Vietnam Journal of Mathematics (2023) 51:199–211
https://doi.org/10.1007/s10013-022-00590-3

ORIGINAL ARTICLE

Model Reduction Using Sparse Polynomial Interpolation for the Incompressible Navier–Stokes Equations

Martin Hess[1] · **Gianluigi Rozza**[1]

Received: 9 January 2022 / Accepted: 5 August 2022 / Published online: 19 October 2022

Abstract
This work investigates the use of sparse polynomial interpolation as a model order reduction method for the parametrized incompressible Navier–Stokes equations. Numerical results are presented underscoring the validity of sparse polynomial approximations and comparing with established reduced basis techniques. Two numerical models serve to assess the accuracy of the reduced order models (ROMs), in particular parametric nonlinearities arising from curved geometries are investigated in detail. Besides the accuracy of the ROMs, other important features of the method are covered, such as offline-online splitting, run time and ease of implementation. The findings provide a clear indication that sparse polynomial interpolation is a valid instrument in the toolbox of ROM methods.

Keywords Model order reduction · Sparse polynomial interpolation · Navier–stokes equations · Parametric systems · Curved geometries

Mathematics Subject Classification (2010) 78M34 · 76Bxx · 65Mxx

1 Introduction

A reduced order model (ROM) computes inexpensive yet accurate approximate solutions to parametrized partial differential equations (PDEs) in a fast and computationally efficient manner. For an introduction and overview to ROM methods, in particular the reduced basis (RB) method, see for example [1–4, 15, 21].

This work aims to establish sparse polynomial interpolation as a ROM method for the parametrized incompressible Navier–Stokes equations [18]. Two numerical models serve to assess the accuracy of the ROM approximation. Numerical results for both models are

Dedicated to Alfio Quarteroni for his 70th birthday with deep gratitude.

✉ Gianluigi Rozza
grozza@sissa.it

Martin Hess
mhess@sissa.it

[1] mathLab, SISSA, via Bonomea 265, Trieste, 34136, Italy

available using the RB method [11, 12], which allows not only to compare the accuracy of both methods, but also the run time, implementation effort and other desirable features, such as offline-online splitting. The first model has one parametric variation in geometry, which is affinely parametrized. The second model has two parametric variations, the kinematic viscosity and the curvature. The curvature introduces a nonlinear parameter dependency, which can not be easily resolved by RB methods.

There exists a significant body of literature on sparse polynomial interpolation, see [6–9], which establishes the theoretical rationale behind the method. In particular, estimates on the accuracy of the ROM approximations w.r.t. the full order model (FOM) are established. This could be a promising step towards alleviating the so-called *curse of dimensionality*. The curse of dimensionality refers to a (sub-)exponentially increasing computational effort with increasing parameter space dimension. It poses a bottleneck for complex applications where many parameters are present. For example models in uncertainty quantification become unfeasible as the resolution in parameter space is increased.

The remainder of the work is structured as follows. Section 2 introduces the incompressible Navier–Stokes equations and the non-linear solver, while Section 3 recapitulates the sparse interpolation procedure. Section 4 provides and discusses the numerical results and Section 5 concludes the findings and gives a brief outlook.

2 Model Setup

For simplicity of notation, the model setup is presented without explicit parameter dependence. Let $\Omega \in \mathbb{R}^2$ denote the computational domain. Incompressible, viscous fluid motion in spatial domain Ω over a time interval $(0, T)$ is governed by the incompressible Navier–Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu_{\text{visc}} \Delta \mathbf{u} + \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$ is the vector-valued velocity, p is the scalar-valued pressure, ν_{visc} is the kinematic viscosity and $\mathbf{f}$ is a body forcing. Boundary and initial conditions are prescribed as

$$\begin{aligned} \mathbf{u} &= \mathbf{d} \quad \text{on } \Gamma_D \times (0, T), \\ \nabla \mathbf{u} \cdot \mathbf{n} &= \mathbf{g} \quad \text{on } \Gamma_N \times (0, T), \\ \mathbf{u} &= \mathbf{u}_0 \quad \text{in } \Omega \times 0, \end{aligned}$$

with $\mathbf{d}$, $\mathbf{g}$ and $\mathbf{u}_0$ given and $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. The *Reynolds* number Re, which characterizes the flow ([16]), depends on the kinematic viscosity ν_{visc}, on a characteristic velocity U, and on a characteristic length L:

$$Re = \frac{UL}{\nu_{\text{visc}}}. \tag{3}$$

We are interested in the steady states, i.e., solutions where $\frac{\partial \mathbf{u}}{\partial t}$ vanishes. The high-order simulations are computed through time-advancement, while the RB reduced order solutions are computed through fixed-point iterations of a nonlinear solver.

The *Oseen*-iteration is a secant modulus fixed-point iteration, which in general exhibits a linear rate of convergence. It solves for a steady-state solution, i.e., $\frac{\partial \mathbf{u}}{\partial t} = 0$ is assumed.

