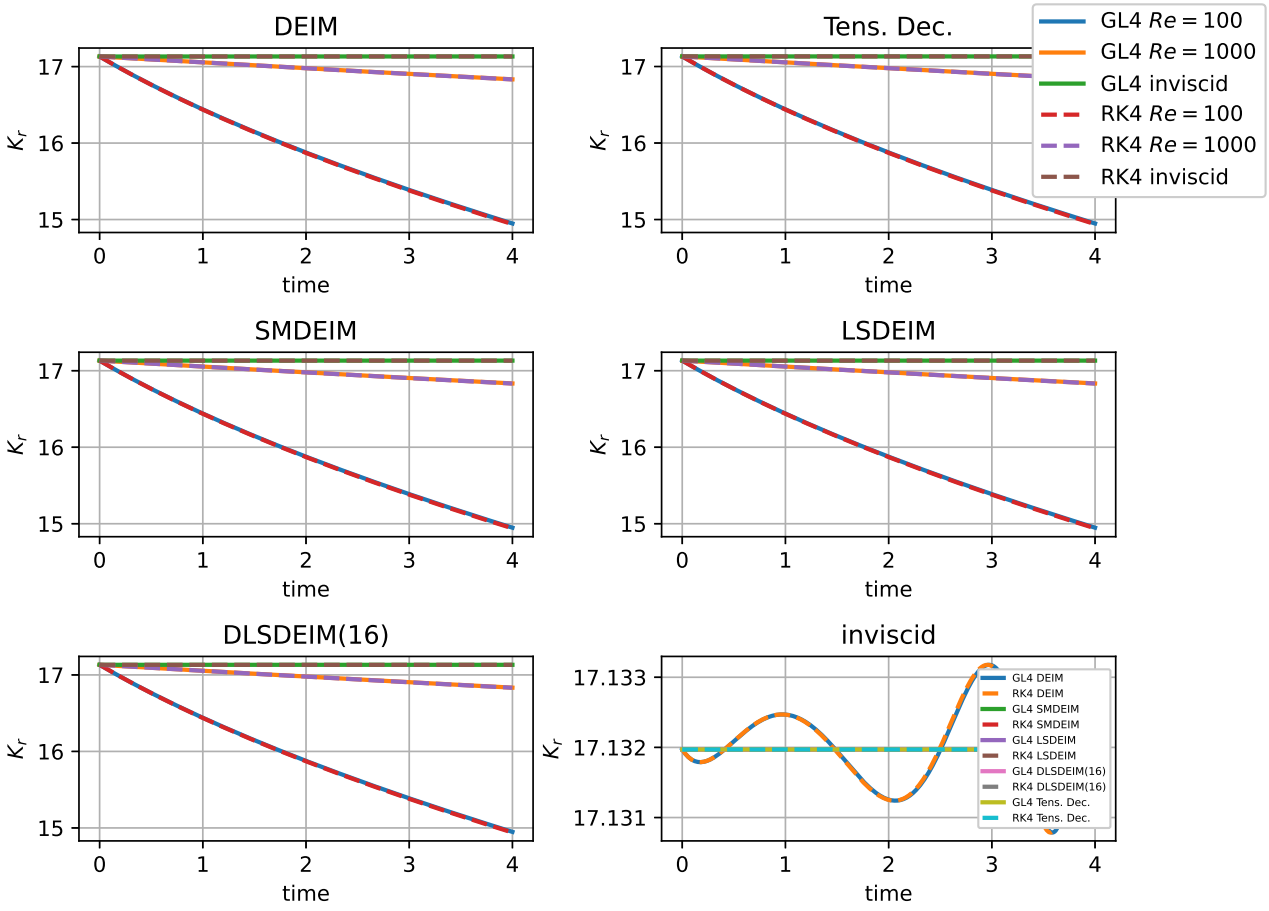


Figure 5.7: Kinetic energy conservation properties for different hyper-reduction methods and Reynolds numbers using 8 POD and 8 DEIM modes.

hROM in time. The second metric is the best approximation error that forms an lower bound for what can be obtained using r POD modes:

$$\epsilon_b(t) = \left|\left|(I - \Phi\Phi^T\Omega_h)\boldsymbol{u}_h(t)\right|\right|_{\Omega_h}.$$

This error simply measures the difference between a FOM snapshot and its reconstruction in the Ω_h-orthogonal POD basis. Essentially, this provides some idea on how far the FOM solution is outside of the reduced space $\mathcal{V}$ by providing the shortest distance between the FOM solution and a point in $\mathcal{V}$ (measured in the Ω_h-norm). Indeed, a hROM cannot provide a solution closer to the FOM snapshot than this. Comparing ϵ_u and ϵ_b will give an indication on how close to optimal the hROM is.

For these experiments the choice of using $\Delta t = 0.01$ for both the hROM as the FOM, saving snapshots every timestep, will be made again. Firstly, the behaviour of ϵ_u with respect to changes in POD and DEIM space dimensions will be analysed. Secondly, the temporal evolution of ϵ_u and ϵ_b will be analysed.

Effect of reduced and DEIM space dimensions

To assess the effect of increasing $r := \dim(\mathcal{V})$ and $m := \dim(\mathcal{M}_d)$ on ϵ_u, several experiments will be performed increasing both r and m independently. This will be done for the SLR on a 128×128 grid. To prevent problems associated to modelling convection-dominated flows using hROMs (instability, inaccuracy, slow Kolmogorov N-width decay) a low Reynolds number of $Re = 100$ will be used. As the error between the usage of GL4 and RK4 has thus far been negligibly small, the RK4 time-integrator will be used for its efficiency. During the experiments

both dimensions will have the ranges $r, m \in \{5, 15, 30\}$ and the truncated bases will capture about 97%, 99.99% and 99.99999% of the energy present in the full POD bases, respectively. The dimensions of the measurement space of the DLSDEIM will simply be taken as twice its DEIM space dimension. This size of the measurement space is considered to emphasize the difference between using and not using oversampling. Finally, the error will be measured at $t = 8$.

The results of this experiment are displayed in Figure 5.8. It is clear that increasing r and m generally has positive effects on the accuracy of the reduced velocity. The SMDEIM, even in this simple case, seems to perform poorly in comparison with the other methods in most cases. Its error also does not necessarily go down as the reduced spaces increase in dimension. It even became unstable for $r = 30, m = 15$, note that the RK4 time-integrator was used which does not offer nonlinear stability in a fully-discrete setting. On average the DLSDEIM(2m) performs best, as it tends to give the smallest error ϵ_u. However, due to the low value of Re this experiment is not very representative of the performance of the methods for practical cases. For this reason the experiment mainly serves to provide an understanding of the effect of increasing r and m on the hROM error obtained from using certain hyper-reduction methods. The error ϵ_u of the DEIM, LSDEIM and DLSDEIM seems to decrease monotonically as r and m are increased, whereas the error of the SMDEIM behaves more erratically as a function of r and m.

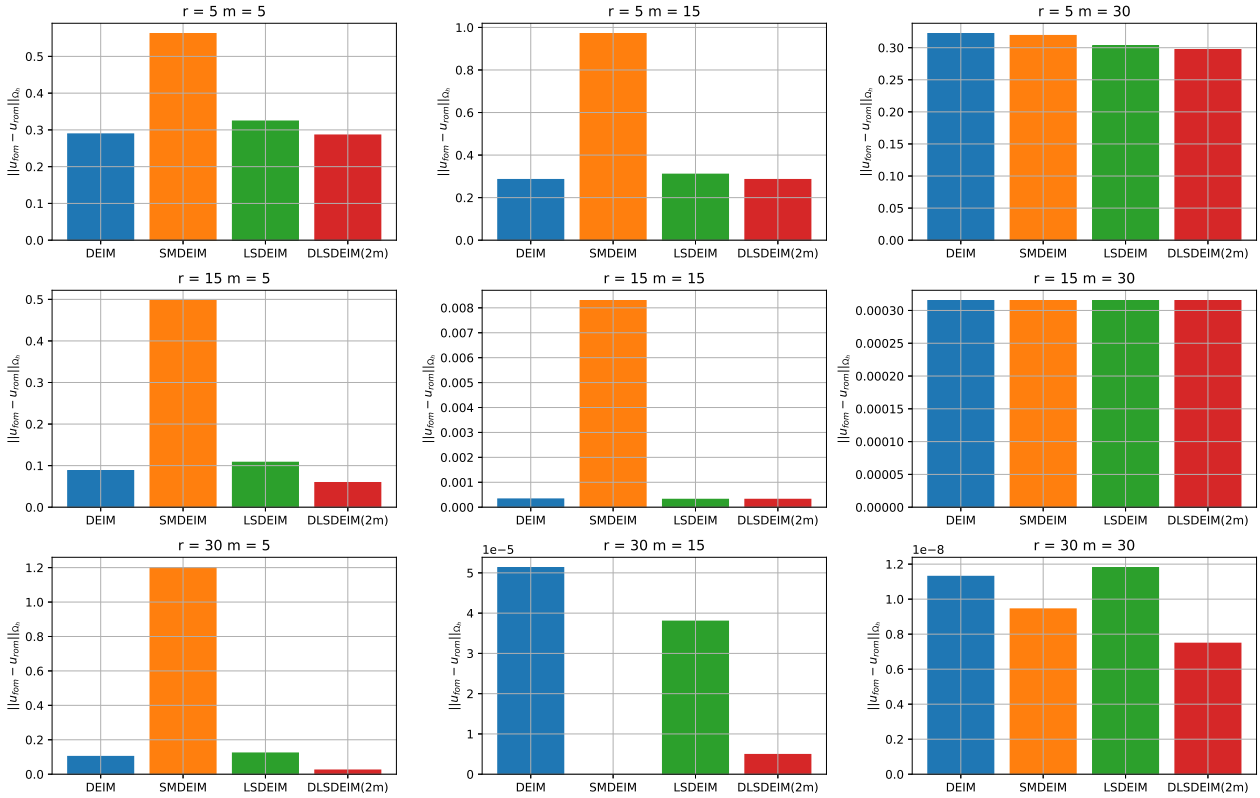


Figure 5.8: Errors between FOM and hROM for different hyper-reduction methods using the Runge-Kutta 4 time-integrator at $t = 8, Re = 100$ on a 128×128 grid. The value of the SMDEIM error at $r = 15, m = 30$ is NaN.

Temporal error evolution

In the following experiment both ϵ_u and ϵ_b will be measured as a function of time. This will give precise insights into what phase of the SLR flow presents the most difficulties for the respective hROMs. The Reynolds number of this experiment will be set to $Re = 1000$ to test the hROMs' limits of performance. The experiment will be carried out on a 256×256 grid with a timestep of $\Delta t = 0.01$, both the RK4 and GL4 time-integrators will be tested

to investigate whether the unconditional stability offered by GL4 for the structure-preserving DEIM variants offers some benefits to the accuracy. All hROMs will have $r = 30$ and $m = 40$, which are appropriate amounts to capture about 99.99% of the energy in the full POD and DEIM bases, respectively. To obtain accurate solutions it was found that the DLSDEIM required a measurement space of 900 measurement points. This is in line with the findings of [26], in which 884 measurement points were used to model a similar flow, namely the Kelvin-Helmholtz instability, on a 200×200 grid. A measurement space of 900 measurement points is more than can be found using Algorithm 2 with the available number of 800 DEIM modes. The extra points are found by sampling 100 non-repeating vector indices (ranging from 0 to $2 \times 256^2 - 1$) from a uniform distribution and adding them to the measurement space. To ensure a set of unique measurement points is obtained, a realization that is already present in the measurement space is discarded and a new random sample is taken until a point is found that is not already in the measurement space. The spatial distribution of the measurement points is depicted in Figure 5.9.

The evolution of ϵ_u and ϵ_b is displayed in Figure 5.10. The SMDEIM can be seen to perform poorly. For GL4 the Newton-Raphson iteration procedure to solve the involved nonlinear systems of equations (4.60) does not converge and for RK4 the simulation becomes unstable. The other methods start to diverge significantly in accuracy around $t = 5$, with the LSDEIM performing poorly, the conventional DEIM performing somewhat acceptably and the DLSDEIM(900) performing well. As depicted in Figure 5.1, at $t = 5$ the roll-up of the shear layer starts to occur. This may explain this divergence in hROM performance. Namely, the roll-up, being a convective phenomenon, is difficult to capture for the hROMs. The difference in performance between the DLSDEIM and the other methods could now be explained by its measurement space. As depicted in Figure 5.9, the DEIM measurement space algorithm managed to identify the problematic vortex edges where a lot of the convection happens and has placed many measurement points there. This seems to have sufficiently increased the capability of the DLSDEIM to capture the nonlinear operator output and hence to stay close to the best approximation error.

The velocity and vorticity fields produced by the hROMs using GL4 are shown in Figure 5.11. The LSDEIM clearly shows traces of POD modes associated to the first four seconds of the SLR still being present in the velocity field. Namely the horizontal lines of low velocity are associated to the shear layer's initial shape. Other spurious modes that are present are highly oscillatory modes in the roll-up region of the flow. These highly oscillatory spurious modes also seem to be excited in the conventional DEIM solution to a much lesser extent. This can be seen in the DEIM's vorticity plot in Figure 5.11, where the contour lines are not very smooth and a wavey pattern is present near the roll-ups[2]. Using the DLSDEIM these modes are not present or to a negligible extent.

