SIAM/ASA J. Uncertainty Quantification
Vol. 10, No. 3, pp. 1101–1129

Stochastic Galerkin Methods for Linear Stability Analysis of Systems with Parametric Uncertainty*

Bedřich Sousedík† and Kookjin Lee‡

Abstract. We present a method for linear stability analysis of systems with parametric uncertainty formulated in the stochastic Galerkin framework. Specifically, we assume that for a model partial differential equation, the parameter is given in the form of generalized polynomial chaos expansion. The stability analysis leads to the solution of a stochastic eigenvalue problem, and we wish to characterize the rightmost eigenvalue. We focus, in particular, on problems with nonsymmetric matrix operators, for which the eigenvalue of interest may be a complex conjugate pair, and we develop methods for their efficient solution. These methods are based on inexact, line-search Newton iteration, which entails use of preconditioned GMRES. The method is applied to linear stability analysis of the Navier–Stokes equations with stochastic viscosity, its accuracy is compared to that of Monte Carlo and stochastic collocation, and the efficiency is illustrated by numerical experiments.

Key words. linear stability, eigenvalue analysis, uncertainty quantification, spectral stochastic finite element method, Navier–Stokes equation, preconditioning, stochastic Galerkin method

MSC codes. 35R60, 60H15, 65F15, 65L07, 65N22, 65N25

DOI. 10.1137/21M1415595

1. Introduction.

The identification of instability in large-scale dynamical systems is important in a number of applications such as fluid dynamics, epidemic models, pharmacokinetics, analysis of power systems and power grids, and quantum mechanics and plasma physics. A steady solution u is stable if, when in a transient simulation, it is introduced with a small perturbation as initial data and the simulation reverts to u, and it is unstable otherwise. This is of fundamental importance since unstable solutions may lead to inexplicable dynamic behavior. Linear stability analysis entails computing the rightmost eigenvalue of the Jacobian evaluated at u, and thus it leads to solution of, in general, large sparse generalized eigenvalue problems; see, e.g., [4, 5, 8, 13, 16, 27] and the references therein. Typically, a complex pair of rightmost eigenvalues leads to a Hopf bifurcation, and a real rightmost eigenvalue may lead to a pitchfork bifurcation. The analysis is further complicated if the parameters in the systems are functions of one or more random variables. This is quite common in many real-world applications, since the precise values of model coefficients or boundary conditions are often not known. A popular method for this type of problem is Monte Carlo, which is known for

* Received by the editors April 27, 2021; accepted for publication (in revised form) March 2, 2022; published electronically September 27, 2022.
https://doi.org/10.1137/21M1415595
Funding: This work was supported by the U. S. National Science Foundation under grant DMS-1913201.

† Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250 USA (sousedik@umbc.edu).
‡ The School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (klee263@asu.edu).

its robustness but also its slow convergence. In this study, we use spectral stochastic finite element methods [12, 17, 33, 34], with the main focus on the so-called stochastic Galerkin method, for the linear stability analysis of Navier–Stokes equation with stochastic viscosity. Specifically, we consider the parameterized viscosity given in the form of generalized polynomial chaos (gPC) expansion. In the first step, we apply the algorithms developed in [18, 30] (see also [24]) to find a gPC expansion of the solution of the Navier–Stokes equation. In the second step, we use the expansions of the solution and viscosity to set up a generalized eigenvalue problem with a nonsymmetric matrix operator, and in the assessment of linear stability of this problem we identify the gPC expansions of the rightmost eigenvalue. The main contribution in this study is the development of the stochastic Galerkin method for nonsymmetric eigenvalue problems. Our approach is based on inexact Newton iteration: the linear systems with Jacobian matrices are solved using GMRES, for which we also develop several preconditioners. The preconditioners are motivated by our prior work on (truncated) hierarchical preconditioning [32, 19]; see also [2]. For an overview of literature on solving eigenvalue problems in the context of spectral stochastic finite element methods we refer the reader to [1, 19, 29] and the references therein. Recently, Hakula and Laaksonen [15] studied crossing of eigenmodes in the stochastic parameter space, and Elman and Su [9] developed a low-rank inverse subspace iteration. However, to the best of our knowledge, there are only a few references addressing nonsymmetric stochastic eigenvalue problems: by Sarrouy, Dessombz, and Sinou [25, 26], though there is no discussion of efficient solution strategies, and also by Sonday et al. [28], who studied distribution of eigenvalues for the Jacobian in the context of the stochastic Galerkin method. Most recently, the authors with collaborators also compared surrogate learning strategies based on a sampling method, Gaussian process regression, and a neural network in [31].

A study of linear stability of Navier–Stokes equation under uncertainty was conducted by Elman and Silvester [7]. The study was based on a judiciously chosen perturbation of the state variable, and a stochastic collocation method was used to characterize the rightmost eigenvalue. Our approach here is different. We consider parametric uncertainty (of the viscosity), and the solution strategy is based on the stochastic Galerkin method. In fact, also the variant of the collocation method used here is based on the stochastic Galerkin projection (sometimes called a *nonintrusive stochastic Galerkin method* in the literature; see [33, Chapter 7] for a discussion). From this perspective, our study can be viewed as an extension of the setup from [5] to problems with viscosity given in the form of stochastic expansion and their efficient solution using the stochastic Galerkin method. However, more importantly, we illustrate that the inexact methods for stochastic eigenvalue problems proposed recently by Lee and Sousedík [19] can be also applied to problems with the nonsymmetric matrix operator.[1] This in general allows us to perform a linear stability analysis for other types of problems as well. We do not address eigenvalue crossing here, which is a somewhat delicate task for gPC-based techniques. We assume that the eigenvalue of interest is sufficiently separated from the rest of the spectrum and no crossing occurs. This is often the case for outliers and other eigenvalues that may be of interest in applications. A suitability of the algorithm we propose in this

[1]Specifically, the methods based on inexact Newton iteration, since in our experience the stochastic Rayleigh quotient and inverse iteration methods are less effective for nonsymmetric problems.

study can be assessed, e.g., by running first a low-fidelity (quasi-)Monte Carlo simulation. From our experience with the problem at hand, we can also note that this indicates that the rightmost eigenvalue remains relatively well separated from the rest of the spectrum, and it is less prone to switching, unlike the other eigenvalues, even with moderate values of coefficient of variation, which in turn allows its efficient characterization by a gPC expansion.