Given a current iterate (or initial condition) $\mathbf{u}^k$, the next iterate $\mathbf{u}^{k+1}$ is computed by solving the following linear system:

$$\begin{aligned} -\nu_{\text{visc}}\Delta\mathbf{u}^{k+1} + (\mathbf{u}^k \cdot \nabla)\mathbf{u}^{k+1} + \nabla p &= \mathbf{f} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^{k+1} &= 0 \quad \text{in } \Omega, \\ \mathbf{u}^{k+1} &= \mathbf{d} \quad \text{on } \Gamma_D, \\ \nabla\mathbf{u}^{k+1} \cdot \mathbf{n} &= \mathbf{g} \quad \text{on } \Gamma_N. \end{aligned}$$

Iterations are stopped when the relative difference between iterates falls below a predefined tolerance in a suitable norm, like the $L^2(\Omega)$ or $H_0^1(\Omega)$ norm.

3 Sparse Polynomial Interpolation

The presented sparse polynomial interpolation approach is based on the literature references [6] and [7]. The aim is to define a sparse interpolation operator I_Λ for a multiindex set Λ, which acts on a function g as $I_\Lambda[g]$. Lagrange interpolation will be used to define I_Λ, but first some preliminaries will be established. The function g is defined over the parameter domain $\mathcal{P}$ and the parameter domain $\mathcal{P}$ consists of vectors $\mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$ with d the number of parameters. Each parameter direction has been normalized to the interval $[-1, 1]$, such that

$$\mathcal{P} = [-1, 1]^d \subset \mathbb{R}^d.$$

Let $\mathcal{F}$ denote the set multiindices $\nu = (\nu_1, \nu_2, \ldots, \nu_d) \in \mathbb{N}_0^d$. For two multiindices ν and μ, the relation $\nu \geq \mu$ is defined as $\nu_i \geq \mu_i$ for all parameter directions i. An index set Λ of (multi-)indices ν, which fulfills the property that

$$(\nu \in \Lambda \text{ and } \nu \geq \mu) \;\Rightarrow\; \mu \in \Lambda,$$

is called a *downward closed* set. In particular, consider a hierarchical sequence of downward closed sets $\{\Lambda_1, \ldots, \Lambda_n\}$. Let $(z_k)_{k\geq 0}$ be a sequence of mutually distinct points in $[-1, 1]$. The notation z_ν refers to the point $z_\nu = (z_{\nu_1}, z_{\nu_2}, \ldots, z_{\nu_d}) \in \mathbb{R}^d$. The superscript ν^k refers to a multiindex with entries $\nu^k = (\nu_1^k, \nu_2^k, \ldots, \nu_d^k) \in \mathbb{N}_0^d$.

The sparse interpolation operator is of the form

$$I_{\Lambda_n}[g] = \sum_{k=0}^{n} g_{\nu^k} H_{\nu^k} \tag{4}$$

with

$$g_{\nu^k} = g(z_{\nu^k}) - \sum_{i=1}^{k-1} g_{\nu^i} H_{\nu^i}(z_{\nu^k}),$$

where $g(z_{\nu^k})$ is the coefficient vector in the discrete basis of the solution to the PDE (1)–(2) at the parameter point z_{ν^k}.

The polynomials H_ν are defined over the parameter domain by tensorization as

$$H_\nu(\mathbf{y}) = \prod_{j=1}^{d} h_{\nu_j}(y_j),$$

using Lagrange interpolation in each parameter direction as

$$h_k(y) = \prod_{j=0}^{k-1} \frac{y - z_j}{z_k - z_j}, \quad k > 0, \quad h_0(y) = 1.$$

The efficient hierarchical computation of the sparse interpolation operator is shown in more detail in [6].

3.1 Leja Points

It remains to provide the sequences $(z_k)_{k\geq 0}$. The suggested point rules in [6] are Leja sequences, composed of Leja points. Leja points are defined recursively by maximizing

$$F^N(y) = \prod_{i=1}^{N-1} |(y - x_i)|$$

over $[-1, 1]$ for a given initial x_1, such that

$$x_N = \arg \max_{y\in[-1,1]} F^N(y). \tag{5}$$

The equation (5) is not solved, but a fine grid of the interval $[-1, 1]$ is used in practice. Symmetrized Leja points are defined by choosing $x_1 = 0$, $x_2 = 1$, $x_3 = -1$ and then evaluating (5) on a fine grid for even N while for odd N the previous point is repeated with a sign change. A set of points can be put in Leja ordering by restricting the maximization in (5) to the set itself.

4 Numerical Simulations

The sparse polynomial approach is used to generate reduced order models for parametrized channel flows with one and two parameters. To access the quality of the approximations, they are compared against reduced basis (RB) methods based on the proper orthogonal decomposition (POD).