5.2.4. Computational Performance

To test the computational performance of the hROMs, measurements will be made of the execution times of their offline and online phases. These measurements will be compared against the FOM execution time. To promote stability, a low Reynolds number of $Re = 100$ will be considered on a 256×256 numerical grid. These measures are predominantly taken for the SMDEIM, which was shown to not be very robust. The simulation will be run until $t = 8$ with a timestep size of $\Delta t = 0.01$. A reduced space dimension of $r = 35$ will be maintained throughout all experiments. This seems unnecessarily large; however, this is to simulate the computational burden of a larger ROM for higher Reynolds numbers without actually having to simulate higher Reynolds numbers. The offline phase measurements will be split in time spent on precomputing operators and time spent on performing the SVD. This is to prevent implementation specific aspects of the SVD giving a skewed perspective on the time spent on the offline phase. The online phase will be considered to be the time spent integrating the system in time. Implementation details like setting up a mesh for the FOM will be excluded from the timing. Also the time for saving snapshot data by the FOM integrator will be excluded. The DEIM spaces considered will be $m \in \{5, 10, 20, 40, 80\}$ and the DLSDEIM will have a

[2]When the digital format of this document is used, consider zooming on the DEIM's vorticity panel.

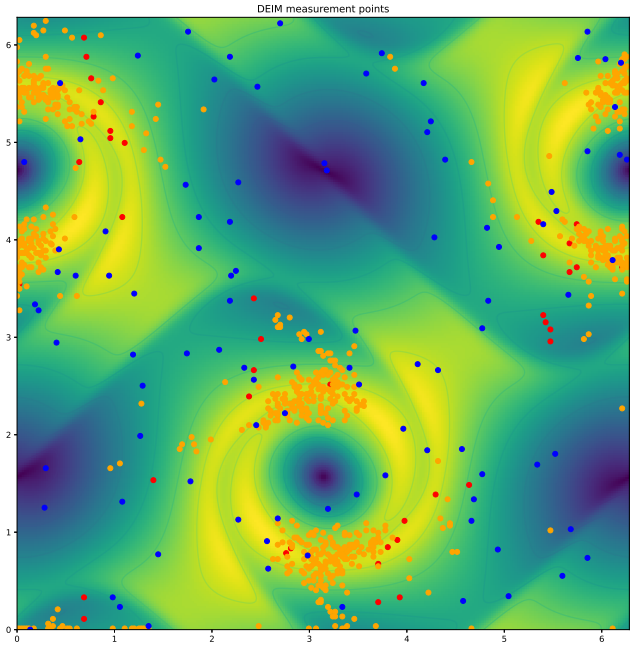


Figure 5.9: DEIM measurement points on a 256×256 grid for $Re = 1000$ (Red: first 40 points, Orange: following 760 points, Blue: 100 random extra points).

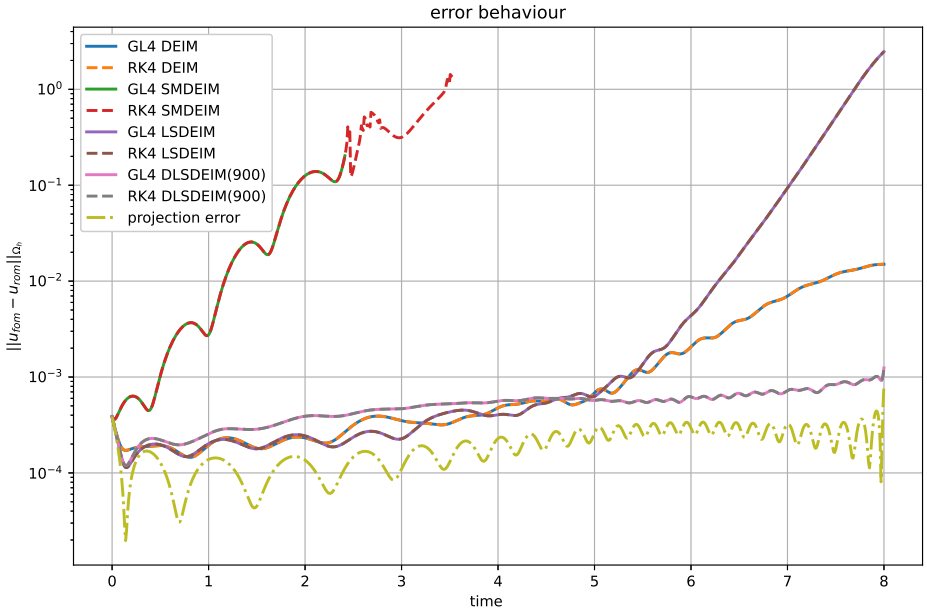


Figure 5.10: Evolution of hROM errors for different hyper-reduction methods ($r = 30$, $m = 40$) and time-integrators on a 256×256 grid for $Re = 1000$ and the best approximation error.

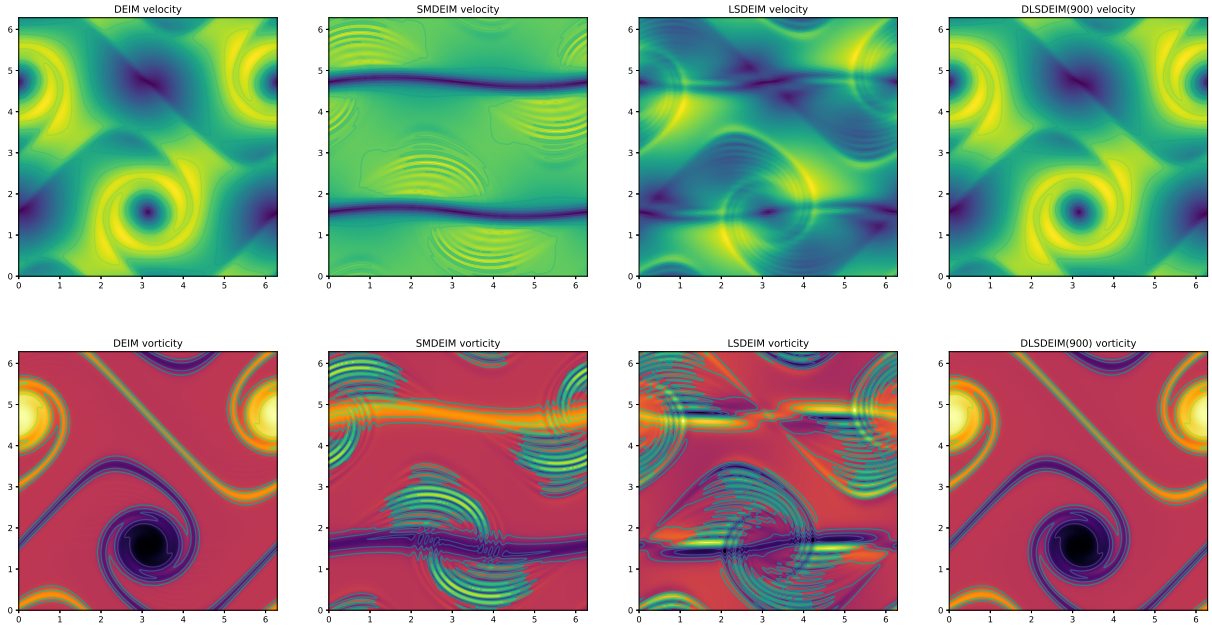


Figure 5.11: hROM velocity and vorticity fields for different hyper-reduction methods ($r = 30$, $m = 40$) on a 256×256 grid for $Re = 1000$ using the Gauss-Legendre 4 time-integrator.

measurement space dimension of $2m$.

The results for RK4 are provided in Figure 5.12. A speedup of the hROM online phase compared to the time integration phase of the FOM of several orders of magnitude can be observed. Here the more practical cases of $m \in \{20, 40, 80\}$ have speed ups ranging from approximately $100\times$ to $500\times$. Recalling that eight seconds of physical fluid flow time were simulated, the hROMs provided significantly faster than real-time simulations. Precomputing the operators happens on equivalent timescales as the full online phase for all DEIM algorithms for the ranges of m tested. However the DLSDEIM's precomputation phase will overtake the cost of its online phase when the measurement space dimensions get sufficiently large due to the large number of linear solve steps required in Algorithm 2 in this case. Furthermore, all hROM related computational scalings appear linear in m. This is in contrast to the theoretical estimates in chapter 4, which predict $\mathcal{O}(m^2)$ scaling due to the LU-Solves required to determine the DEIM coordinates $\boldsymbol{c}$. This typically happens when the dimensions of the problem are not large or when constants multiplying the $\mathcal{O}(m)$ computation times present in the DEIM algorithms outweigh the $\mathcal{O}(m^2)$ operations. The DLSDEIM can be observed to be the slowest method as may be explained by the fact that the costs of the $\mathcal{O}(m)$ operations dominate the higher order costs in the cases tested. Since any sampling operation is an $\mathcal{O}(m)$ operation, these types of operation will clearly be more expensive for the DLSDEIM. This is a result of its measurement space being of size $2m$ and thus twice the amount of samples compared to the other methods will need to be taken. The SMDEIM and the conventional DEIM are the fastest, however the SMDEIM produced NaN results for several of the smaller m values. The costs of the SVD are relatively large compared to the rest of the hROM operations, this is an implementation detail of Armadillo and outside the control of this research. If it is desired to accelerate this step one may consider the use of the method of snapshots [94], however as it is not very relevant to the research in this thesis this will not be done here.

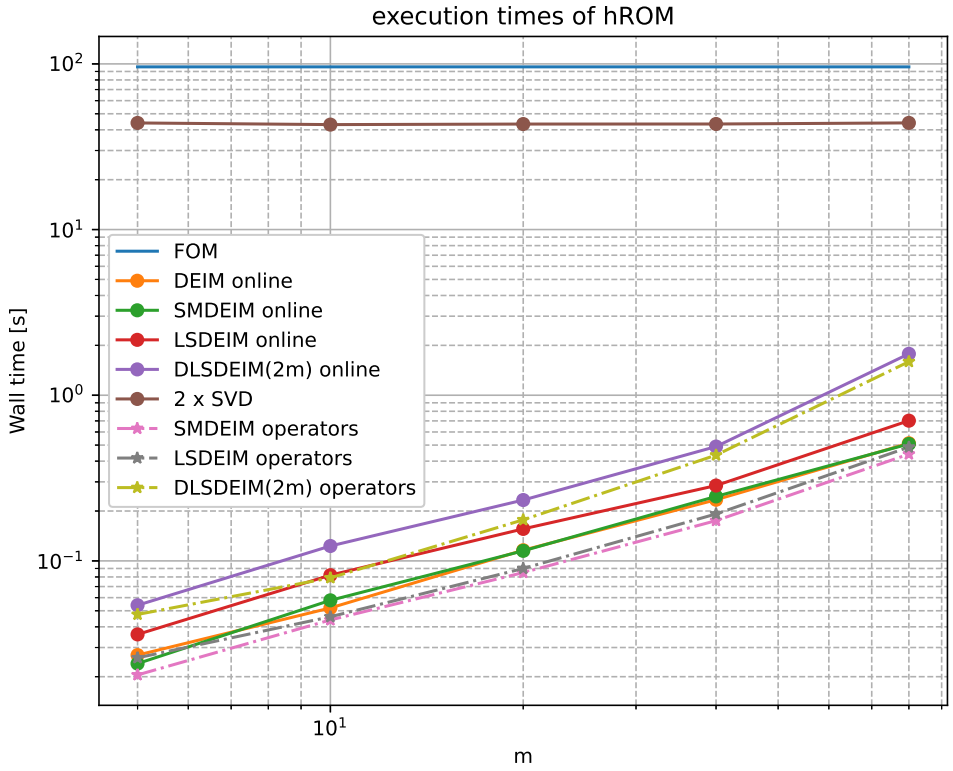


Figure 5.12: Execution times of the FOM and the online phase of several hROMs ($r = 35$) using RK4 on a 256×256 grid for $Re = 100$ and $t = 8$ with $\Delta t = 0.01$, the SVDs for the POD and DEIM and the precomputation of the hyper-reduction operators.

5.2.5. Discussion

The experiments in this section were intended to select the overall best-performing hyper-reduction method. This method will be used in the 2DT case in the following section. Determining the overal performance of the proposed methods has been done by considering the structure-preserving capabilities, accuracy and computation times for the relatively complicated SLR flow using either $Re = 1000$ or $Re = 100$. It was confirmed that the structure-preserving DEIM algorithms managed to conserve both reduced total momentum, as does the conventional DEIM, using the altered POD basis of [86] and also reduced total kinetic energy using energy-conserving Runge-Kutta methods. The observation made in [86] that high-order explicit Runge-Kutta methods in a lot of practical cases are also sufficient to keep errors in energy conservation negligible was confirmed for the proposed structure-preserving hyper-reduction methods as well. Analysing an inviscid case without strong convective effects being present (yet) it was shown that the kinetic energy of the reduced velocity field produced by the conventional DEIM was already oscillating with increasing amplitude. This indicates that for high Re flows where much spatial transport occurs, the conventional DEIM cannot be expected to remain stable, providing an advantage to using structure-preserving methods.

In all accuracy tests the SMDEIM failed to perform well. Independently increasing the reduced and DEIM space dimensions r and m was observed to have a positive effect on the accuracy of the solutions produced by the DEIM, LSDEIM and DLSDEIM. To a lesser extent, this also appeared to be the case for the SMDEIM, due to the erratic behaviour shown by the error as a function of r and m. Applying the methods to a convection-dominated flow, the SLR at $Re = 1000$, it was observed that when the convection phenomena started to dominate the flow field, the LSDEIM, DEIM and DLSDEIM started to diverge in accuracy. After this moment the DLSDEIM performed the best. This may be attributed to its increased measure-

ment space dimension, where the additional points found by Algorithm 2 managed to identify problematic convection dominated regions of the flow. This builds quite a strong case for the use of oversampling in the hyper-reduction of high Re flow simulations, which is an observation shared by [107, 59].