The rest of the paper is organized as follows. In section 2 we recall the basic elements of linear stability analysis and link it to an eigenvalue problem for a specific model given by the Navier–Stokes equation. In section 3 we introduce the stochastic eigenvalue problem. In section 4 we formulate the sampling methods and in section 5 the stochastic Galerkin method and inexact Newton iteration for its solution. In section 6 we apply the algorithms to linear stability analysis of Navier–Stokes equation with stochastic viscosity, and in section 7 we report the results of numerical experiments. Finally, in section 8 we summarize and conclude our work.

2. Linear stability and deterministic model problem.

Following [5], let us consider the dynamical system

$$Mu_t = f(u, \nu), \tag{2.1}$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a nonlinear mapping, $u \in \mathbb{R}^n$ is the state variable (velocity, pressure, temperature, deformation, etc.), $M \in \mathbb{R}^{n\times n}$ is the mass matrix in the finite element setting, and ν is a parameter. Let u denote the steady-state solution to (2.1), i.e., $u_t = 0$. We are interested in the *stability* of$u$: if a small perturbation $\delta(0)$ is introduced to u at time $t = 0$, does $\delta(t)$ grow with time, or does it decay? For a fixed value of ν, linear stability of the steady-state solution is determined by the spectrum of the eigenvalue problem

$$Jv = \lambda Mv, \tag{2.2}$$

where $J = \frac{\partial f}{\partial u}(u(\nu), \nu)$ is the Jacobian matrix of f evaluated at ν. The eigenvalues have a general form $\lambda = \alpha + i\beta$, where $\alpha = \mathrm{Re}\lambda$ and $\beta = \mathrm{Im}\lambda$. Then $e^{\lambda t} = e^{\alpha t}e^{i\beta t}$, and since $\left|e^{\lambda t}\right| = \left|e^{\alpha t}\right|$, there are in general two cases: if $\alpha < 0$, the perturbation decays with time, and if $\alpha > 0$, the perturbation grows. We refer the reader to, e.g., [4, 13] and the references therein for a detailed discussion. That is, if all the eigenvalues have strictly negative real part, then u is a stable steady solution, and if some eigenvalues have nonnegative real part, then$u$ is unstable. Therefore, a change of stability can be detected by monitoring the rightmost eigenvalues of (2.2). A steady-state solution may lose its stability in one of two ways: either the rightmost eigenvalue of (2.2) is real and passes through zero from negative to positive as ν varies, or (2.2) has a complex pair of rightmost eigenvalues and they cross the imaginary axis as ν varies, which leads to a Hopf bifurcation with the consequent birth of periodic solutions of (2.1).

Consider a special case of (2.1), the time-dependent Navier–Stokes equations governing viscous incompressible flow,

$$\begin{aligned} \vec{u}_t &= \nu\nabla^2\vec{u} - (\vec{u}\cdot\nabla)\,\vec{u} - \nabla p, \\ 0 &= \nabla\cdot\vec{u}, \end{aligned} \tag{2.3}$$

subject to appropriate boundary and initial conditions in a bounded physical domain$D$, where ν is the kinematic viscosity, $\vec{u}$ is the velocity, and p is the pressure. Properties of the flow are usually characterized by the Reynolds number

$$Re = \frac{UL}{\nu},$$

where U is the characteristic velocity and L is the characteristic length. For convenience, we will also sometimes refer to the Reynolds number instead of the viscosity. Mixed finite element discretization of (2.3) gives the following Jacobian and the mass matrix (see [5] and [8, Chapter 8] for more details):

$$J = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \qquad M = \begin{bmatrix} -G & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \tag{2.4}$$

where $n_x = n_u + n_p$ is the number of velocity and pressure degrees of freedom, respectively, $n_u > n_p$, $F \in \mathbb{R}^{n_u \times n_u}$ is nonsymmetric, $B \in \mathbb{R}^{n_p \times n_u}$ is the divergence matrix, and the velocity mass matrix $G^{n_u \times n_u}$ is symmetric positive definite. The matrices are sparse and n_x is typically large. We note that the mass matrix$M$ is singular, and (2.2) has an infinite eigenvalue. As suggested in [3], we replace the singular mass matrix$M$ with the nonsingular, shifted mass matrix

$$M_\sigma = \begin{bmatrix} -G & \sigma B^T \\ \sigma B & 0 \end{bmatrix}, \tag{2.5}$$

which is symmetric but in general indefinite, and it maps the infinite eigenvalues of (2.2) to σ^{-1} and leaves the finite ones unchanged. Then the generalized eigenvalue problem (2.2) can be transformed into an eigenvalue problem

$$Jv = \lambda M_\sigma v. \tag{2.6}$$

The flow is considered stable if $\mathrm{Re}\lambda < 0$, and we wish to detect the onset of instability, that is, to find when the rightmost eigenvalue crosses the imaginary axis. Efficient methods for finding the rightmost pair of complex eigenvalues of (2.2) (or (2.6)) were studied in [5]. Here, our goal is different. We consider parametric uncertainty in the sense that the parameter $\nu \equiv \nu(\xi)$, where ξ is a set of random variables and which is given by the so-called generalized polynomial chaos expansion. To this end, we first recast the eigenvalue problem (2.6) in the spectral stochastic finite element framework, then we show how to efficiently solve it, and finally we apply the stability analysis to the Navier–Stokes equation with stochastic viscosity.