The models are discretized with the spectral element method (SEM) [17] using the PDE framework `Nektar++`[1] and the model reduction software `ITHACA-SEM`[2].

4.1 Channel with a Narrowing of Varying Width

Consider a channel flow with a narrowing of varying width. The velocity field solution at the reference parameter $\mu_{ref} = 1$ is shown in Fig. 1. Some more field solutions are shown in [11] and closely related models have been computed also in [13, 14] and [19]. The geometry is decomposed into 36 triangular spectral elements and the velocity is resolved with modal Legendre polynomials of order 11. The inflow profile on the left side is parabolic with $u_x(0, y) = y(3 - y)$ for $y \in [0, 3]$. At the outlet, a stress-free boundary condition is set and everywhere else hold no-slip conditions.

The parameter domain is $\mu \in [0.1, 2.9]$, which is affinely mapped to the interval $\nu \in [-1, 1]$ to conform with the sparse polynomial approximation assumptions. With a changing parameter, the geometry always remains symmetric to the horizontal centerline at $y = 1.5$. The kinematic viscosity is kept constant at $\nu_{\text{visc}} = 1$.

The parametric variation in geometry allows an affine decomposition of the Navier–Stokes element matrix in the parameter. In particular, Figs. 2 and 3 show the mean and maximum error with growing reduced order dimension for the standard reduced

[1] www.Nektar.info

[2] https://github.com/mathLab/ITHACA-SEM

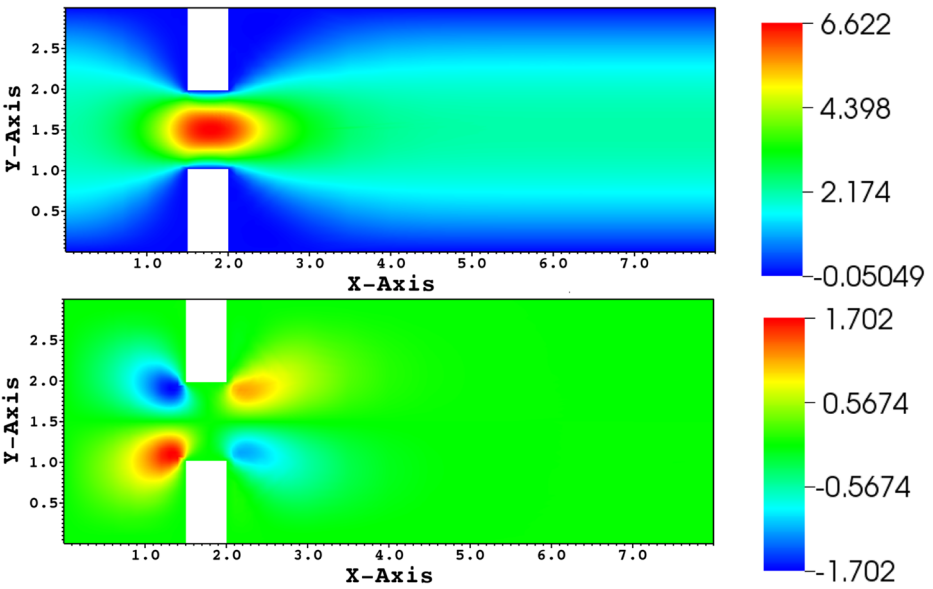

Fig. 1 Full order, steady-state solution for $\mu = 1$: velocity in x-direction (top) and y-direction (bottom)

basis procedure, the Lagrange interpolation with Leja points, the Lagrange interpolation with symmetrized Leja points and Lagrange interpolation with equidistant points in Leja ordering.

Although the sparse polynomial interpolation and the RB are two very different approximation algorithms, the results are both plotted versus the reduced dimension N. In both cases the size of the ROM model is compared. For the sparse interpolation the reduced dimension refers to the number of PDE solutions in the polynomial expansion, i.e., the $n+1$

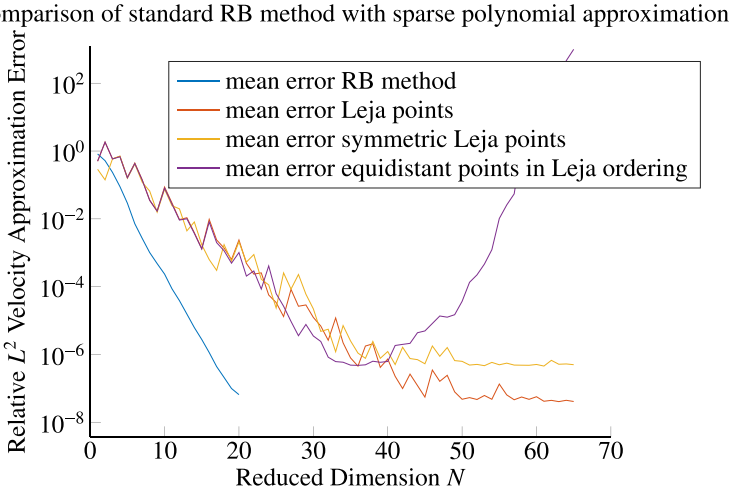

Fig. 2 Comparison of mean error of the RB method with sparse polynomial approximations using Lagrange interpolation and various point rules