In the experiments for computational performance it was found that the operations scaling linearly in m dominated the computational cost. This resulted in the DLSDEIM performing the worst, due to its larger measurement space. However the differences in execution times are still negligible to the cost of the FOM. Moreover, all hyper-reduction methods were faster than real-time.

To conclude, although the DLSDEIM was slower than the other methods in terms of execution times, its flexibility in increasing the measurement space dimensions and the effect it had on its capability to capture convective phenomena compared to the other methods make it the best performing structure-preserving hyper-reduction method. Especially the DLSDEIM's performance for convection-dominated flows is relevant, as most flows of interest to engineering purposes often have high Reynolds numbers and a lot of convected spatial details. For this reason, the DLSDEIM will be used in the numerical experiments of the following section.

5.3. Results: Freely Decaying Two-Dimensional Turbulence

In this section results will be described of numerical experiments carried out on the DLSDEIM using the 2DT flow. Besides testing if the hROM using the DLSDEIM is capable of reproducing the complex spatiotemporal features of the 2DT flow, emphasis will be laid on if the correct two-dimensional turbulent physics are maintained in a reduced setting. In this section the PID will be used in the construction of the reduced spaces $\mathcal{V}$ and $\mathcal{M}_d$ in an attempt to increase the system's Kolmogorov N-width decay. As the actual Kolmogorov N-width decay is difficult to determine, the effect of the PID will be measured by comparing the decay of the singular values of the full snapshot matrices and the individual partitioned snapshot matrices. The replication of two-dimensional turbulent physics will be tested by considering the temporal evolution of reduced total kinetic energy and reduced total enstrophy and also by comparing the angle averaged energy spectra of the hROM and FOM velocity fields.

To make sure the FOM simulation is well resolved, simulation parameters will be chosen based on relevant non-dimensional numbers. In this thesis simulation parameters will be sought that satisfy the following conditions:

$$\text{CFL} = \frac{\sqrt{u_{\max}^2 + v_{\max}^2}\Delta t}{\Delta x} \leq 1 \tag{5.6}$$

$$Re_h = \frac{\sqrt{u_{\max}^2 + v_{\max}^2}\Delta x}{\nu} = \mathcal{O}(1) \tag{5.7}$$

$$\text{Diff} = \frac{\nu \Delta t}{\Delta x^2} \leq \frac{1}{2}. \tag{5.8}$$

Condition (5.6) is the well-known CFL-number which makes sure information propagation through the grid is well-resolved, condition (5.7) is the grid Reynolds number and condition (5.8) is the stability limit of diffusion based on diffusive time scales being well-resolved. It may be checked that a combination of $Re = 1000$, $\Delta x = 1/1024$ and $\Delta t = 2 \times 10^{-4}$ satisfy all conditions. All experiments will thus be performed on a 1024×1024 numerical grid using these parameters (FOM and hROM). Using the theory described in [35], the two-dimensional turbulent length- and timescales, scaling as $\mathcal{O}(L_c Re^{-\frac{1}{2}})$ and $\mathcal{O}(L_c/\sqrt{u_{\max}^2 + v_{\max}^2})$ respectively, are also resolved. The realization of the stochastic initial conditions used for all experiments in this thesis is displayed in terms of absolute velocity and vorticity in Figure 5.13 and Figure 5.14 respectively. All experiments will be carried out until $t = 3.5$ using the RK4 time-integrator (FOM and hROM) for computational efficiency. Saving snapshots every timestep for this test case is not feasible given the simple workstation used to perform all calculation in this thesis. Hence, the simulation will be sampled such that the Nyquist-Shannon criterion is met [3, 4]; this criterion is satisfied by sampling every $\Delta t_s = 0.005$. This will result in two data sets (solution snapshots and convection operator snapshots) of $2 \times 1024^2 \times \frac{3.5}{0.005} \times 8$ Bytes ≈ 11.7GB of data each when using double precision.

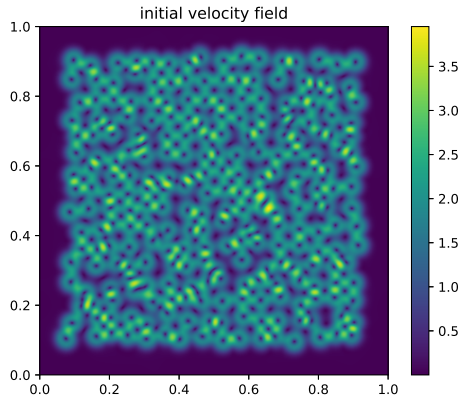


Figure 5.13: 2DT absolute magnitude of initial velocity field.

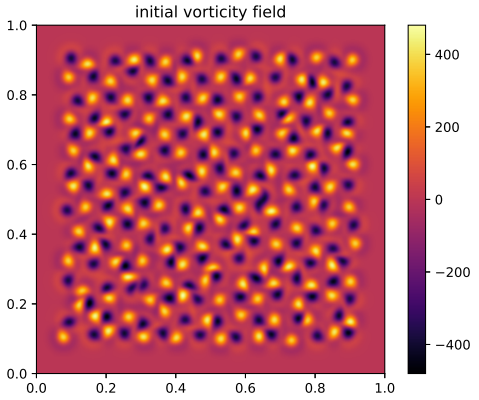


Figure 5.14: 2DT initial vorticity field.

5.3.1. The Need for Temporal Localization

As illustrated in [4] the PID is a useful tool for increasing the Kolmogorov N-width decay and to capture small timescales that would otherwise be averaged out in ordinary POD. Now some experiments will be performed to demonstrate that the application of the PID can also be extended from the construction of $\mathcal{V}$ to the construction of $\mathcal{M}_d$ for use in hyper-reduced models.

Singular value decay

The first experiment will be to compare the singular value decay of both X and Ξ to those of individual submatrices X_i and Ξ_i of the respective snapshot matrices. Before this can be done a partition must be made. For simplicity this will be done in equal parts. The number of intervals is chosen such that each interval has sufficient snapshots to work with, yet are still minimal in duration to optimally resolve local timescales. To this end intervals of length $0.5s$ are chosen, which means every interval has 100 snapshots. This division will be maintained throughout all other experiments using the 2DT flow throughout this thesis.

The singular values of X, Ξ, all X_i and all Ξ_i are shown in Figure 5.15 and Figure 5.16 for the solution snapshots and operator snapshots, respectively. Every set of singular values is normalized against the largest value in the set. The number of singular values of X and Ξ is cut off at approximately 400 as the singular values with higher indices were valued NaN[3] (Not a number). It can clearly be seen that the singular values associated to the individual intervals decay significantly faster than those of the full snapshot matrices for both the solution snapshots and the operator snapshots. This indicates that the individual intervals are more amenable to dimensionality reduction using the POD than the full snapshot matrices.

Interestingly, it can also be seen that later intervals have a faster decay of singular values than earlier intervals. This is reflective of the inverse energy cascade as a result of which large scale coherent structures dominate the flow after sufficiently long times. Flows containing large scale coherent structures are well-known to have fast Kolmogorov N-width decay [47, 94].

FOM and PID-DLSDEIM hROM comparison

Now a simulation will be done using the PID-DLSDEIM hROM and the RK4 time-integrator as it has shown to lead to negligible energy errors thus far. The reduced, DEIM and measurement space dimensions are tabulated in Table 5.1. The resulting FOM and PID-DLSDEIM hROM velocity fields are provided in Figure 5.17a and the associated vorticity fields are provided in Figure 5.17b, both for several instances in time. Both the velocity and vorticity fields calculated by the PID-DLSDEIM hROM and FOM are almost identical. Additionally, an inverse energy

[3] This is a result of the fact that the singular values have been determined from the eigenvalues of $X^T X$ and $\Xi^T \Xi$ by taking their square-root; however, many eigenvalues had values of approximately $-|\epsilon|$, where ϵ denotes machine-precision, resulting in NaN-values.

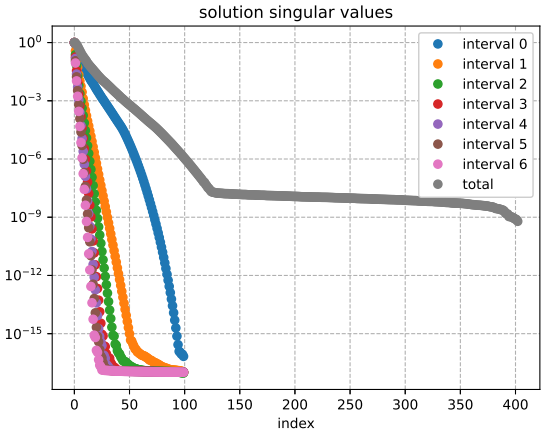


Figure 5.15: 2DT singular values for several intervals of duration 0.5 labeled in chronological order and the total solution snapshot matrix X.

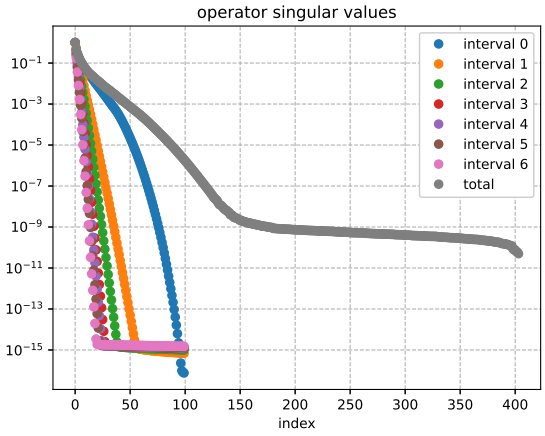


Figure 5.16: 2DT singular values for several intervals of duration 0.5 labeled in chronological order and the total operator snapshot matrix Ξ.

cascade has clearly taken place in both the FOM as the PID-DLSDEIM hROM, as both produce fields with increasing presence of large scale coherent structures.

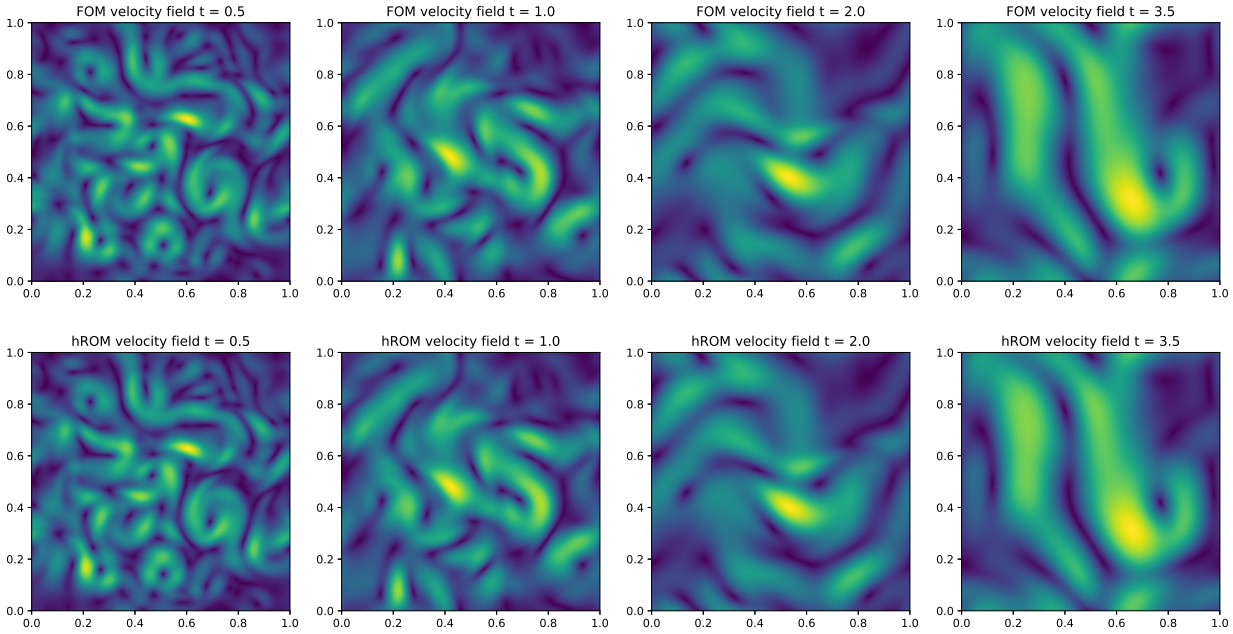
interval	0	1	2	3	4	5	6
r	37	19	12	12	12	12	12
m	37	19	14	14	14	14	14
m_p	300	100	100	100	100	100	100

Table 5.1: Table of reduce space dimensions used for the individual interval of the PID-DLSDEIM hROM.