3. Stochastic eigenvalue problem. Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a complete probability space, that is, Ω is the sample space with σ-algebra$\mathcal{F}$ and probability measure$\mathcal{P}$. We assume that the randomness in the mathematical model is induced by a vector $\xi : \Omega \mapsto \Gamma \subset \mathbb{R}^{m_\xi}$ of independent random variables $\xi_1(\omega), \ldots, \xi_{m_\xi}(\omega)$, where $\omega \in \Omega$. Let $\mathcal{B}(\Gamma)$ denote the Borel σ-algebra onΓ induced by ξ and μ the induced measure. The expected value of the product of measurable functions onΓ determines a Hilbert space $T_\Gamma \equiv L^2(\Gamma, \mathcal{B}(\Gamma), \mu)$ with inner product

$$\langle u, v\rangle = \mathbb{E}\left[uv\right] = \int_\Gamma u(\xi)v(\xi)d\mu(\xi), \tag{3.1}$$

where the symbol $\mathbb{E}$ denotes the mathematical expectation.

In computations we will use a finite-dimensional subspace $T_p \subset T_\Gamma$ spanned by a set of multivariate polynomials $\{\psi_\ell(\xi)\}$ that are orthonormal with respect to the density function μ, that is, $\mathbb{E}\left[\psi_k\psi_\ell\right] = \delta_{k\ell}$, and ψ_1 is constant. This will be referred to as the gPC basis [34]. The dimension of the spaceT_p depends on the polynomial degree. For polynomials of total degree$p$, the dimension is $n_\xi = \binom{m_\xi+p}{p}$.

Suppose that we are given an affine expansion of a matrix operator, which may correspond to the Jacobian matrix in (2.2), as

$$K(\xi) = \sum_{\ell=1}^{n_\nu} K_\ell\psi_\ell(\xi), \tag{3.2}$$

where each$K_\ell \in \mathbb{R}^{n_x\times n_x}$ is a deterministic matrix, and K_1 is the mean-value matrix $K_1 = \mathbb{E}\left[K(\xi)\right]$. The representation (3.2) is obtained from either Karhunen–Loève or, more generally, a stochastic expansion of an underlying random process; a specific example is provided in section 7.

We are interested in a solution of the following stochastic eigenvalue problem: find a specific eigenvalue $\lambda(\xi)$ and possibly also the corresponding eigenvector$v(\xi)$ which satisfy in$D$ almost surely (a.s.) the equation

$$K(\xi)v(\xi) = \lambda(\xi)M_\sigma v(\xi), \tag{3.3}$$

where $K(\xi) \in \mathbb{R}^{n_x\times n_x}$ is a nonsymmetric matrix operator, $M_\sigma\in \mathbb{R}^{n_x\times n_x}$ is a deterministic mass matrix, $\lambda(\xi) \in \mathbb{C}$ and $v(\xi) \in \mathbb{C}^{n_x}$ along with a normalization condition

$$(v(\xi))^* v(\xi) = \text{constant}, \tag{3.4}$$

which is further specified in section 5.

We will search for expansions of the eigenpair $(\lambda(\xi), v(\xi))$ in the form

$$\lambda(\xi) = \sum_{k=1}^{n_\xi} \lambda_k\psi_k(\xi), \quad v(\xi) = \sum_{k=1}^{n_\xi} v_k\psi_k(\xi), \tag{3.5}$$

where $\lambda_k \in \mathbb{C}$ and $v_k \in \mathbb{C}^{n_x}$ are coefficients corresponding to the basis$\{\psi_k\}$. We note that the series for $\lambda(\xi)$ in (3.5) converges for $n_\xi \to \infty$ in the space T_Γ under the assumption that the gPC polynomials provide its orthonormal basis and provided that $\lambda(\xi)$ has finite second moments; see, e.g., [10] or [33, Chapter 5]. Convergence analysis of this approximation for self-adjoint problems can be found in [1].

4. Monte Carlo and stochastic collocation. Both Monte Carlo and stochastic collocation are based on sampling. The coefficients are defined by a discrete projection

$$\lambda_k = \langle\lambda, \psi_k\rangle, \quad v_k = \langle v, \psi_k\rangle, \qquad k = 1, \ldots, n_\xi. \tag{4.1}$$

The evaluations of coefficients in (4.1) entail solving a set of independent deterministic eigenvalue problems at a set of sample points $\{\xi^{(q)}\}$, $q = 1, \ldots, n_{MC}$, or n_q,

$$K(\xi^{(q)})v(\xi^{(q)}) = \lambda\left(\xi^{(q)}\right) M_\sigma v\left(\xi^{(q)}\right). \tag{4.2}$$

In the Monte Carlo method, the sample points $\xi^{(q)}$, $q = 1, \ldots, n_{MC}$, are generated randomly following the distribution of the random variables ξ_i, $i = 1, \ldots, m_\xi$, and moments of solution are computed by ensemble averaging. In addition, the coefficients in (3.5) could be computed using Monte Carlo integration as[2]

$$\lambda_k = \frac{1}{n_{MC}} \sum_{q=1}^{n_{MC}} \lambda(\xi^{(q)})\psi_k\left(\xi^{(q)}\right), \qquad v_{mk} = \frac{1}{n_{MC}} \sum_{q=1}^{n_{MC}} v(x_m, \xi^{(q)})\psi_k(\xi^{(q)}).$$