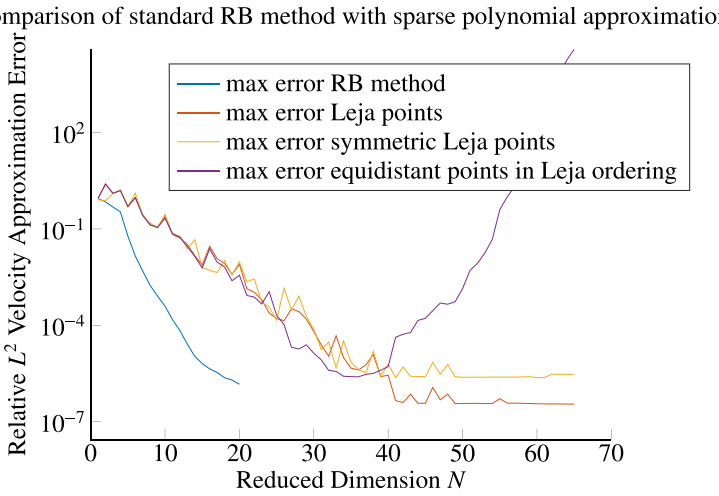

Fig. 3 Comparison of maximum error of the RB method with sparse polynomial approximations using Lagrange interpolation and various point rules

in (4). For the RB the reduced dimension refers to the number of POD modes used to project the equations. This is not a perfectly fair comparison, since more PDE solutions were used in the RB method to generate the POD modes (namely 40 for this model), but it still allows to draw some conclusions.

It can be observed that all methods show a faster than polynomial order convergence, since the slope is linear in a semi-log plot. The three sparse polynomial approximations reach a mean accuracy of six digits in the velocity at a ROM dimension of about 35 and five digits accuracy in the maximum at this dimension. The RB reaches a mean accuracy of six digits in the velocity at a ROM dimension of 20 and five digits accuracy in the maximum at ROM dimension 20.

All three point rules provide a similar approximation quality up a reduced dimension of 35. The equidistantly distributed points in Leja ordering then diminish in approximation quality, which is a known phenomenon, since high order Lagrange interpolants without special choice of interpolation points are ill-conditioned. Using equidistantly distributed points *without* Leja ordering provides no approximation at all and has consistently a mean relative error of about 100%.

The approximations with Leja points do not provide a better approximation than single precision, which can usually be improved up to double precision by re-arranging how terms are computed. The same holds true for RB approximations. However, a stable approximation with six digits of accuracy is usually enough in practical applications.

4.2 Channel with a Narrowing of Varying Curvature

A two parameter model is considered with parametric variation in the curvature of the narrowing and variation in the kinematic viscosity. This model was analysed in [12] and the results established there with the RB method and the *empirical interpolation method* (EIM) will serve to compare the accuracy of ROMs by sparse polynomial interpolation.

Consider the channel flow through a narrowing created by walls of varying curvature and with variable kinematic viscosity. See Figs. 4 and 5 for the steady-state velocity components

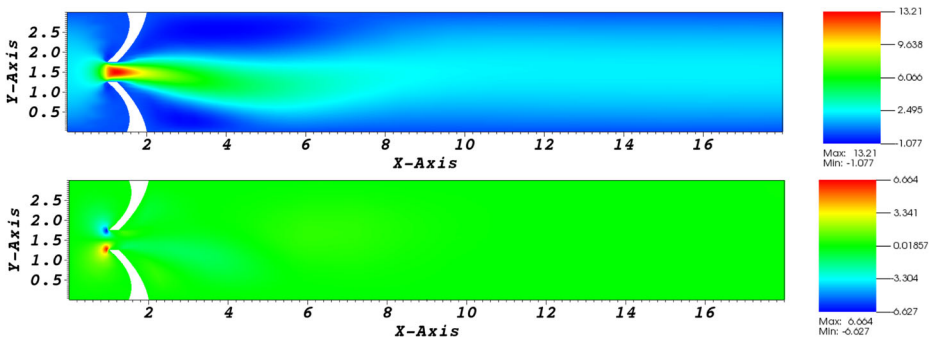

Fig. 4 Full order, steady-state solution in the geometry with curved walls and for $\nu = 0.15$: velocity in x-direction (top) and y-direction (bottom)

for $\nu = 0.15$ in a geometry with curved walls and straight walls, respectively. Figures 6 and 7 show the steady-state velocity components for $\nu = 0.2$ in a geometry with curved walls and straight walls, respectively. These are the four corners of the rectangular parameter domain and constitute the most extreme solutions. Figures 4 and 6 also show the strongest curvature of all configurations.