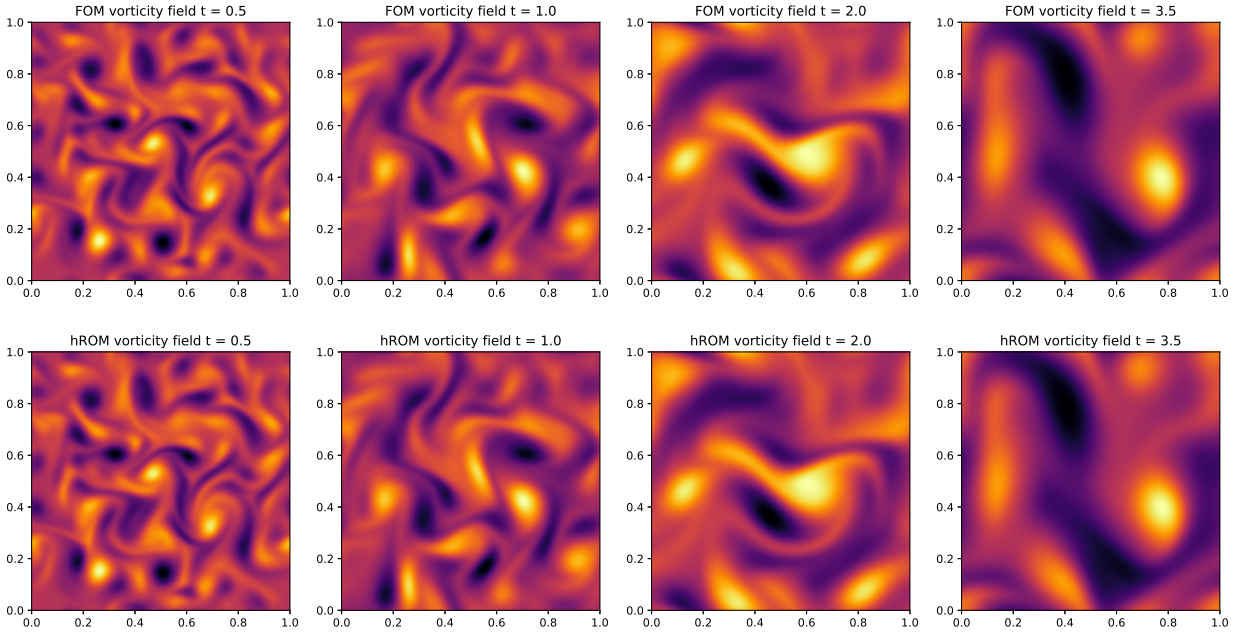
5.3.2. Kinetic Energy and Enstrophy

A way to confirm whether the conservation of kinetic energy and the enstrophy cascade are captured by the PID-DLSDEIM hROM is to plot the temporal evolution of $\mathcal{K}_r$ and $\mathcal{E}_r$ for increasing Reynolds numbers. Following the theory of two-dimensional turbulence, the reduced total kinetic energy curve should then flatten off as Re is increased, whereas the reduced total enstrophy should be continuously decreasing after some initial organisation phase. However, with the current second order implementation of the FOM this is computationally infeasible due to the sizes of the numerical grids that would be necessary to resolve e.g. a $Re = 1 \times 10^6$ flow. To still obtain an idea of whether the PID-DLSDEIM hROM captures the enstrophy cascade and kinetic energy conservation, the results on the 2DT flow obtained in the numerical experiment performed in this thesis will be compared to simulation results of other authors. More specifically, it will be analysed if the evolution of reduced total kinetic energy and reduced total enstrophy as calculated by the proposed PID-DLSDEIM hROM is in line with the results for larger Reynolds numbers provided by [91]. Results for the currently implemented FOM will also be provided for the experiment performed in this thesis such that the FOM and PID-DLSDEIM hROM can be compared. The research of [91] has been chosen to compare the current results to. This is both because [91] also uses a velocity-pressure formulation of the incompressible Navier-Stokes equations and that the initial conditions in their simulations are similar to those used in this thesis. Most other authors [4, 83] use vorticity-streamfunction formulations and have initial conditions formed from an initial angle averaged energy spectrum combined with random phases.

In Figure 5.18a and Figure 5.18b the results of the experiment are provided in terms of reduced total kinetic energy and reduced total enstrophy evolution. The PID-DLSDEIM hROM results follow the trend of increased dissipation of both quantities as the Reynolds number is decreased. In accordance with the theory, the dissipation rate of both kinetic energy and enstrophy is larger for 2DT with $Re = 1000$ than the higher Re cases. Consistently with the



(a) 2DT velocity fields using the FOM and PID-DLSDEIM hROM at different instances in time ($t = 0.5, 1.0, 2.0, 3.5$) for $Re = 1000$ on a 1024×1024 grid.



(b) 2DT vorticity fields using the FOM and PID-DLSDEIM hROM at different instances in time ($t = 0.5, 1.0, 2.0, 3.5$) for $Re = 1000$ on a 1024×1024 grid.

Figure 5.17

case of $Re = 2 \times 10^4$ by [91] there is also no presence of an initial transition period for the $Re = 1000$ case due to the large dissipation in these cases. The curves for both total kinetic energy and total enstrophy overlap well between the PID-DLSDEIM hROM and the FOM, where the PID-DLSDEIM hROM seems to have dissipated marginally less kinetic energy than the FOM nearing the end of the simulation.

It can also be seen in Figure 5.18a that at the interfaces between different PID intervals there is no significant error in the total kinetic energy. This provides some evidence that in practise a non-structure-preserving transition mapping, like the one obtained from condition (4.65), may be sufficiently accurate to prevent significant energy errors between intervals. This is similar to how RK4 in some practical scenarios also does not cause significant energy errors, even though it is not part of the family of energy-conserving Runge-Kutta methods.

5.3.3. The Energy Spectrum

The angle averaged energy spectrum $E(k,t)$ is a useful tool in the analysis of two-dimensional turbulence. In this thesis a discrete version $E_k(t)$ of this spectrum will be derived to analyse the discrete velocity fields produced in numerical experiments. This spectrum is calculated on a 1024×1024 uniform grid using the following formula:

$$E_k(t) = \sum_{k \leq \sqrt{m^2+n^2} < k+1} 4\pi^2 \cdot \frac{1}{2}\left(m^2+n^2\right)|\hat{\psi}_{m,n}(t)|^2,$$

where $k \in \{0, 1, ..., \lfloor k_{\max} \rfloor\}$, $k_{\max} = \sqrt{2}\cdot 512$ and $m, n \in \{-512, -511, ..., 511\}$ and where $\hat{\psi}_{m,n}(t)$ is the Fourier coefficient with indices m and n of the Fourier series for the streamfunction:

$$\psi(\boldsymbol{x},t) \approx \sum_{m,n=-512}^{511} \hat{\psi}_{m,n}(t) e^{i\left(m\frac{2\pi}{L}x + n\frac{2\pi}{L}y\right)}.$$

The angle averaged energy spectrum satisfies:

$$\widetilde{K}_h(t) = \sum_{k \in \{0,1,...,\lfloor k_{\max} \rfloor\}} E_k(t),$$

where:

$$\widetilde{K}_h(t) = \frac{1}{2}\left|\left|\widetilde{\boldsymbol{u}}(\boldsymbol{x},t)\right|\right|^2_{L^2(\Omega)},$$

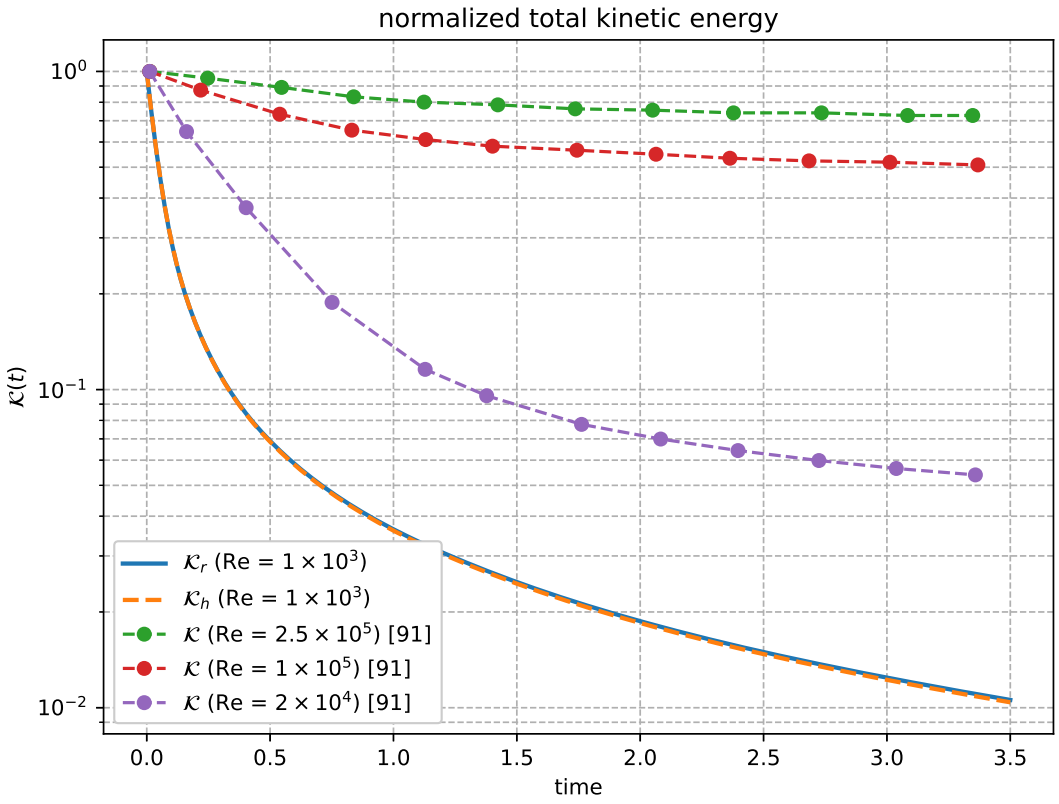
and $\widetilde{\boldsymbol{u}}(\boldsymbol{x},t) : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}^d$ is the *continuous* velocity field found from representing the *discrete* velocity field $\boldsymbol{u}_h(t) : \mathbb{R}^+ \to \mathbb{R}^N$ in terms of a Fourier series with 1024^2 two-dimensional Fourier modes. Classical theory on two-dimensional turbulence [35] states that in the inertial region of the turbulence $E_k(t)$ should scale as $\mathcal{O}(k^{-3})$ in the inviscid limit.

Comparing energy spectra

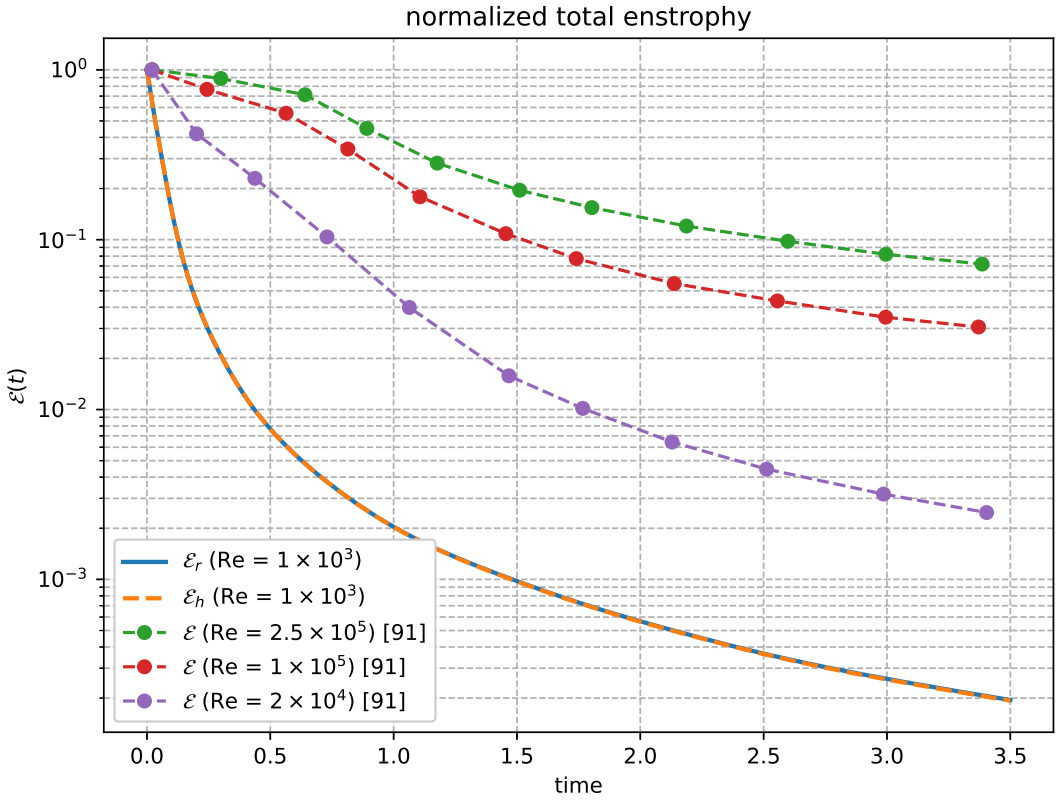
In the following experiment the discrete angle averaged energy spectrum will be determined for both the FOM and the PID-DLSDEIM hROM for different instances in time. It will be analysed if the PID-DLSDEIM hROM can reproduce the correct turbulent physics across the full range of physical length scales by comparing the discrete spectrum of the PID-DLSDEIM hROM to that of the FOM. The following time instances will be considered $t \in \{0.5, 1.0, 2.0, 3.5\}$.

The results of the experiment are displayed in Figure 5.19. It can be seen that the spectra of the FOM and the PID-DLSDEIM hROM generally overlap well for larger length scales. However, for the smaller length scales present in the flow, the PID-DLSDEIM hROM overestimates the energy consistently at every time instance. It should be noted that these errors carry minimal amounts of energy, but they may still be important for correct energy dissipation. Two reasons for this behaviour are hypothesized:

1. ROMs of fluid flow without closure models have been hypothesized to not accurately capture dissipation [5, 13] as often truncated POD bases contain only slowly varying high energy structures, whereas quickly-varying low-energy structures are typically responsible for dissipation of energy. This would cause energy dissipation to be underestimated and hence some remaining energy in the system.



(a) 2DT total kinetic energy evolution using the FOM and PID-DLSDEIM hROM for $t = 3.5$, $Re = 1000$ on a 1024×1024 grid as compared to higher Re results of [91]. All results are normalized against their respective initial values.



(b) 2DT total enstrophy evolution using the FOM and PID-DLSDEIM hROM for $t = 3.5$, $Re = 1000$ on a 1024×1024 grid as compared to higher Re results of [91]. All results are normalized against their respective initial values.