For stochastic collocation, which is used here as the so-called nonintrusive stochastic Galerkin method, the sample points $\xi^{(q)}$, $q = 1, \ldots, n_q$, consist of a predetermined set of *collocation points*, and the coefficients λ_k and v_k in the expansions (3.5) are determined by evaluating (4.1) in the sense of (3.1) using numerical quadrature as

$$\lambda_k = \sum_{q=1}^{n_q} \lambda(\xi^{(q)})\psi_k(\xi^{(q)})w^{(q)}, \qquad v_{mk} = \sum_{q=1}^{n_q} v(x_m, \xi^{(q)})\psi_k(\xi^{(q)})w^{(q)},$$

where $\xi^{(q)}$ are the quadrature (collocation) points and $w^{(q)}$ are quadrature weights. Details of the rule we use in our numerical experiments are discussed in section 7, and we refer the reader to monograph [17] for a detailed discussion of quadrature rules.

5. Stochastic Galerkin method and Newton iteration. The stochastic Galerkin method is based on the projection

$$\langle Kv, \psi_k\rangle = \langle \lambda M_\sigma v, \psi_k\rangle, \qquad k = 1, \ldots, n_\xi, \tag{5.1}$$

$$\langle v^* v, \psi_k\rangle = \text{const}\cdot\delta_{k1}, \qquad k = 1, \ldots, n_\xi, \ \ \text{const} \in \mathbb{R}. \tag{5.2}$$

Let us introduce the notation

$$[H_\ell]_{jk} = h_{\ell,jk}, \quad h_{\ell,jk} \equiv \mathbb{E}\left[\psi_\ell\psi_j\psi_k\right], \qquad \ell = 1, \ldots, n_\nu, \quad j,k = 1, \ldots, n_\xi. \tag{5.3}$$

In order to formulate an efficient algorithm for eigenvalue problem (3.3) with nonsymmetric matrix operator using the stochastic Galerkin formulation, we introduce a separated representation of the eigenpair using real and imaginary parts,

$$v(\xi) = v_{\text{Re}}(\xi) + \mathrm{i}v_{\text{Im}}(\xi), \quad \lambda(\xi) = \lambda_{\text{Re}}(\xi) + \mathrm{i}\lambda_{\text{Im}}(\xi),$$

where $v_{\text{Re}}(\xi), v_{\text{Im}}(\xi) \in \mathbb{R}^{n_x}$ and $\lambda_{\text{Re}}(\xi), \lambda_{\text{Im}}(\xi) \in \mathbb{R}$. Then replacing $v(\xi)$ and $\lambda(\xi)$ in eigenvalue problem (3.3) results in

$$K(\xi)(v_{\text{Re}}(\xi) + \mathrm{i}v_{\text{Im}}(\xi)) = (\lambda_{\text{Re}}(\xi) + \mathrm{i}\lambda_{\text{Im}}(\xi))\, M_\sigma(v_{\text{Re}}(\xi) + \mathrm{i}v_{\text{Im}}(\xi)). \tag{5.4}$$

[2]In numerical experiments, we avoid this approximation of the gPC coefficients and directly work with the sampled quantities.

Expanding the terms in (5.4) and collecting the real and imaginary parts yields a system of equations that can be written in a separated representation as

$$\begin{bmatrix} K(\xi)v_{\mathrm{Re}}(\xi) \\ K(\xi)v_{\mathrm{Im}}(\xi) \end{bmatrix} = \begin{bmatrix} \lambda_{\mathrm{Re}}(\xi)M_\sigma & -\lambda_{\mathrm{Im}}(\xi)M_\sigma \\ \lambda_{\mathrm{Im}}(\xi)M_\sigma & \lambda_{\mathrm{Re}}(\xi)M_\sigma \end{bmatrix} \begin{bmatrix} v_{\mathrm{Re}}(\xi) \\ v_{\mathrm{Im}}(\xi) \end{bmatrix}, \tag{5.5}$$

and the normalization condition corresponding to the separated representation in (5.5) is taken as

$$v_{\mathrm{Re}}(\xi)^T v_{\mathrm{Re}}(\xi) = 1, \quad v_{\mathrm{Im}}(\xi)^T v_{\mathrm{Im}}(\xi) = 1. \tag{5.6}$$

Now, we seek expansions of type (3.5) for $v_{\mathrm{Re}}(\xi)$, $v_{\mathrm{Im}}(\xi)$, $\lambda_{\mathrm{Re}}(\xi)$, and $\lambda_{\mathrm{Im}}(\xi)$, that is,

$$v_s(\xi) = (\Psi(\xi)^T \otimes I_{n_x})\bar{v}_s, \quad \lambda_s(\xi) = \Psi(\xi)^T \bar{\lambda}_s, \quad s = \mathrm{Re}, \mathrm{Im}, \tag{5.7}$$

where the symbol $\otimes$ denotes the Kronecker product, $\Psi(\xi) = [\psi_1(\xi), \ldots, \psi_{n_\xi}(\xi)]^T$, $\bar{\lambda}_s = [\lambda_{s,1}, \ldots, \lambda_{s,n_\xi}]^T \in \mathbb{R}^{n_\xi}$, and $\bar{v}_s = [v_{s,1}^T, \ldots, v_{s,n_\xi}^T]^T \in \mathbb{R}^{n_x n_\xi}$ for $s = \mathrm{Re}, \mathrm{Im}$.