The spectral element expansion uses modal Legendre polynomials of order $p = 10$ for the velocity. The pressure *ansatz* space is chosen of order $p-2$ to fulfill the inf-sup stability condition ([5, 22]). A parabolic inflow profile is prescribed at the inlet (i.e., $x = 0$) with horizontal velocity component $u_x(0, y) = y(3-y)$ for $y \in [0, 3]$. At the outlet (i.e., $x = 18$) a stress-free boundary condition is imposed, while everywhere else a no-slip condition is prescribed. Symmetric boundary conditions are considered in order to study the symmetry breaking due to the nonlinearity in problem (1)–(2). A more realistic setting considers also different inlet velocity profiles and the pulsatility of the flow and would then include the Strouhal number as a parameter.

The viscosity varies in the interval $\nu_{\text{visc}} \in [0.15, 0.2]$. The Reynolds number Re, see (3), depends on the kinematic viscosity. As Re is varied for each fixed geometry, a supercritical pitchfork bifurcation occurs: for Re higher than the critical bifurcation point, three solutions exist. Two of these solutions are stable, one with a jet towards the top wall and one with a jet towards the bottom wall and one is unstable. The unstable solution is symmetric to the

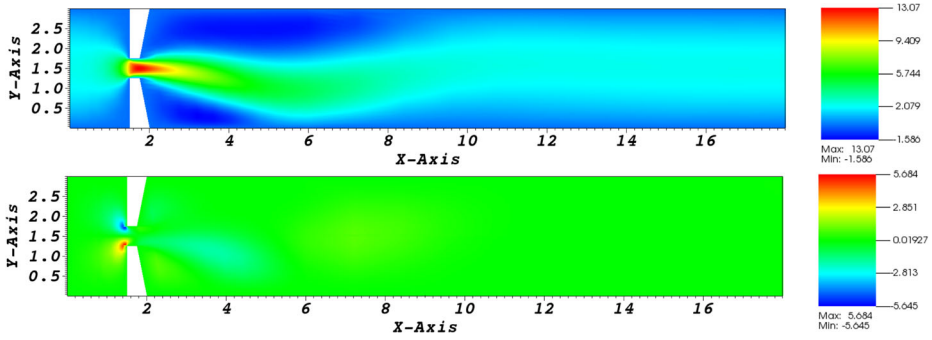

Fig. 5 Full order, steady-state solution in the geometry with straight walls and for $\nu = 0.15$: velocity in x-direction (top) and y-direction (bottom)

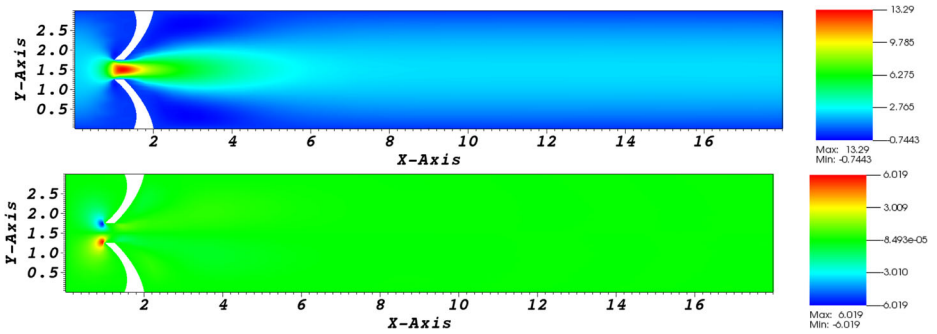

Fig. 6 Full order, steady-state solution in the geometry with curved walls and for $\nu = 0.2$: velocity in x-direction (top) and y-direction (bottom)

horizontal centerline at $y = 1.5$, while the jet of the stable solutions undergoes the *Coanda effect*.

In this investigation, we do not deal with recovering all bifurcation branches, but limit our attention to the stable branch of solutions with jets hugging the bottom wall. However, we remark that recovering all bifurcating solutions with model reduction methods is also possible, see, e.g., ([10]) and ([20]).

4.2.1 Generating Curved Geometries

The different curvatures are approximated by polynomials. Each curved wall is defined by a second order polynomial, interpolating three prescribed points. While the points at the domain boundary $y = 0$ and $y = 3$ are kept fixed, the inner points are moved towards $x = 0$ in order to create an increasing curvature. The tip of both narrowings and one intermediate point between the tip and the wall are prescribed, while the point where the wall and the narrowing meet remains constant. These three points define a quadratic polynomial, which is used to model the edge from the tips to upper and lower wall, respectively. This is a standard feature implemented in the PDE solver `Nektar++`.