Figure 5.18

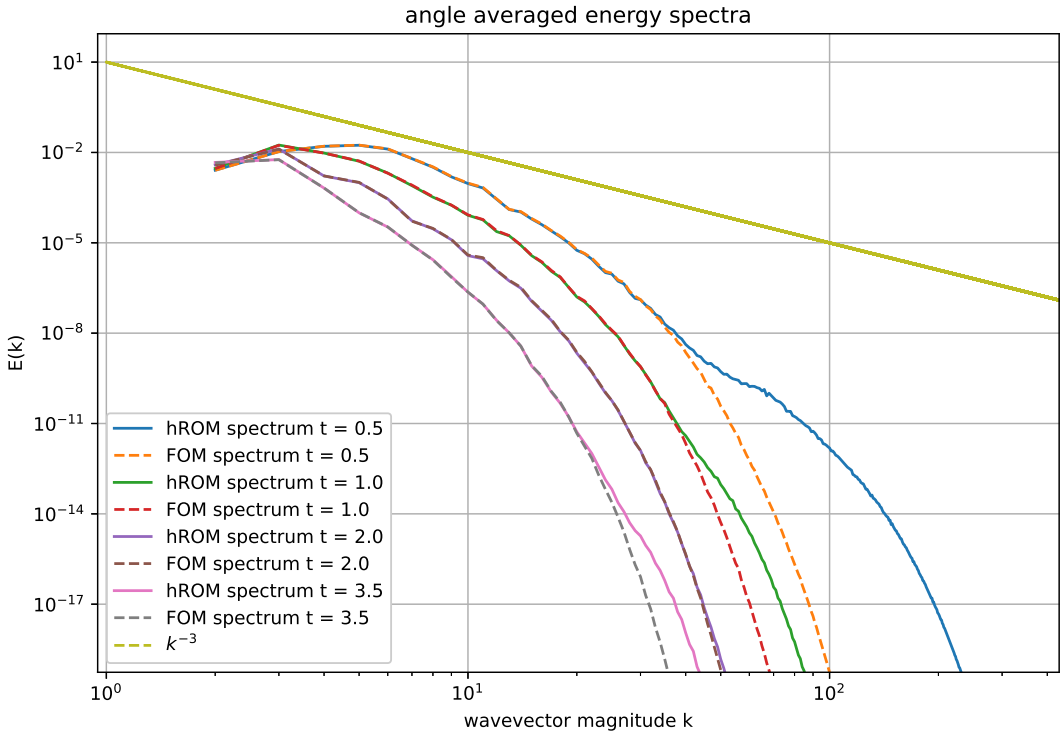


Figure 5.19: 2DT discrete angle averaged energy spectrum $E_k(t)$ using the FOM and PID-DLSDEIM hROM at several instances in time ($t = 0.5, 1.0, 2.0, 3.5$) for $Re = 1000$ on a 1024×1024 grid.

2. Oscillatory spurious modes as in Figure 5.11 are present to a minimal extent causing small scale structures in the reduced reconstruction of the flow to have too much energy.

Like already observed in Figure 5.17a and Figure 5.17b, an inverse energy cascade has clearly taken place as the energy present in smaller length scales decays quickly over time whereas the larger length scales keep their energy for longer times. The energy spectra of both the FOM and PID-DLSDEIM hROM do not show evident scaling with $\mathcal{O}(k^{-3})$, in contrast to the results for $Re = 1000$ obtained in [83].

5.3.4. Discussion

In the previous subsection the potential of the most accurate structure-preserving hyper-reduction method (DLSDEIM) to reduce the dimensionality of very convection-dominated flows has been tested using the 2DT flow. Specific attention was paid to the correct reproduction of the turbulent physics that take place in this flow. Furthermore, it was tested whether the PID can aid in the construction of a reduced space $\mathcal{V}$ that accurately captures the solution manifold of the FOM in low-dimensional fashion. Even more importantly, it was tested whether this notion also generalizes to the construction of DEIM spaces.

In the first experiment the decay of the singular values of several PID intervals was compared to their snapshot matrix of origin. It was observed that the singular values of the intervals decayed significantly faster than those of the associated full snapshot matrix for both the solution as the operator snapshots. This implies that, using the PID, the FOM solution manifold and the manifold on which the FOM operator evolves can be captured in low-dimensional ROMs more easily. This was tested by running the simulation for the PID-DLSDEIM hROM and comparing results with the FOM. A nearly identical match was obtained.

The second and third experiment focused on whether the turbulent physics could be accurately captured in low-dimensional reduced spaces constructed using the PID. This was tested by considering the decay of both reduced total kinetic energy and reduced total enstrophy. As it was not feasible to run well-resolved high-fidelity simulations to gather data for high

Reynolds numbers only the a simulation of $Re = 1000$ was run. It was then checked whether the results of this simulation were in line with results from high-fidelity simulations of higher Re from the literature [91]. This seemed to be the case; however, strong conclusions should not be made on the capability of the PID-DLSDEIM method to accurately capture such high Re flows until a structure-preserving FOM is constructed that can efficiently handle larger values of Re, providing a starting point for model reduction. Finally, discrete angle averaged energy spectra of both the FOM and PID-DLSDEIM hROM were considered. It was observed that smaller length scales in the flow contained too much energy. Two possible causes based on the literature and previous observations on the SLR were given. As noted these errors carry very little energy and as seen in Figure 5.17a and Figure 5.17b have not resulted in significant effects on the accuracy of the PID-DLSDEIM hROM for the case considered.

6

Conclusion

In this thesis three new structure-preserving hyper-reduction methods have been proposed that are based on the discrete empirical interpolation method (DEIM) [29]. The first method has been named the least square discrete empirical interpolation method (LSDEIM) and is characterized by relaxing the condition of exact correspondence between the FOM nonlinearity and the DEIM approximation in the measurement space. Instead, a residual minimization problem is solved at the operator level which is constrained such that the DEIM approximation mimics the energy-conservation property of the FOM's convection operator. Due to the convexity of the minimization problem and the feasible set of the associated constraint, it could be shown that unique solutions exist for the LSDEIM making it particularly robust. The second method has been named the sherman-morrisson discrete empirical interpolation method (SMDEIM). The SMDEIM distinguished itself by decreasing the dimensions of the measurement space by one such that an energy conservation condition could be imposed on the DEIM approximation. This resulted in a linear system that depended on newly determined generalized coordinates in the reduced space. The problem of having to solve this system every timestep was solved by using the efficient Sherman-Morrison inverse formula. The last method is the decoupled least-squares discrete empirical interpolation method (DLSDEIM), which constituted a generalization of the LSDEIM to decoupled DEIM and measurement space dimensions. The DLSDEIM satisfied similar existence and uniqueness properties as those satisfied by the LSDEIM. Both the LSDEIM and DLSDEIM were shown to be consistent in a similar sense as the conventional DEIM. Namely, given a sufficiently accurate DEIM space, energy-conserving convection operator outputs could be exactly reconstructed. Furthermore, the potential to perform oversampling made the DLSDEIM especially flexible and suitable for problems with a large range of spatio-temporal features.

To bypass the slow Kolmogorov N-width decay of convection dominated turbulent flows and to allow the reduced spaces to properly resolve short timescales in the flow, the use of the PID [49, 18] for both the construction of the reduced space associated to the solution and also the DEIM space has been applied in this thesis. Although reduced total momentum and reduced total kinetic energy were conserved within the intervals of the PID, these conservation properties were lost at interfaces between intervals. Some novel preliminary suggestions have been provided to solve this problem, but these remain to be analysed further and to be put into practise.

Numerical experiments were performed to test the structure-preservation properties, the accuracy, robustness and efficiency of the proposed structure-preserving hyper-reduction methods for increasingly more convection-dominated flows. The first test case considered was the shear layer roll-up (SLR). This test case was intended to identify the best performing method among the three newly proposed hyper-reduction methods in terms of the aforementioned criteria of accuracy, efficiency and robustness. The test case was also used to confirm the structure-preserving properties. It was shown that in the inviscid case all structure-preserving hyper-reduction methods conserved reduced total kinetic energy, whereas the conventional DEIM showed signs of increasingly unstable kinetic energy evolution. In the viscous case all methods

showed dissipation of kinetic energy as expected. Analysing the error behaviour as a function of reduced space dimensions for low Reynolds number flows the SMDEIM already showed poor accuracy. The other hyper-reduction methods exhibited monotonic decreases in errors as reduced space dimensions were increased. Increasing the Reynolds number to convection-dominated flows it was shown that the DLSDEIM was the most accurate and only slightly more expensive than the other hyper-reduction methods. It was therefore used in experiments using the second test case; two-dimensional isotropic decaying turbulence (2DT).

The aim of the 2DT test case was to test if the DLSDEIM in combination with the PID was capable of both simulating a very convection-dominated flow accurately and reproducing several important features of two-dimensional turbulence physics. It was shown that the PID in construction of both the reduced and DEIM spaces was a useful tool to resolve the short time scales of 2DT and to increase the Kolmogorov N-width decay. Nearly identical velocity and vorticity reconstructions were obtained from the PID-DLSDEIM hROM in comparison to the FOM results. Moreover, a clear inverse energy cascade was exhibited by the hROM. Analysing the decay of reduced total kinetic energy and reduced total enstrophy, the PID-DLSDEIM hROM showed to be in line with the system's physical behaviour at higher Reynolds numbers obtained from simulations performed in [91]. Furthermore, the non-structure-preserving interface condition of the PID did not result in significant energy errors at interfaces between different PID intervals. Finally, the discrete angle averaged energy spectra of the velocity fields predicted by the PID-DLSDEIM hROM and the FOM were compared. Good correspondence was observed for large spatial structures but the smaller low-energy structures showed some disparity. This disparity did not cause notable differences in the velocity field as it only happened at very low energies. Some hypotheses were provided as to the reason of this behaviour, attributing it to either lack of dissipation due to the absence of higher-order modes or the negligible presence of oscillatory spurious modes.

Other contributions of this thesis were that using the momentum conserving POD basis proposed in [86], reduced total momentum could also be conserved by the DEIM approximations. This was a result of the eigenvalue problem underlying the SVD in the construction of the basis of the DEIM space. Moreover, it was demonstrated that the energy-conserving Runge-Kutta methods could be used in combination with structure-preserving DEIM variants to obtain nonlinearly stable fully-discrete hROMs as could be done with the FOM and ROM on which the hROM is built [86]. The observation made in [86] that the use of high-order explicit Runge-Kutta methods could result in negligible energy errors for a structure-preserving semi-discrete ROM has been extended in this thesis to the structure-preserving hROMs. This provides a cheap alternative to the implicit energy-conserving Runge-Kutta methods.

From all of the above reasoning it can be concluded that the PID-DLSDEIM is a useful hyper-reduction method for larger convection dominated systems where exact methods are prohibitively expensive or not available. It has shown to be the most accurate and robust method and was comparable in terms of computational cost to the cheaper methods proposed in thesis. It could also, to a satisfying extent, reproduce much of the physical behaviour shown in two-dimensional turbulent flow providing some confidence that this can be extended to three-dimensional turbulent cases. The research goal as formulated in the introduction of this thesis seems to be achieved, at least for the cases considered in chapter 5. Of course, it remains to be seen how well the PID-DLSDEIM will perform for industrial cases in three-dimensions as the transition from 2DT to these flows is not trivial.

There are many topics regarding structure-preserving hyper-reduction left for future research. Some of such topics are provided here:

1. Although, not the focus of this thesis, hyper-reduction of Hamiltonian systems is an important field of model reduction. The methods proposed here form a cheaper alternative to those suggested in [63]. Analysing the performance of the presently proposed methods in a Hamiltonian setting would form an interesting topic of research.
2. Structure-preserving interface conditions for the PID are a topic of research that this thesis has not delved into. Though, for highly convection-dominated systems where the PID could form an attractive solution it would be beneficial to have the certainty of nonlinear stability offered by structure-preservation. Therefore, structure-preserving interface conditions are suggested as possible future research topic.

3. This thesis has dealt primarily with periodic boundary conditions to aid the development process. However, in practical applications different boundary conditions are often encountered. Developing the hyper-reduction methods further in a way that properly captures the energy evolution in the presence of these boundary conditions is a useful research topic.

References

[1] B.M. Afkham and J.S. Hesthaven. "Structure-preserving model-reduction of dissipative Hamiltonian systems". In: *Journal of Scientific Computing* 81.1 (2019), pp. 3–21.

[2] B.M. Afkham, N. Ripamonti, Q. Wang, and J.S. Hesthaven. "Conservative model order reduction for fluid flow". In: *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*. Ed. by M. D'Elia, M. Gunzburger, and G. Rozza. Cham: Springer International Publishing, 2020, pp. 67–99. ISBN: 978-3-030-48721-8. DOI: 10.1007/978-3-030-48721-8_4.

[3] S Ahmed, O. San, D. Bistrian, and I. Navon. "Sampling and resolution characteristics in reduced order models of shallow water equations: Intrusive vs nonintrusive". In: *International Journal for Numerical Methods in Fluids* 92 (Jan. 2020). DOI: 10.1002/fld.4815.

[4] S.E. Ahmed and O. San. "Breaking the Kolmogorov barrier in model reduction of fluid flows". In: *Fluids* 5.1 (2020). ISSN: 2311-5521.

[5] N. Akkari, F. Casenave, and V. Moureau. "Time stable reduced order modeling by an enhanced reduced order basis of the turbulent and incompressible 3D Navier–Stokes equations". In: *Mathematical and Computational Applications* 24 (Apr. 2019), p. 45. DOI: 10.3390/mca24020045.