Let us consider expansions (5.7) as approximations to the solution of (5.5)–(5.6). Then we can write the residual of (5.5) as

$$\begin{aligned} &\widetilde{F}(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}}) \\ &\quad = \begin{bmatrix} \left(\Psi(\xi)^T \otimes K(\xi)\right)\bar{v}_{\mathrm{Re}} - \left(\bar{\lambda}_{\mathrm{Re}}^T \Psi(\xi)\Psi(\xi)^T \otimes M_\sigma\right)\bar{v}_{\mathrm{Re}} + \left(\bar{\lambda}_{\mathrm{Im}}^T \Psi(\xi)\Psi(\xi)^T \otimes M_\sigma\right)\bar{v}_{\mathrm{Im}} \\ \left(\Psi(\xi)^T \otimes K(\xi)\right)\bar{v}_{\mathrm{Im}} - \left(\bar{\lambda}_{\mathrm{Im}}^T \Psi(\xi)\Psi(\xi)^T \otimes M_\sigma\right)\bar{v}_{\mathrm{Re}} - \left(\bar{\lambda}_{\mathrm{Re}}^T \Psi(\xi)\Psi(\xi)^T \otimes M_\sigma\right)\bar{v}_{\mathrm{Im}} \end{bmatrix}, \end{aligned}$$

and the residual of (5.6) as

$$\widetilde{G}(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}) = \begin{bmatrix} \bar{v}_{\mathrm{Re}}^T \left(\Psi(\xi)\Psi(\xi)^T \otimes I_{n_x}\right)\bar{v}_{\mathrm{Re}} - 1 \\ \bar{v}_{\mathrm{Im}}^T \left(\Psi(\xi)\Psi(\xi)^T \otimes I_{n_x}\right)\bar{v}_{\mathrm{Im}} - 1 \end{bmatrix}.$$

Imposing the stochastic Galerkin orthogonality conditions (5.1) and (5.2) to $\widetilde{F}$ and $\widetilde{G}$, respectively, we get a system of nonlinear equations

$$r(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}}) = \begin{bmatrix} F(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}}) \\ G(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}) \end{bmatrix} = 0 \in \mathbb{R}^{2(n_x+1)n_\xi}, \tag{5.8}$$

where

$$\begin{aligned} &F(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}}) \\ &\quad = \begin{bmatrix} \mathbb{E}[\Psi\Psi^T \otimes K]\bar{v}_{\mathrm{Re}} - \mathbb{E}[(\bar{\lambda}_{\mathrm{Re}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma]\bar{v}_{\mathrm{Re}} + \mathbb{E}[(\bar{\lambda}_{\mathrm{Im}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma]\bar{v}_{\mathrm{Im}} \\ \mathbb{E}[\Psi\Psi^T \otimes K]\bar{v}_{\mathrm{Im}} - \mathbb{E}[(\bar{\lambda}_{\mathrm{Im}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma]\bar{v}_{\mathrm{Re}} - \mathbb{E}[(\bar{\lambda}_{\mathrm{Re}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma]\bar{v}_{\mathrm{Im}} \end{bmatrix} \end{aligned}$$

and

$$G(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}) = \begin{bmatrix} \mathbb{E}[\Psi \otimes ((\bar{v}_{\mathrm{Re}}{}^T(\Psi\Psi^T \otimes I_{n_x})\bar{v}_{\mathrm{Re}}) - 1)] \\ \mathbb{E}[\Psi \otimes ((\bar{v}_{\mathrm{Im}}{}^T(\Psi\Psi^T \otimes I_{n_x})\bar{v}_{\mathrm{Im}}) - 1)] \end{bmatrix}.$$

We will use Newton iteration to solve the system of nonlinear equations (5.8). The Jacobian matrix$DJ(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}})$ of (5.8) can be written as

$$DJ(\bar{v}_{\mathrm{Re}}, \bar{v}_{\mathrm{Im}}, \bar{\lambda}_{\mathrm{Re}}, \bar{\lambda}_{\mathrm{Im}}) = \begin{bmatrix} \frac{\partial F}{\partial \bar{v}_{\mathrm{Re}}} & \frac{\partial F}{\partial \bar{v}_{\mathrm{Im}}} & \frac{\partial F}{\partial \bar{\lambda}_{\mathrm{Re}}} & \frac{\partial F}{\partial \bar{\lambda}_{\mathrm{Im}}} \\ \frac{\partial G}{\partial \bar{v}_{\mathrm{Re}}} & \frac{\partial G}{\partial \bar{v}_{\mathrm{Im}}} & \frac{\partial G}{\partial \bar{\lambda}_{\mathrm{Re}}} & \frac{\partial G}{\partial \bar{\lambda}_{\mathrm{Im}}} \end{bmatrix} \in \mathbb{R}^{(2(n_x+1)n_\xi)\times(2(n_x+1)n_\xi)},$$

where

(5.9)
$$\frac{\partial F}{\partial \bar{v}_{\mathrm{Re}}} = \begin{bmatrix} \mathbb{E}[\Psi\Psi^T \otimes K] - \mathbb{E}[(\bar{\lambda}_{\mathrm{Re}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma] \\ -\mathbb{E}[(\bar{\lambda}_{\mathrm{Im}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma] \end{bmatrix}, \quad \frac{\partial F}{\partial \bar{\lambda}_{\mathrm{Re}}} = \begin{bmatrix} -\mathbb{E}[\Psi^T \otimes (\Psi\Psi^T \otimes M_\sigma)]\bar{v}_{\mathrm{Re}} \\ -\mathbb{E}[\Psi^T \otimes (\Psi\Psi^T \otimes M_\sigma)]\bar{v}_{\mathrm{Im}} \end{bmatrix},$$