The mesh remains topologically equivalent for each parametric configuration. This allows to easily map the mesh to the reference configuration via a plain pullback (see [9]

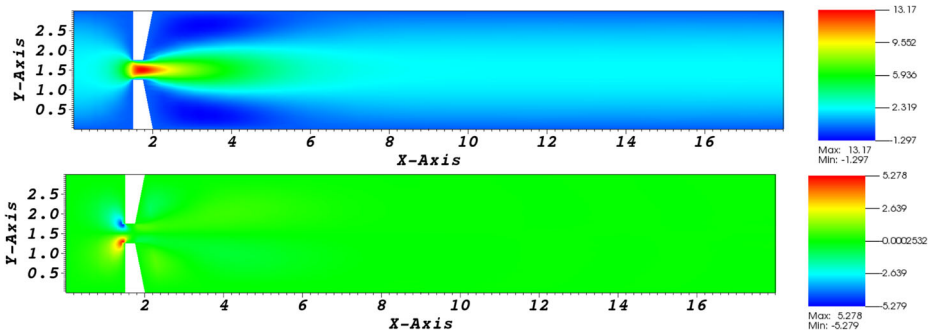

Fig. 7 Full order, steady-state solution in the geometry with straight walls and for $\nu = 0.2$: velocity in x-direction (top) and y-direction (bottom)

for a discussion of the plain pullback) acting directly on the degrees of freedom, i.e., the entries of the solution vector.

4.2.2 Numerical Results

The Leja points are computed as in (5) in each parameter direction and a tensorized grid of the two dimensional parameter domain is used. A grid rule has to be chosen that determines how the sequence of multiindices $(\nu^n)_{n\geq 1}$ has to be chosen that defines the multivariate points z_{ν^n}. Here, a downward closed set is formed in a canonical way by choosing $\nu^1 = (0,0)$, $\nu^2 = (1,0)$, $\nu^3 = (0,1)$, $\nu^4 = (2,0)$, $\nu^5 = (1,1)$, ..., i.e., increasing the sum $\nu_1^n + \nu_2^n$ only after all possible combinations of elements with the same sum $\nu_1^n + \nu_2^n$ have been added to $(\nu^n)_{n\geq 1}$. The chosen points are depicted in Fig. 8 as blue crosses.

The set Λ, which defines the sparse interpolation operator (4) is given by $\Lambda_n = \{\nu^i, i = 1, \ldots, n\}$. This defines a hierarchical sequence of index sets $\Lambda_1 \subset \Lambda_2 \subset \Lambda_3 \subset, \ldots, \Lambda_n$, which allows to reuse the computed snapshot solutions when updating the interpolation operator $I_{\Lambda_{n-1}}$ to I_{Λ_n}.

A second grid rule chooses the next point adaptively. Initialise with $\nu^1 = (0,0)$ and then choose the next grid point by considering all points which would leave the following index set in the hierarchy of index sets a downward closed set. These points are also called the set of neighbours in [6]. Compute the PDE solutions at these points and compute the relative error between the exact solution and the current approximation. Then add the grid point which has the maximum error. The chosen points are depicted in Fig. 8 as red circles.

The interpolation operators are computed up to dimension 41 and the maximum and mean L^2 errors in the velocity are computed for 72 reference solutions. Only the Leja-points without explicit symmetrization are used, since the first numerical test could show that the results of the point rules are similar. Figure 9 shows the relative error in the velocity for increasing ROM size of the sparse interpolation. Without adaptivity, a maximum error of less than 1% is reached at dimension 17 and a maximum error of less than 0.1% is reached at dimension 41. A mean error of less than 1% is reached at dimension 10 and a mean error

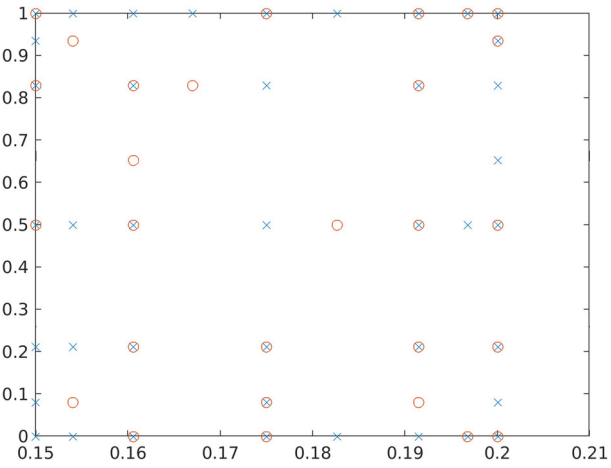

Fig. 8 Chosen Leja points plotted with the kinematic viscosity on the x-axis and a measure of the curvature on the y-axis, where the '1' refers to the maximum curvature. Blue crosses indicate no adaptivity and red circles are with adaptivity