[6] D. Amsallem and C. Farhat. "Stabilization of projection-based reduced-order models". In: *International Journal for Numerical Methods in Engineering* 91.4 (2012), pp. 358–377. DOI: 10.1002/nme.4274.

[7] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2009. ISBN: 9780898716580.

[8] A. Arakawa. "Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I". In: *Journal of Computational Physics* 1.1 (1966), pp. 119–143. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(66)90015-5.

[9] J. P. Argaud, B. Bouriquet, H. Gong, Y. Maday, and O. Mula. "Stabilization of (G)EIM in presence of measurement noise: Application to nuclear reactor physics". In: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*. Ed. by M.L. Bittencourt, N.A. Dumont, and J.S. Hesthaven. Cham: Springer International Publishing, 2017, pp. 133–145. ISBN: 978-3-319-65870-4.

[10] R. Aris. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Dover Books on Mathematics. Dover Publications, 1990. ISBN: 9780486661100.

[11] V.I. Arnol'd. "The Hamiltonian nature of the Euler equations in the dynamics of a rigid body and of an ideal fluid". In: (1969).

[12] M. Balajewicz and E.H. Dowell. "Stabilization of projection-based reduced order models of the Navier-Stokes". In: *Nonlinear Dynamics* 70.2 (2012), pp. 1619–1632. DOI: 10.1007/s11071-012-0561-5.

[13] M. Balajewicz, E.H. Dowell, and B. Noack. "Low-dimensional modelling of high Reynolds number shear flows incorporating constraints from the Navier-Stokes equation". In: *Journal of Fluid Mechanics* 729 (Aug. 2013), pp. 285–308. DOI: 10.1017/jfm.2013.278.

[14] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. "An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2004.08.006.

[15] G.K. Batchelor. “Computation of the energy spectrum in homogeneous two-dimensional turbulence”. In: *Physics of Fluids* 12 (1969).

[16] P. Benner, P. Goyal, J. Heiland, and I. Pontes Duff. “Operator inference and physics-informed learning of low-dimensional models for incompressible flows”. In: *ArXiv* (2020).

[17] P. Blonigan, K. Carlberg, F. Rizzi, M. Howard, and J. Fike. “Model reduction for hypersonic aerodynamics via conservative LSPG projection and hyper-reduction”. In: *AIAA Paper 2020-0104, AIAA Scitech 2020 Forum, Orlando, FL* (Jan. 2020).

[18] J. Borggaard, A. Hay, and D. Pelletier. “Interval-based reduced-order models for unsteady fluid flow”. In: *International Journal of Numerical Analysis and Modeling* 4 (Jan. 2007), pp. 353–367.

[19] S. Börm. *Efficient Numerical Methods for Non-local Operators: H2-matrix Compression, Algorithms and Analysis*. EMS tracts in mathematics. European Mathematical Society, 2010. ISBN: 9783037190913.

[20] S. Boyd, S.P. Boyd, and L. Vandenberghe. *Convex Optimization*. Berichte über verteilte messysteme pt. 1. Cambridge University Press, 2004. ISBN: 9780521833783.

[21] S.L. Brunton and J.N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2022. ISBN: 9781009098489.

[22] K. Carlberg, M. Barone, and H. Antil. “Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction”. In: *Journal of Computational Physics* 330 (2017), pp. 693–734. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2016.10.033. URL: https://www.sciencedirect.com/science/article/pii/S0021999116305319.

[23] K. Carlberg, C. Bou-Mosleh, and C. Farhat. “Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations”. In: *International Journal for Numerical Methods in Engineering* 86 (Apr. 2011), pp. 155–181. DOI: 10.1002/nme.3050.

[24] K. Carlberg, Y. Choi, and S. Sargsyan. “Conservative model reduction for finite-volume models”. In: *Journal of Computational Physics* 371 (Nov. 2017). DOI: 10.1016/j.jcp.2018.05.019.

[25] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. “The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows”. In: *Journal of Computational Physics* 242 (2013), pp. 623–647. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2013.02.028.

[26] J. Chan. “Entropy stable reduced order modeling of nonlinear conservation laws”. In: *Journal of Computational Physics* 423 (Dec. 2020), p. 109789. DOI: 10.1016/j.jcp.2020.109789.

[27] S. Chaturantabut, C. Beattie, and S. Gugercin. “Structure-preserving model reduction for nonlinear port-Hamiltonian systems”. In: *SIAM Journal on Scientific Computing* 38.5 (2016), B837–B865.

[28] S. Chaturantabut and D.C. Sorensen. “A state space error estimate for POD-DEIM nonlinear model reduction”. In: *SIAM J. Numer. Anal.* 50 (2012), pp. 46–63.

[29] S. Chaturantabut and D.C. Sorensen. “Nonlinear model reduction via discrete empirical interpolation”. In: *SIAM J. Sci. Comput.* 32 (2010), pp. 2737–2764.

[30] Saifon Chaturantabut. “Nonlinear Model Reduction via Discrete Empirical Interpolation”. PhD thesis. Rice University, 2011.

[31] I.M. Cohen, P.K. Kundu, and H. Hu. *Fluid Mechanics*. Elsevier Science, 2004. ISBN: 9780080470238.

[32] D.M. Copeland, S.W. Cheung, K. Huynh, and Y. Choi. “Reduced order models for Lagrangian hydrodynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 388 (2022), p. 114259. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.114259.

[33] G. Coppola, F. Capuano, and L. De Luca. "Discrete energy-conservation properties in the numerical simulation of the Navier–Stokes equations". In: *Applied Mechanics Reviews* 71 (Feb. 2019). DOI: 10.1115/1.4042820.

[34] M. D'Elia, M. Gunzburger, and G. Rozza. *Quantification of Uncertainty: Improving Efficiency and Technology QUIET selected contributions: QUIET selected contributions*. Jan. 2020. ISBN: 978-3-030-48720-1. DOI: 10.1007/978-3-030-48721-8.

[35] P.A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. Oxford University Press, 2015. ISBN: 9780198722588.

[36] C. Farhat, P. Avery, T. Chapman, and J. Cortial. "Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency". In: *International Journal for Numerical Methods in Engineering* 98 (June 2014). DOI: 10.1002/nme.4668.

[37] C. Farhat, T. Chapman, and P. Avery. "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models". In: *International Journal for Numerical Methods in Engineering* 102 (2015), pp. 1077–1110.

[38] L. Fick, Y. Maday, A.T. Patera, and T. Taddei. "A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection". In: *Journal of Computational Physics* 371 (2018), pp. 214–243. DOI: 10.1016/j.jcp.2018.05.027.

[39] D. Fortunato and A. Townsend. "Fast Poisson solvers for spectral methods". In: *IMA Journal of Numerical Analysis* 40.3 (Nov. 2019), pp. 1994–2018. ISSN: 0272-4979. DOI: 10.1093/imanum/drz034.

[40] Y. Gong, Q. Wang, and Z. Wang. "Structure-preserving Galerkin POD reduced-order modeling of Hamiltonian systems". In: *Computer Methods in Applied Mechanics and Engineering* 315 (2017), pp. 780–798. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.11.016.

[41] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh. "Mesh sampling and weighting for the hyperreduction of nonlinear Petrov-Galerkin reduced-order models with local reduced-order bases". In: *International Journal for Numerical Methods in Engineering* 122 (Dec. 2020). DOI: 10.1002/nme.6603.

[42] S. Grimberg, C. Farhat, and N. Youkilis. "On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows". In: *J. Comput. Phys.* 419 (2020), p. 109681.

[43] C.A. Hall. "Numerical solution of Navier-Stokes problems by the dual variable method". English. In: *SIAM Journal on Matrix Analysis and Applications* 6.2 (Apr. 1985), pp. 220–17.

[44] F. H. Harlow and J. E. Welch. "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface". In: *The Physics of Fluids* 8.12 (1965), pp. 2182–2189. DOI: 10.1063/1.1761178.

[45] D. Hartmann, M. Herz, and U. Wever. "Model order reduction a key technology for digital twins". In: *Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing*. Apr. 2018, pp. 167–179. ISBN: 978-3-319-75318-8. DOI: 10.1007/978-3-319-75319-5_8.

[46] J.S. Hesthaven, C. Pagliantini, and N. Ripamonti. "Structure-preserving model order reduction of Hamiltonian systems". In: *ArXiv* (2021).

[47] P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 2012. ISBN: 9781107008250.

[48] C. Huang, C. Wentland, K. Duraisamy, and C. Merkle. "Model reduction for multi-scale transport problems using structure-preserving least-squares projections with variable transformation". In: *ArXiv* (Nov. 2020).

[49] W.L. IJzerman. “Signal Representation and Modeling of Spatial Structures in Fluids”. English. PhD thesis. University of Twente, Apr. 2000. ISBN: 90-365-1425-8.

[50] I. Kalashnikova and S. Arunajatesan. “A stable Galerkin reduced order model for compressible flow”. In: *Blucher Mechanical Engineering Proceedings* 1.1 (2014), pp. 1399–1423. ISSN: 2358-0828. DOI: http://dx.doi.org/10.5151/meceng-wccm2012-18407.

[51] D. I. Ketcheson. “Relaxation Runge–Kutta methods: Conservation and stability for inner-product norms”. In: *SIAM Journal on Numerical Analysis* 57.6 (2019), pp. 2850–2870. DOI: 10.1137/19M1263662.

[52] R. Knikker. “Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows”. In: *International Journal for Numerical Methods in Fluids* 59 (2009).

[53] B. Koren, R. Abgrall, P. Bochev, J. Frank, and B. Perot. “Physics-compatible numerical methods”. In: *Journal of Computational Physics* 257 (2014). Physics-compatible numerical methods, p. 1039. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2013.10.015.

[54] R.H. Kraichnan and D. Montgomery. “Two-dimensional turbulence”. In: *Reports on Progress in Physics* 43.5 (1980), p. 547.

[55] K. Lee and K.T. Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (2020), p. 108973. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2019.108973.

[56] J. Loiseau and S.L. Brunton. “Constrained sparse Galerkin regression”. In: *Journal of Fluid Mechanics* 838 (2018), pp. 42–67. DOI: 10.1017/jfm.2017.823.

[57] T. Lu and S. Shiou. “Inverses of 2 × 2 block matrices”. In: *Computers Mathematics with Applications* 43.1 (2002), pp. 119–129. ISSN: 0898-1221. DOI: https://doi.org/10.1016/S0898-1221(01)00278-4.

[58] Barone M.F., Kalashnikova I., Segalman D.J., and Thornquist H.K. “Stable Galerkin reduced order models for linearized compressible flow”. In: *Journal of Computational Physics* 228.6 (2009), pp. 1932–1946. DOI: 10.1016/j.jcp.2008.11.015.

[59] K. Manohar, B.W. Brunton, J.N. Kutz, and S.L. Brunton. “Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns”. In: *IEEE Control Systems* 38.3 (June 2018), pp. 63–86. DOI: 10.1109/mcs.2018.2810460.

[60] R. Maulik, B. Lusch, and P. Balaprakash. “Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders”. In: *Physics of Fluids* 33.3 (Mar. 2021), p. 037106. ISSN: 1089-7666. DOI: 10.1063/5.0039986. URL: http://dx.doi.org/10.1063/5.0039986.

[61] S.A. McQuarrie, S. Huang, and K.E. Willcox. “Data-driven reduced-order models via regularised operator inference for a single-injector combustion process”. In: *Journal of the Royal Society of New Zealand* 51.2 (2021), pp. 194–211. DOI: 10.1080/03036758.2020.1863237.

[62] M.L. Minion and D.L. Brown. “Performance of under-resolved two-dimensional incompressible flow simulations, II”. In: *Journal of Computational Physics* 138.2 (1997), pp. 734–765. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1997.5843.

[63] Y. Miyatake. “Structure-preserving model reduction for dynamical systems with a first integral”. In: *Japan Journal of Industrial and Applied Mathematics* (2019).

[64] M. Mohebujjaman, L. Rebholz, and T. Iliescu. “Physically-constrained data-driven correction for reduced order modeling of fluid flows”. In: *International Journal for Numerical Methods in Fluids* 89 (Sept. 2018). DOI: 10.1002/fld.4684.

[65] M. Mohebujjaman, L.G. Rebholz, X. Xie, and T. Iliescu. “Energy balance and mass conservation in reduced order models of fluid flows”. In: *Journal of Computational Physics* 346 (Oct. 2017), pp. 262–277. DOI: 10.1016/j.jcp.2017.06.019.

[66] N.T. Mücke, S.M. Bohté, and C.W. Oosterlee. "Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning". In: *J. Comput. Sci.* 53 (2021), p. 101408.

[67] A. Nishida. "Experience in Developing an Open Source Scalable Software Infrastructure in Japan". In: *Computational Science and Its Applications – ICCSA 2010*. Ed. by D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B.O. Apduhan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 448–462. ISBN: 978-3-642-12165-4.

[68] B.R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*. CISM International Centre for Mechanical Sciences. Springer Vienna, 2011. ISBN: 9783709107584.

[69] E. Noether. "Invariante variationsprobleme". ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1918 (1918), pp. 235–257.

[70] P.J. Olver. "A nonlinear Hamiltonian structure for the euler equations". In: *Journal of Mathematical Analysis and Applications* (1982).

[71] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz. "Localized discrete empirical interpolation method". In: *SIAM Journal on Scientific Computing* 36.1 (2014), A168–A192. DOI: 10.1137/130924408.

[72] B. Peherstorfer, Z. Drmavc, and S. Gugercin. "Stabilizing discrete empirical interpolation via randomized and deterministic oversampling". In: *ArXiv* (2018).

[73] B. Peherstorfer and K. Willcox. "Data-driven operator inference for nonintrusive projection-based model reduction". In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.03.025.