(5.10)
$$\frac{\partial F}{\partial \bar{v}_{\mathrm{Im}}} = \begin{bmatrix} \mathbb{E}[(\bar{\lambda}_{\mathrm{Im}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma] \\ \mathbb{E}[\Psi\Psi^T \otimes K] - \mathbb{E}[(\bar{\lambda}_{\mathrm{Re}}{}^T\Psi)\Psi\Psi^T \otimes M_\sigma] \end{bmatrix}, \quad \frac{\partial F}{\partial \bar{\lambda}_{\mathrm{Im}}} = \begin{bmatrix} \mathbb{E}[\Psi^T \otimes (\Psi\Psi^T \otimes M_\sigma)]\bar{v}_{\mathrm{Im}} \\ -\mathbb{E}[\Psi^T \otimes (\Psi\Psi^T \otimes M_\sigma)]\bar{v}_{\mathrm{Re}} \end{bmatrix}$$

and

$$\frac{\partial G}{\partial \bar{v}_{\mathrm{Re}}} = \begin{bmatrix} 2\mathbb{E}[\Psi \otimes (\bar{v}_{\mathrm{Re}}^T\Psi\Psi^T \otimes I_{n_x})] \\ 0 \end{bmatrix}, \quad \frac{\partial G}{\partial \bar{\lambda}_{\mathrm{Re}}} = 0, \tag{5.11}$$

$$\frac{\partial G}{\partial \bar{v}_{\mathrm{Im}}} = \begin{bmatrix} 0 \\ 2\mathbb{E}[\Psi \otimes (\bar{v}_{\mathrm{Im}}^T\Psi\Psi^T \otimes I_{n_x})] \end{bmatrix}, \quad \frac{\partial G}{\partial \bar{\lambda}_{\mathrm{Im}}} = 0. \tag{5.12}$$

However, for convenience in the formulation of the preconditioners presented later, we formulate Newton iteration with rescaled Jacobian matrix$\widehat{DJ}(\bar{v}_{\mathrm{Re}}^{(n)}, \bar{v}_{\mathrm{Im}}^{(n)}, \bar{\lambda}_{\mathrm{Re}}^{(n)}, \bar{\lambda}_{\mathrm{Im}}^{(n)})$, which at step$n$ entails solving a linear system

$$\begin{bmatrix} \frac{\partial F^{(n)}}{\partial \bar{v}_{\mathrm{Re}}} & \frac{\partial F^{(n)}}{\partial \bar{v}_{\mathrm{Im}}} & \frac{\partial F^{(n)}}{\partial \bar{\lambda}_{\mathrm{Re}}} & \frac{\partial F^{(n)}}{\partial \bar{\lambda}_{\mathrm{Im}}} \\ -\frac{1}{2}\frac{\partial G^{(n)}}{\partial \bar{v}_{\mathrm{Re}}} & -\frac{1}{2}\frac{\partial G^{(n)}}{\partial \bar{v}_{\mathrm{Im}}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\bar{v}_{\mathrm{Re}} \\ \delta\bar{v}_{\mathrm{Im}} \\ \delta\bar{\lambda}_{\mathrm{Re}} \\ \delta\bar{\lambda}_{\mathrm{Im}} \end{bmatrix} = -\begin{bmatrix} F^{(n)} \\ -\frac{1}{2}G^{(n)} \end{bmatrix}, \tag{5.13}$$

followed by an update

$$\begin{bmatrix} \bar{v}_{\mathrm{Re}}^{(n+1)} \\ \bar{v}_{\mathrm{Im}}^{(n+1)} \\ \bar{\lambda}_{\mathrm{Re}}^{(n+1)} \\ \bar{\lambda}_{\mathrm{Im}}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \bar{v}_{\mathrm{Re}}^{(n)} \\ \bar{v}_{\mathrm{Im}}^{(n)} \\ \bar{\lambda}_{\mathrm{Re}}^{(n)} \\ \bar{\lambda}_{\mathrm{Im}}^{(n)} \end{bmatrix} + \begin{bmatrix} \delta\bar{v}_{\mathrm{Re}} \\ \delta\bar{v}_{\mathrm{Im}} \\ \delta\bar{\lambda}_{\mathrm{Re}} \\ \delta\bar{\lambda}_{\mathrm{Im}} \end{bmatrix}. \tag{5.14}$$

Remark 5.1. We used the rescaling of the Jacobian in [19] in order to symmetrize the matrix in (5.13); however, we note that here it is still in general nonsymmetric.

Linear systems (5.13) are solved inexactly using a preconditioned GMRES method. We refer the reader to Appendix A for details of the evaluation of the right-hand side and of the matrix-vector product, and to [18] for a discussion of GMRES in a related context.

5.1. Inexact Newton iteration. As in [19], we consider a line-search modification of this method following [23, Algorithm 11.4] in order to improve global convergence behavior of Newton iteration. Denoting

$$\bar{v}^{(n)} = [(\bar{v}_{\mathrm{Re}}^{(n)})^T, (\bar{v}_{\mathrm{Im}}^{(n)})^T]^T \quad \text{and} \quad \bar{\lambda}^{(n)} = [(\bar{\lambda}_{\mathrm{Re}}^{(n)})^T, (\bar{\lambda}_{\mathrm{Im}}^{(n)})^T]^T,$$

we define the merit function as the sum of squares,

$$f(\bar{v}^{(n)}, \bar{\lambda}^{(n)}) = \frac{1}{2}\|\widehat{r}(\bar{v}^{(n)}, \bar{\lambda}^{(n)})\|_2^2, \quad \text{where } \widehat{r}(\bar{v}^{(n)}, \bar{\lambda}^{(n)}) = \begin{bmatrix} F^{(n)} \\ -\frac{1}{2}G^{(n)} \end{bmatrix},$$

that is, $\widehat{r}(\bar{v}^{(n)}, \bar{\lambda}^{(n)})$ is the negative right-hand side of (5.13), i.e., it is a rescaled residual of (5.8), and we also denote

$$f_n = f(\bar{v}^{(n)}, \bar{\lambda}^{(n)}), \qquad \widehat{r}_n = \widehat{r}(\bar{v}^{(n)}, \bar{\lambda}^{(n)}), \qquad \widehat{DJ}_n = \widehat{DJ}(\bar{v}^{(n)}, \bar{\lambda}^{(n)}).$$