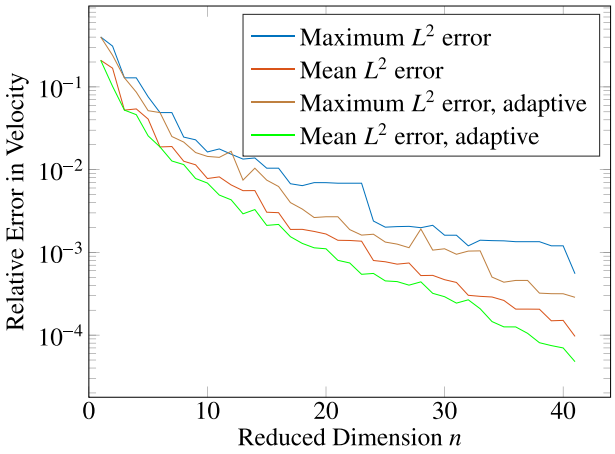

Fig. 9 Relative error for increasing sparse interpolation ROM dimension evaluated over a fine grid of 72 snapshots with and without adaptivity

of less than 0.1% is reached at dimension 24. With adaptivity, a maximum error of less than 1% is reached at dimension 13 and a maximum error of less than 0.1% is reached at dimension 31. A mean error of less than 1% is reached at dimension 9 and a mean error of less than 0.1% is reached at dimension 21. The error does not jump above these thresholds for higher dimensions. This indicates, that the sparse polynomial interpolation generates usable and reliable ROMs, that can be refined to higher accuracy as needed and there is some benefit gained from using adaptivity.

The same model has been investigated using a reduced-basis (RB) framework with the empirical interpolation method (EIM) in [12]. The reduced order model with $N = 20$ basis functions, showed an absolute error at the bifurcation point value of less than 0.01 at 46 parameter locations and less than 0.1 at 63 parameter locations. The chosen bifurcation point is the vertical velocity at the point (2, 1.5), which was used in [12] to access the accuracy. This means, that the RB method was not able to generate overall accurate methods in each of the 72 test points, in contrast to the sparse polynomial interpolation. To plot the RB approximation accuracy as in Fig. 9 would not add helpful information as the error was jumping up frequently when increasing the ROM dimension.

Another issue is that the EIM relies on the fast computation of a few matrix entries during the online phase. However, the spectral element *ansatz* functions have support over a whole spectral element, so this operation cannot be performed as fast as with a finite element method for example. The speed-up will thus not be as significant when using EIM. Also in the case of a finite-element or finite-volume discretization the sparse polynomial approach would likely perform better, since the sparse polynomial approach avoids the EIM completely. However, the gain in computational speed-up would not be as significant when comparing to a finite-element or finite-volume discretization.

A few points are worth highlighting:

Relevance of Results The sparse polynomial interpolation was able to compute accurate ROMs, which are stable when increasing the ROM size. While the first numerical example shows that the ROM sizes are not as small as for the RB method, the RB still needs to compute

a similar number of overall snapshots for the POD sample matrix. Taking this into account, both methods show a comparable performance. The curved boundaries in the second numerical example allow topologically equivalent mesh changes, but introduce a parametric nonlinearity. The sparse interpolation nevertheless produced accurate ROMs, where the RB largely failed. Parametric nonlinearities are inherently difficult for many ROM methods, but at least in the case of topologically equivalent meshes, the sparse polynomial interpolation could be a method of choice. Additionally, the sparse polynomial interpolation offers adaptivity in choosing the interpolation points with a heuristic error estimator, see [6].

Offline-Online Decomposition The offline-online decomposition separates the computations in two parts. The offline phase performs time-intensive computations such as the snapshot computations, while the online phase is quickly solves the ROM for many parameters of interest or in a real-time context. The offline-online splitting is also present in the sparse interpolation, since the snapshot solutions can be computed on a high-performance cluster (HPC), while the evaluation of the sparse operator for a parameter of interest can be done efficiently without a large computational effort. This is important for a wide applicability of the method and the sparse interpolation shares this property with the RB.

Run Time Both methods require the computation of the snapshot solutions. There is hardly any additional run time effort for the sparse interpolation, since the interpolation operators can be computed in a hierarchical way as long as the index sets are hierarchical. The RB on the other hand is much more involved. For the assembly of the reduced order systems a reduced trilinear form is computed in the incompressible Navier–Stokes case, which takes approximately as much time as computing the initial snapshot solutions in the shown examples. Additionally, the compute time for an EIM can be significant in the RB method.

Implementation The sparse interpolation can be implemented as outlined in [6] with a hierarchical computation. There are several choices for point rules and polynomials, but the implementational effort is light when compared to the reduced basis method. The RB for incompressible Navier–Stokes requires to compute the reduced operators of the affine form and the EIM requires to identify the degrees of freedom, which most significantly contribute to the system matrix. This is a significant implementation effort, which is not necessary for and has no counterpart in the sparse interpolation.