[74] B. Peherstorfer and K. Willcox. "Online adaptive model reduction for nonlinear systems via low-rank updates". In: *SIAM Journal on Scientific Computing* 37.4 (2015), A2123–A2150. DOI: 10.1137/140989169.

[75] N.A. Phillips. "An example of non-linear computational instability". In: *The Atmosphere and the Sea in Motion* (1959).

[76] R. Polyuga and A. Schaft. "Effort and flow-constraint reduction methods for structure preserving model reduction of port-Hamiltonian systems". In: *Systems Control Letters* 61 (Mar. 2012). DOI: 10.1016/j.sysconle.2011.12.008.

[77] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Different Equations*. Vol. 23. Jan. 1994.

[78] F. Romor, G. Stabile, and G. Rozza. "Non-linear manifold ROM with convolutional autoencoders and reduced over-collocation method". In: *ArXiv* (2022). DOI: 10.48550/ARXIV.2203.00360.

[79] C.W. Rowley, T. Colonius, and R.M. Murray. "Model reduction for compressible flows using POD and Galerkin projection". In: *Physica D: Nonlinear Phenomena* 189.1 (2004), pp. 115–129. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2003.03.001.

[80] G. Rozza, M. Hess, G. Stabile, M. Tezzele, and F. Ballarin. "1 Basic ideas and tools for projection-based model reduction of parametric partial differential equations". In: *Volume 2 Snapshot-Based Methods and Algorithms: Volume 2: Snapshot-Based Methods and Algorithms*. Ed. by P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L.M. Silveira. De Gruyter, 2020, pp. 1–47. DOI: doi:10.1515/9783110671490-001. URL: https://doi.org/10.1515/9783110671490-001.

[81] Y. Saad and M. H. Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869. DOI: 10.1137/0907058.

[82] A.K. Saibaba. "Randomized discrete empirical interpolation method for nonlinear model reduction". In: *SIAM Journal on Scientific Computing* 42.3 (2020), A1582–A1608. DOI: 10.1137/19M1243270.

[83] O. San and A. Staples. "High-order methods for decaying two-dimensional homogeneous isotropic turbulence". In: *Computers Fluids* 63 (Dec. 2012). DOI: 10.1016/j.compfluid.2012.04.006.

[84] B. Sanderse. "Energy-Conserving Discretization Methods for the Incompressible Navier-Stokes Equations : Application to the Simulation of Wind-Turbine Wakes". English. PhD thesis. Centrum voor Wiskunde en Informatica, 2013. ISBN: 978-90-386-3338-1. DOI: 10.6100/IR750543.

[85] B. Sanderse. "Energy-conserving Runge–Kutta methods for the incompressible Navier–Stokes equations". In: *Journal of Computational Physics* 233 (2013), pp. 100–131. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2012.07.039.

[86] B. Sanderse. "Non-linearly stable reduced-order models for incompressible flow with energy-conserving finite volume methods". In: *Journal of Computational Physics* 421 (July 2020), p. 109736. DOI: 10.1016/j.jcp.2020.109736.

[87] B. Sanderse and B. Koren. "Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations". In: *Journal of Computational Physics* 231.8 (2012), pp. 3041–3063. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2011.11.028.

[88] C. Sanderson and R. Curtin. "A user-friendly hybrid sparse matrix class in C++". In: *Mathematical Software – ICMS 2018*. Ed. by J.H. Davenport, M. Kauers, G. Labahn, and J. Urban. Cham: Springer International Publishing, 2018, pp. 422–430. ISBN: 978-3-319-96418-8.

[89] C. Sanderson and R. Curtin. "Armadillo: A template-based C++ library for linear algebra". In: *Journal of Open Source Software* 1 (July 2016), p. 26. DOI: 10.21105/joss.00026.

[90] A.M. Schein, K.T. Carlberg, and M.J. Zahr. "Preserving general physical properties in model reduction of dynamical systems via constrained-optimization projection". In: *ArXiv* abs/2011.13998 (2020).

[91] P. Schroeder and G. Lube. "Divergence-free H(div)-FEM for time-dependent incompressible flows with applications to high Reynolds number vortex dynamics". In: *Journal of Scientific Computing* 75 (May 2018), pp. 830–858. DOI: 10.1007/s10915-017-0561-1.

[92] H. Sharma, Z. Wang, and B. Kramer. "Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems". In: *Physica D: Nonlinear Phenomena* 431 (2022), p. 133122. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2021.133122.

[93] J. Sherman and W.J. Morrison. "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix". In: *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127. DOI: 10.1214/aoms/1177729893.

[94] L. Sirovich. "Turbulence and the dynamics of coherent structures. I - Coherent structures. II - Symmetries and transformations. III - Dynamics and scaling". In: *Quarterly of Applied Mathematics - QUART APPL MATH* 45 (Oct. 1987). DOI: 10.1090/qam/910463.

[95] B.E. Sonday, A. Singer, C.W. Gear, and I.G. Kevrekidis. "Manifold learning techniques and model reduction applied to dissipative PDEs". In: *ArXiv* (2010).

[96] B. S. Southworth, O. Krzysik, and W. Pazner. "Fast solution of fully implicit Runge–Kutta and discontinuous Galerkin in time for numerical PDEs, part II: Nonlinearities and DAEs". In: *SIAM Journal on Scientific Computing* 44.2 (2022), A636–A663. DOI: 10.1137/21M1390438.

[97] R. Temam, J.L. Lions, G. Papanicolaou, and R.T. Rockafellar. *Navier—Stokes Equations: Theory and Numerical Analysis*. Studies in mathematics and its applications. Elsevier Science, 2016. ISBN: 9781483256856.

[98] I. Tezaur and M. Barone. “Stable and efficient Galerkin reduced order models for nonlinear fluid flow”. In: 6th AIAA Theoretical Fluid Mechanics Conference. June 2011. ISBN: 978-1-62410-144-1. DOI: 10.2514/6.2011-3110.

[99] A. Treuille, A. Lewis, and Z. Popovic. “Model reduction for real-time fluids”. In: *ACM Trans. Graph.* 25 (July 2006), pp. 826–834. DOI: 10.1145/1141911.1141962.

[100] R. Verstappen and A. Veldman. “Symmetry-preserving discretization of turbulent flow”. In: *Journal of Computational Physics* 187 (May 2003), pp. 343–368. DOI: 10.1016/S0021-9991(03)00126-8.

[101] S. Volkwein. “Proper orthogonal decomposition: Theory and reduced-order modelling”. In: *Lecture Notes, University of Konstanz* (Jan. 2012).

[102] Z. Wang. “Structure-preserving Galerkin POD-DEIM reduced-order modeling of Hamiltonian systems”. In: *ArXiv* abs/2103.00388 (2021).

[103] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer Series in Computational Mathematics. Springer, 2001. ISBN: 9783540678533.

[104] M. Xiao, P. Breitkopf, R. Filomeno Coelho, C. Knopf-Lenoir, P. Villon, and W. Zhang. “Constrained proper orthogonal decomposition based on QR-factorization for aerodynamical shape optimization”. In: *Applied Mathematics and Computation* 223 (2013), pp. 254–263. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2013.07.086. URL: https://www.sciencedirect.com/science/article/pii/S0096300313008424.

[105] X. Xie, M. Mohebujjaman, L. Rebholz, and T. Iliescu. “Data-driven filtered reduced order modeling of fluid flows”. In: *SIAM Journal on Scientific Computing* 40 (Sept. 2017). DOI: 10.1137/17M1145136.

[106] Y. Zhang, A. Palha, M. Gerritsma, and L. Rebholz. “A mass-, kinetic energy- and helicity-conserving mimetic dual-field discretization for three-dimensional incompressible Navier-Stokes equations, part I: Periodic domains”. In: (Apr. 2021).

[107] R. Zimmermann and K.E. Willcox. “An accelerated greedy missing point estimation procedure”. In: *SIAM J. Sci. Comput.* 38 (2016).

A

Consistency of the DEIM

The DEIM provides an exact reconstruction of a nonlinearity $N(\boldsymbol{u}) : \mathbb{R}^N \to \mathbb{R}^N$ when the following condition is satisfied:

$$N(\boldsymbol{u}) \in \mathcal{M}_d. \tag{A.1}$$

The proof of this is as follows. If condition (A.1) holds then the output of the nonlinearity may be written as:

$$N(\boldsymbol{u}) = M\boldsymbol{c}_u.$$

Approximating the nonlinearity using the DEIM approximation (eq.(4.32)) and performing the procedure to find the DEIM coordinates (eq.(4.34)), the following can be written:

$$P^T M\boldsymbol{c} = P^T N(\boldsymbol{u}) = P^T M\boldsymbol{c}_u.$$

Solving for $\boldsymbol{c}$ gives:

$$\boldsymbol{c} = \boldsymbol{c}_u.$$

Thus, using the DEIM, the exact reconstruction of a nonlinear operator $N(\boldsymbol{u})$ can be obtained when condition (A.1) is satisfied. Condition (A.1) is definitely satisfied if $m = N$, but may be satisfied in the case $m \leq N$ when $\mathcal{M}_d$ is sufficiently accurate.

B

Jacobian Derivations

B.1. ROM Jacobian

The Jacobian of the reduced convection operator $J_r(\boldsymbol{a}) = \frac{\partial C_r}{\partial \boldsymbol{a}}(\boldsymbol{a})$ can be expressed in terms of the Jacobian of the FOM operator C_h as follows:

$$
\begin{aligned}
(J_r(\boldsymbol{a}))_{ij} &= \frac{\partial}{\partial a_j} C_r(\boldsymbol{a})_i \\
&= \frac{\partial}{\partial a_j} \left(\sum_k \left(\Phi^T\right)_{ik} C_h(\Phi \boldsymbol{a})_k \right) \\
&= \sum_k \left(\Phi^T\right)_{ik} \frac{\partial}{\partial a_j} \left(C_h(\Phi \boldsymbol{a})_k \right) \\
&= \sum_k \left(\Phi^T\right)_{ik} \sum_l \frac{\partial}{\partial q_l} \left(C_h(\boldsymbol{q})_k \right) \frac{\partial q_l}{\partial a_j} \\
&= \sum_k \left(\Phi^T\right)_{ik} \sum_l \left(J_h(\boldsymbol{q}) \right)_{kl} \frac{\partial q_l}{\partial a_j} \\
&= \sum_k \left(\Phi^T\right)_{ik} \sum_l \left(J_h(\Phi \boldsymbol{a}) \right)_{kl} \frac{\partial}{\partial a_j} \sum_m (\Phi)_{lm} a_m \\
&= \sum_k \left(\Phi^T\right)_{ik} \sum_l \left(J_h(\Phi \boldsymbol{a}) \right)_{kl} (\Phi)_{lj} \\
&= \left(\Phi^T J_h \left(\Phi \boldsymbol{a} \right) \Phi \right)_{ij}.
\end{aligned}
$$

B.2. Exact Tensor Decomposition Jacobian

$$
\begin{aligned}
\left(\frac{\partial C_r(\boldsymbol{a} \otimes \boldsymbol{a})}{\partial \boldsymbol{a}} \right)_{ij} &= \frac{\partial}{\partial a_j} \left(\sum_{k,l=1}^{r} (\Phi^T \widetilde{C}_h(\Phi_{,k}) \Phi)_{il} a_k a_l \right) \\
&= \sum_{k,l=1}^{r} (\Phi^T \widetilde{C}_h(\Phi_{,k}) \Phi)_{il} \frac{\partial (a_k a_l)}{\partial a_j} \\
&= \sum_{k,l=1}^{r} (\Phi^T \widetilde{C}_h(\Phi_{,k}) \Phi)_{il} \left(a_l \delta_{kj} + a_k \delta_{lj} \right),
\end{aligned}
$$

B.3. DEIM Jacobian

In case it is desired to integrate (4.37) in time using implicit time-integration schemes, the Jacobian $J_m(\boldsymbol{a}) : \mathbb{R}^r \to \mathbb{R}^{r\times r}$ of the DEIM approximation can be determined as follows:

$$J_m(\boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{a}} \left(\Phi^T M \boldsymbol{c}(\boldsymbol{a})\right) = \Phi^T M \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}.$$

Now using (4.38) gives:

$$\Phi^T M \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} = \Phi^T M \frac{\partial}{\partial \boldsymbol{a}} \left((P^T M)^{-1} P^T C_h(\Phi \boldsymbol{a})\right) = \Phi^T M (P^T M)^{-1} P^T J_h(\Phi \boldsymbol{a}) \Phi.$$

Thus, for the DEIM Jacobian $J_m(\boldsymbol{a})$ it can then be written:

$$J_m(\boldsymbol{a}) = \Phi^T M (P^T M)^{-1} P^T J_h(\Phi \boldsymbol{a}) \Phi,$$

which can be interpreted as selected rows of the FOM Jacobian being interpolated using the DEIM procedure, written in terms of $\boldsymbol{a}$ using the chain-rule and reduced using the Galerkin projection.