As the initial approximation of the solution, we use the eigenvectors and eigenvalues of the associated mean problem

$$K_1 \left([v_{\mathrm{Re}}^{(0)}]_1 + i[v_{\mathrm{Im}}^{(0)}]_1\right) = \left([\lambda_{\mathrm{Re}}^{(0)}]_1 + i[\lambda_{\mathrm{Im}}^{(0)}]_1\right) M_\sigma \left([v_{\mathrm{Re}}^{(0)}]_1 + i[v_{\mathrm{Im}}^{(0)}]_1\right), \tag{5.15}$$

concatenated by zeros, that is,

$$\bar{v}^{(0)} = \left[([v_{\mathrm{Re}}^{(0)}]_1)^T, 0, \ldots, ([v_{\mathrm{Im}}^{(0)}]_1)^T, 0, \ldots\right]^T,$$
$$\bar{\lambda}^{(0)} = \left[[\lambda_{\mathrm{Re}}^{(0)}]_1, 0, \ldots, [\lambda_{\mathrm{Im}}^{(0)}]_1, 0, \ldots\right]^T,$$

and the initial residual is

$$\widehat{r}_0 = \begin{bmatrix} F(\bar{v}^{(0)}, \bar{\lambda}^{(0)}) \\ -\frac{1}{2}G(\bar{v}^{(0)}) \end{bmatrix}.$$

The line-search Newton method is summarized in our setting as Algorithm 1, and the choice of parametersρ and$c$ in the numerical experiments is discussed in section 7.

The inexact iteration entails in each step a solution of the stochastic Galerkin linear system in line 4 of Algorithm 1 given by (5.13) using a Krylov subspace method. We use preconditioned GMRES with the adaptive stopping criteria,

$$\frac{\|\widehat{r}_n + \widehat{DJ}_n p_n\|_2}{\|\widehat{r}_n\|_2} < \tau \|\widehat{r}_{n-1}\|_2, \tag{5.16}$$

where $\tau = 10^{-1}$. The for-loop is terminated when the convergence check in line 12 is satisfied; in our numerical experiments we check if $\|\widehat{r}_n\|_2 < 10^{-10}$.

Algorithm 1 [23, Algorithm 11.4] Line-search Newton iteration

1: Given $\rho, c \in (0,1)$, set $\alpha^* = 1$.
2: Set $\bar{v}^{(0)}$ and $\bar{\lambda}^{(0)}$.
3: **for** $n = 0, 1, 2, \ldots$ **do**
4: $\widehat{DJ}_n p_n = -\widehat{r}_n$ (Solve inexactly to find the Newton update p_n.)
5: $\begin{bmatrix} \delta\bar{v}^{(n)} \\ \delta\bar{\lambda}^{(n)} \end{bmatrix} = p_n$
6: $\alpha_n = \alpha^*$
7: **while** $f(\bar{v}^{(n)} + \alpha_n \delta\bar{v}^{(n)}, \bar{\lambda}^{(n)} + \alpha_n \delta\bar{\lambda}^{(n)}) > f_n + c\,\alpha_n \nabla f_n^T p_n$ **do**
8: $\alpha_n \leftarrow \rho\,\alpha_n$
9: **end while**
10: $\bar{v}^{(n+1)} \leftarrow \bar{v}^{(n)} + \alpha_n \delta\bar{v}^{(n)}$
11: $\bar{\lambda}^{(n+1)} \leftarrow \bar{\lambda}^{(n)} + \alpha_n \delta\bar{\lambda}^{(n)}$
12: Check for convergence.
13: **end for**

Our implementation of the solvers is based on the so-called *matricized* formulation, in which we make use of isomorphism between $\mathbb{R}^{n_x n_\xi}$ and $\mathbb{R}^{n_x \times n_\xi}$ determined by the operators vec and mat: $\bar{v} = \mathrm{vec}(\bar{V})$, $\bar{V} = \mathrm{mat}(\bar{v})$, where $\bar{v} \in \mathbb{R}^{n_x n_\xi}$, $\bar{V} \in \mathbb{R}^{n_x \times n_\xi}$. The uppercase/lowercase notation is assumed throughout the paper, so $\bar{R} = \mathrm{mat}(\bar{r})$, etc. Specifically, we define the *matricized* coefficients of the eigenvector expansion

$$\bar{V} = \mathrm{mat}(\bar{v}) = \left[v_1, v_2, \ldots, v_{n_\xi}\right] \in \mathbb{R}^{n_x \times n_\xi}, \tag{5.17}$$

where the column k contains the coefficients associated with the basis functionψ_k. A detailed formulation of the GMRES in the matricized format can be found, e.g., in [18]. We only describe computation of the matrix-vector product (Appendix A), and in the next section we formulate several preconditioners.