5 Conclusion and Outlook

The sparse polynomial interpolation generates comparable reduced order models (ROMs) to the reduced basis (RB) method in terms of accuracy and model size. In terms of applicability, some parametric nonlinearities of the geometry can be treated without altering the method, in particular if the mesh topology remains intact. Regarding the run time of the method and the ease of implementation, the sparse interpolation is even superior to the RB. The offline-online splitting is also present in the sparse interpolation, which allows to offload time-consuming snapshot computations to a high-performance cluster, while evaluating the ROM on nearly any machine. The literature on sparse interpolation offers a lot more than what is discussed here. Namely bounds on the approximation error and techniques for dealing with more complicated nonlinearities as well as adaptive choices of sample sets. Connecting these topics with the numerical models can be the topic of future research.

Acknowledgements We acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD, PI G. Rozza, and INdAM-GNCS 2019-2020 projects.

Funding Open access funding provided by Scuola Internazionale Superiore di Studi Avanzati - SISSA within the CRUI-CARE Agreement.

References

1. Benner, P., Ohlberger, M., Cohen, A., Willcox, K.: Model Reduction and Approximation. Society for Industrial And Applied Mathematics, Philadelphia (2017)
2. Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L.M. (eds.): Model Order Reduction: Volume 1: System- and Data-Driven Methods and Algorithms. De Gruyter, Berlin (2021)
3. Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L.M. (eds.): Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms. De Gruyter, Berlin (2021)
4. Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L.M. (eds.): Model Order Reduction: Volume 3: Applications. De Gruyter, Berlin (2021)
5. Boffi, D., Brezzi, F., Fortin, M.: Mixed Finite Element Methods and Applications. Springer Series in Computational Mathematics, vol. 44. Springer, Heidelberg (2013)
6. Chkifa, A., Cohen, A., Schwab, C.: High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. Found. Comput. Math. **14**, 601–633 (2014)
7. Chkifa, A., Cohen, A., Schwab, C.: Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. J. Math. Pures Appl. **103**, 400–428 (2015)
8. Cohen, A., DeVore, R., Schwab, C.: Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's. Anal. Appl. **9**, 11–47 (2011)
9. Cohen, A., Schwab, C., Zech, J.: Shape holomorphy of the stationary Navier–Stokes equations. SIAM J. Math. Anal. **50**, 1720–1752 (2018)
10. Herrero, H., Maday, Y., Pla, F.: RB (Reduced Basis) for RB (Rayleigh–Bénard). Comput. Methods Appl. Mech. Eng. **261–262**, 132–141 (2013)
11. Hess, M.W., Quaini, A., Rozza, G.: A spectral element reduced basis method for Navier–Stokes equations with geometric variations. In: Sherwin, S.J., Moxey, D., Peiró, J., Vincent, P.E., Schwab, C. (eds.) Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018, pp. 561–571. Springer, Cham (2020)
12. Hess, M.W., Quaini, A., Rozza, G.: Reduced basis model order reduction for Navier–Stokes equations in domains with walls of varying curvature. Int. J. Comput. Fluid Dyn. **34**, 119–126 (2020)
13. Hess, M., Alla, A., Quaini, A., Rozza, G., Gunzburger, M.: A localized reduced-order modeling approach for PDEs with bifurcating solutions. Comput. Methods Appl. Mech. Eng. **351**, 379–403 (2019)
14. Hess, M.W., Rozza, G.: A spectral element reduced basis method in parametric CFD. In: Radu, F.A., Kumar, K., Berre, I., Nordbotten, J.M., Pop, I.S. (eds.) Numerical mathematics and advanced applications ENUMATH 2017. Lecture Notes in Computational Science and Engineering, Vol. 126, pp. 693–701. Springer, Cham (2019)
15. Hesthaven, J., Rozza, G., Stamm, B.: Certified Reduced Basis Methods for Parametrized Partial Differential Equations. Springer, Cham (2015)
16. Holmes, P., Lumley, J.L., Berkooz, G., Rowley, C.W.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, Cambridge (2012)
17. Karniadakis, G., Sherwin, S.: Spectral/hp Element Methods for CFD . Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2005)

18. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: Model order reduction in fluid dynamics: Challenges and perspectives. In: Quarteroni, A., Rozza, G. (eds.) Reduced Order Methods for Modeling and Computational Reduction. Modeling, Simulation and Applications, vol. 9, pp. 235–273. Springer, Milano (2014)
19. Pichi, F., Ballarin, F., Rozza, G., Hesthaven, J.S.: An artificial neural network approach to bifurcating phenomena in computational fluid dynamics. arXiv:2109.10765 (2021)
20. Pintore, M., Pichi, F., Hess, M., Rozza, G., Canuto, C.: Efficient computation of bifurcation diagrams with a deflated approach to reduced basis spectral element method. Adv. Comput. Math. **47**, 1 (2021)
21. Quarteroni, A., Manzoni, A., Negri, F.: Reduced Basis Methods for Partial Differential Equations UNITEXT, vol. 92. Springer, Cham (2016)
22. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Springer, Berlin (1994)