B.4. LSDEIM/DLSDEIM Jacobian

This Jacobian can be calculated as follows:

$$J_m(\boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{a}} \left(\Phi^T M \boldsymbol{c}(\boldsymbol{a})\right) = \Phi^T M \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}.$$

In turn $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}$ can be calculated using (4.46). The Jacobian of the left term in the right-hand side of (4.46) is relatively simple to determine and given as:

$$\frac{\partial}{\partial \boldsymbol{a}} \left(A^{-1}(P^T M)^T P^T C_h(\Phi \boldsymbol{a})\right) = A^{-1}(P^T M)^T P^T J_h(\Phi \boldsymbol{a}) \Phi.$$

The term on the right in the right-hand side of (4.46) requires more attention. Denoting the scalar factor in the right term as:

$$\gamma(\boldsymbol{a}) := \frac{\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{d}(\boldsymbol{a})}{\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{b}(\boldsymbol{a})},$$

where the notation $\boldsymbol{d}(\boldsymbol{a}) := (P^T M)^T P^T C_h(\Phi \boldsymbol{a})$ was introduced, the product rule can be applied:

$$\frac{\partial}{\partial a_j} \left(\gamma(\boldsymbol{a})(A^{-1}\boldsymbol{b}(\boldsymbol{a}))_i\right) = \frac{\partial \gamma}{\partial a_j}(A^{-1}\boldsymbol{b}(\boldsymbol{a}))_i + \sum_{k=1}^{m} \gamma(\boldsymbol{a})(A^{-1})_{ik} \frac{\partial b_k}{\partial a_j}. \tag{B.1}$$

Using the definition of $\boldsymbol{b}(\boldsymbol{a})$ the right term in (B.1) can be written as follows:

$$\sum_{k=1}^{m} \gamma(\boldsymbol{a})(A^{-1})_{ik} \frac{\partial b_k}{\partial a_j} = \sum_{k=1}^{m} \gamma(\boldsymbol{a})(A^{-1})_{ik} (M^T \Phi)_{kj}.$$

The partial derivative in the left term in (B.1) can be calculated using the quotient-rule:

$$\frac{\partial \gamma}{\partial a_j} = \frac{\left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{b}(\boldsymbol{a})\right] \frac{\partial}{\partial a_j} \left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{d}(\boldsymbol{a})\right) - \left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{d}(\boldsymbol{a})\right] \frac{\partial}{\partial a_j} \left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{b}(\boldsymbol{a})\right)}{\left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{b}(\boldsymbol{a})\right)^2}.$$

In turn the partial derivative of $\left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{d}(\boldsymbol{a})\right)$ is given as:

$$\begin{aligned}
\frac{\partial}{\partial a_j} \left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1} \boldsymbol{d}(\boldsymbol{a})\right) &= \frac{\partial}{\partial a_j} \left(\sum_{k=1}^{m} b_k \left[\sum_{q=1}^{m} (A^{-1})_{kq} d_q\right]\right) \\
&= \sum_{k=1}^{m} \sum_{q=1}^{m} \left(\frac{\partial b_k}{\partial a_j} (A^{-1})_{kq} d_q + b_k (A^{-1})_{kq} \frac{\partial d_q}{\partial a_j}\right) \\
&= (\Phi^T M A^{-1} \boldsymbol{c}(\boldsymbol{a}))_j + (\boldsymbol{b}(\boldsymbol{a})^T A^{-1} (P^T M)^T P^T J_h(\Phi \boldsymbol{a}) \Phi)_j,
\end{aligned}$$

where the definitions of $\boldsymbol{b}(\boldsymbol{a})$ and $\boldsymbol{d}(\boldsymbol{a})$ were used to evaluate the partial derivatives in the second line. Finally, the partial derivative of $\left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})\right)$ in (B.1) can be calculated as follows:

$$\begin{aligned}\frac{\partial}{\partial a_j}\left(\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})\right) &= \frac{\partial}{\partial a_j}\left(\boldsymbol{a}^T\Phi^T M A^{-1} M^T\Phi\boldsymbol{a}\right)\\ &= \sum_{k=1}^{r}\sum_{q=1}^{r}\left(\frac{\partial a_k}{\partial a_j}(\Phi^T M A^{-1}M^T\Phi)_{kq}a_q + a_k(\Phi^T M A^{-1}M^T\Phi)_{kq}\frac{\partial a_q}{\partial a_j}\right)\\ &= (\Phi^T M A^{-1}M^T\Phi\boldsymbol{a})_j + (\boldsymbol{a}^T\Phi^T M A^{-1}M^T\Phi)_j.\end{aligned}$$

Combining all of the above an expression for $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}}$ is as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}} =& 2A^{-1}(P^T M)^T P^T J_h(\Phi\boldsymbol{a})\Phi - 2\gamma(\boldsymbol{a})A^{-1}M^T\Phi\\ &- 2A^{-1}\boldsymbol{b}(\boldsymbol{a})\otimes\left[\frac{1}{\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})}\left(\Phi^T M A^{-1}\boldsymbol{c}(\boldsymbol{a}) + \left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1}(P^T M)^T P^T J_h(\Phi\boldsymbol{a})\Phi\right]^T\right)\right]\\ &- 2A^{-1}\boldsymbol{b}(\boldsymbol{a})\otimes\left[\frac{\gamma(\boldsymbol{a})}{\boldsymbol{b}(\boldsymbol{a})^T A^{-1}\boldsymbol{b}(\boldsymbol{a})}\left(\Phi^T M A^{-1}M^T\Phi\boldsymbol{a} + \left[\boldsymbol{b}(\boldsymbol{a})^T A^{-1}M^T\Phi\right]^T\right)\right],\end{aligned}$$

multiplication from the left with $\Phi^T M$ provides an expression for the Jacobian $J_m(\boldsymbol{a})$.

B.5. SMDEIM Jacobian

Using the previously found expression (4.51) for $\boldsymbol{c}_m$ it can be written:

$$\begin{aligned}\frac{\partial \boldsymbol{c}_m}{\partial \boldsymbol{a}} &= \frac{\partial}{\partial \boldsymbol{a}}\left(\left[M_p^{-1} - \frac{M_p^{-1}(P^T M)_{,m+1}\boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}}{1+\boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}(P^T M)_{,m+1}}\right]P^T C_h(\Phi\boldsymbol{a})\right)\\ &= M_p^{-1}P^T\frac{\partial}{\partial \boldsymbol{a}}\left(C_h(\Phi\boldsymbol{a})\right) - M_p^{-1}(P^T M)_{,m+1}\otimes\nabla_a\left(\frac{\boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}P^T C_h(\Phi\boldsymbol{a})\Phi\boldsymbol{a}}{1+\boldsymbol{r}(\boldsymbol{a})^T M_p^{-1}(P^T M)_{,m+1}}\right).\end{aligned}$$

Using the quotient rule the gradient in the second term can be calculated as follows:

$$\nabla_a\left(\frac{\boldsymbol{r}(\boldsymbol{a})^T A\boldsymbol{y}(\boldsymbol{a})}{1+\boldsymbol{r}(\boldsymbol{a})^T\boldsymbol{b}}\right) = \frac{\left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}^T A\boldsymbol{y}(\boldsymbol{a}) + \left(A\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{a}}\right)^T\boldsymbol{r}(\boldsymbol{a})\right](1+\boldsymbol{r}(\boldsymbol{a})^T\boldsymbol{b}) - \left[\boldsymbol{r}(\boldsymbol{a})^T A\boldsymbol{y}(\boldsymbol{a})\right]\left(\frac{\partial \boldsymbol{r}(\boldsymbol{a})}{\partial \boldsymbol{a}}^T\boldsymbol{b}\right)}{(1+\boldsymbol{r}(\boldsymbol{a})^T\boldsymbol{b})^2},$$

with $\boldsymbol{y}(\boldsymbol{a}) := C_h(\Phi\boldsymbol{a})$, $A := M_p^{-1}P^T$ and $\boldsymbol{b} := M_p^{-1}(P^T M)_{,m+1}$. The gradient of c_{m+1} is given by:

$$\begin{aligned}\nabla_a c_{m+1} &= \nabla_a\left(\boldsymbol{r}(\boldsymbol{a})^T\boldsymbol{c}_m(\boldsymbol{a})\right)\\ &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}^T\boldsymbol{c}_m(\boldsymbol{a}) + \frac{\partial \boldsymbol{c}_m}{\partial \boldsymbol{a}}^T\boldsymbol{r}(\boldsymbol{a}).\end{aligned}$$

Finally, to construct the Jacobian it is necessary to find $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}$ and $\frac{\partial}{\partial \boldsymbol{a}}\left(C_h(\Phi\boldsymbol{a})\right)$. Firstly, the partial derivative of $\boldsymbol{r}$ is considered:

$$\begin{aligned}\frac{\partial}{\partial a_j}(r_i(\boldsymbol{a})) &= \frac{\partial}{\partial a_j}\left(-\frac{(\boldsymbol{a}^T\Phi^T M)_i}{(\boldsymbol{a}^T\Phi^T M)_{,m+1}}\right) = -\frac{\partial}{\partial a_j}\left(\frac{\sum_{k=1}^r a_k(\Phi^T M)_{ki}}{\sum_{k=1}^r a_k(\Phi^T M)_{k(m+1)}}\right)\\ &= -\frac{(\Phi^T M)_{ji}\left(\sum_{k=1}^r a_k(\Phi^T M)_{k(m+1)}\right) - \left(\sum_{k=1}^r a_k(\Phi^T M)_{ki}\right)(\Phi^T M)_{j(m+1)}}{\left(\sum_{k=1}^r a_k(\Phi^T M)_{k(m+1)}\right)^2},\end{aligned}$$

in matrix notation this is denoted as:

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} = \left(\frac{1}{(\boldsymbol{a}^T\Phi^T M)_{m+1}}\right)\left[-(\Phi^T M I_m)^T + \boldsymbol{r}(\boldsymbol{a})\otimes(\Phi^T M)_{,m+1}\right],$$

where $I_m \in \mathbf{R}^{(m+1)\times m}$ is the first m columns of the $(m+1)\times(m+1)$ identity matrix. Secondly, the partial derivatives of $\boldsymbol{y}(\boldsymbol{a})$ will be considered:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{a}} = \frac{\partial}{\partial \boldsymbol{a}}\left(C_h(\Phi\boldsymbol{a})\right) = \frac{\partial(C_h(\boldsymbol{u}))}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{a}} = J_h(\Phi\boldsymbol{a})\Phi.$$

B.6. Efficient Evaluation of $P^T J_h(\Phi \boldsymbol{a})\Phi$ for DEIM-like Methods

To efficiently evaluate $P^T J_h(\Phi \boldsymbol{a})\Phi$, the sparsity structure of $J_h(\Phi \boldsymbol{a})$ has to be considered. Firstly, multiplication from the left with P^T requires to only evaluate the rows of the matrix $J_h(\Phi \boldsymbol{a})\Phi$ associated to measurement points. Secondly, denoting $\mathcal{J}(i)$ as the set of all column-indices of nonzero components in row i of $J_h(\Phi \boldsymbol{a})$, the necessary rows of $J_h(\Phi \boldsymbol{a})\Phi$ should be evaluated in practise as:

$$(J_h(\Phi \boldsymbol{a})\Phi)_{i,} = \left[\sum_{j \in \mathcal{J}(i)} (J_h(\Phi \boldsymbol{a}))_{ij} \Phi_{j,0}, ..., \sum_{j \in \mathcal{J}(i)} (J_h(\Phi \boldsymbol{a}))_{ij} \Phi_{j,r} \right],$$

where $(J_h(\Phi \boldsymbol{a})\Phi)_{i,}$ denotes the i^{th} row of $J_h(\Phi \boldsymbol{a})\Phi$. Evaluation of $P^T J_h(\Phi \boldsymbol{a})\Phi$ scales as $\mathcal{O}(rm)$, solving $(P^T M)^{-1} P^T J_h(\Phi \boldsymbol{a})\Phi$ scales computationally as $\mathcal{O}(m^2 r)$ as r LU-solves have to be performed and multiplication with $\Phi^T M$ is of order $\mathcal{O}(r^2 m)$ being a matrix-matrix product with an $r \times m$ and an $m \times r$ matrix. Hence, evaluation of $J_m(\boldsymbol{a})$ is of order $\mathcal{O}(\max(m^2 r, r^2 m))$.

C

A Proof of Divergence-Freeness of Φ

The eigenvector problem in step 3 is given as:

$$\hat{X}\hat{X}^T\hat{\Phi}_{,j} = \sigma_j^2\hat{\Phi}_{,j}$$
$$\left(\Omega_h^{1/2}\widetilde{X}\right)\left(\Omega_h^{1/2}\widetilde{X}\right)^T\hat{\Phi}_{,j} = \sigma_j^2\hat{\Phi}_{,j}$$
$$\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)^T\hat{\Phi}_{,j} = \sigma_j^2\hat{\Phi}_{,j}.$$

Transforming the eigenvectors as $\widetilde{\Phi}_{,j} = \Omega_h^{-1/2}\hat{\Phi}_{,j}$ gives:

$$\Omega_h^{-1/2}\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)^T\hat{\Phi}_{,j} = \sigma_j^2\Omega_h^{-1/2}\hat{\Phi}_{,j} = \sigma_j^2\widetilde{\Phi}_{,j}$$
$$\left[X - EE^T\Omega_h X\right]\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)^T\hat{\Phi}_{,j} = \sigma_j^2\widetilde{\Phi}_{,j}.$$

Now taking the discrete divergence results in:

$$M_h\left[X - EE^T\Omega_h X\right]\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)^T\hat{\Phi}_{,j} = \sigma_j^2 M_h\widetilde{\Phi}_{,j}$$
$$\left[M_h X - M_h EE^T\Omega_h X\right]\left(\Omega_h^{1/2}\left[X - EE^T\Omega_h X\right]\right)^T\hat{\Phi}_{,j} = \sigma_j^2 M_h\widetilde{\Phi}_{,j}$$
$$0 = \sigma_j^2 M_h\widetilde{\Phi}_{,j},$$

since both X and E are divergence-free column-wise. Since $\sigma_j > 0\ \forall\ j \leq d_r$ and $r < d_r$ it must hold that $M_h\widetilde{\Phi}_{,j} = 0,\ \forall\ j \leq d_r$. As Φ is the concatenation of $\widetilde{\Phi}$ and E and both matrices are column-wise divergence-free, Φ is divergence-free, completing the proof.