5.2. Preconditioners for the Newton iteration. In order to allow for an efficient iterative solution of linear systems in line 4 of Algorithm 1 given by (5.13) using a Krylov subspace method, it is necessary to provide a preconditioner. In this section, we adapt the mean-based preconditioner and two of the constraint preconditioners from [19] to the formulation with separated real and complex parts, and we write them in the matricized format. The idea can be motivated as follows. The preconditioners are based on approximations of the blocks in (A.2). The mean-based preconditioner is inspired by the approximation

$$\begin{bmatrix} \overline{A} & 0 \\ 0 & \overline{S} \end{bmatrix},$$

where $\overline{A} \approx [A_{\mathrm{Re}}\ A_{\mathrm{Im}}]$ and the Schur complement $\overline{S} \approx -\frac{1}{2}[C_{\mathrm{Re}}\ C_{\mathrm{Im}}][A_{\mathrm{Re}}\ A_{\mathrm{Im}}]^{-1}[B_{\mathrm{Re}}\ B_{\mathrm{Im}}]$. The constraint preconditioners are based on the approximation

$$\begin{bmatrix} \overline{A} & \overline{B} \\ \overline{C} & 0 \end{bmatrix},$$

Algorithm 2 Mean-based preconditioner (MB)

The preconditioner $\mathcal{M}_{\mathrm{MB}} : \left(\bar{R}^{v_{\mathrm{Re}}}, \bar{R}^{v_{\mathrm{Im}}}, \bar{R}^{\lambda_{\mathrm{Re}}}, \bar{R}^{\lambda_{\mathrm{Im}}}\right) \longmapsto \left(\bar{V}^{v_{\mathrm{Re}}}, \bar{V}^{v_{\mathrm{Im}}}, \bar{V}^{\lambda_{\mathrm{Re}}}, \bar{V}^{\lambda_{\mathrm{Im}}}\right)$ is defined as

$$\mathcal{M}_{\mathrm{MB}} \begin{bmatrix} \bar{V}^{v_{\mathrm{Re}}} \\ \bar{V}^{v_{\mathrm{Im}}} \\ \bar{V}^{\lambda_{\mathrm{Re}}} \\ \bar{V}^{\lambda_{\mathrm{Im}}} \end{bmatrix} = \begin{bmatrix} \bar{R}^{v_{\mathrm{Re}}} \\ \bar{R}^{v_{\mathrm{Im}}} \\ \bar{R}^{\lambda_{\mathrm{Re}}} \\ \bar{R}^{\lambda_{\mathrm{Im}}} \end{bmatrix}. \tag{5.18}$$

Above

$$\mathcal{M}_{\mathrm{MB}} = \begin{bmatrix} \widetilde{A} & 0_{2n_x \times 2} \\ 0_{2\times 2n_x} & \begin{bmatrix} -w_{\mathrm{Re}}^{(0)T} & 0_{n_x\times 1} \\ 0_{n_x\times 1} & -w_{\mathrm{Im}}^{(0)T} \end{bmatrix} \widetilde{A}^{-1} \begin{bmatrix} -M_\sigma w_{\mathrm{Re}}^{(0)} & M_\sigma w_{\mathrm{Im}}^{(0)} \\ -M_\sigma w_{\mathrm{Im}}^{(0)} & -M_\sigma w_{\mathrm{Re}}^{(0)} \end{bmatrix} \end{bmatrix}, \tag{5.19}$$

where $w_{\mathrm{Re}}^{(0)}$ and $w_{\mathrm{Im}}^{(0)}$ are the real and imaginary components of eigenvector w of the stencil (K_1, M_σ) with corresponding eigenvalue $\mu = \mu_{\mathrm{Re}} + \mathrm{i}\mu_{\mathrm{Im}}$, cf. (5.15), and

$$\widetilde{A} = \begin{bmatrix} K_1 - \epsilon_{\mathrm{Re}}\mu_{\mathrm{Re}} M_\sigma & \epsilon_{\mathrm{Im}}\mu_{\mathrm{Im}} M_\sigma \\ -\epsilon_{\mathrm{Im}}\mu_{\mathrm{Im}} M_\sigma & K_1 - \epsilon_{\mathrm{Re}}\mu_{\mathrm{Re}} M_\sigma \end{bmatrix}, \tag{5.20}$$

with constants ϵ_{Re}, ϵ_{Im} further specified in the numerical experiments section.

where $\overline{B} \approx [B_{\mathrm{Re}}\ B_{\mathrm{Im}}]$ and $\overline{C} \approx -\frac{1}{2}[C_{\mathrm{Re}}\ C_{\mathrm{Im}}]$. Next, considering the truncation of the series in (A.3)–(A.8) to the very first term, we get

$$\overline{A} \approx I_{n_\xi} \otimes \begin{bmatrix} K_1 - \lambda_{\mathrm{Re},1} M_\sigma & \lambda_{\mathrm{Im},1} M_\sigma \\ -\lambda_{\mathrm{Im},1} M_\sigma & K_1 - \lambda_{\mathrm{Re},1} M_\sigma \end{bmatrix}$$

(see left columns in (5.9)−(5.10) and (A.3)−(A.4)),

$$\overline{B} \approx I_{n_\xi} \otimes \begin{bmatrix} -M_\sigma v_{\mathrm{Re},1} & M_\sigma v_{\mathrm{Im},1} \\ -M_\sigma v_{\mathrm{Im},1} & -M_\sigma v_{\mathrm{Re},1} \end{bmatrix}$$ (see right columns in (5.9)−(5.10) and (A.5)−(A.6)),

$$\overline{C} \approx I_{n_\xi} \otimes \begin{bmatrix} -v_{\mathrm{Re},1}^T & 0_{n_x\times 1} \\ 0_{n_x\times 1} & -v_{\mathrm{Im},1}^T \end{bmatrix}$$ (see (5.11)−(5.12) and (A.7)−(A.8)).

The precise formulations are listed for the mean-based preconditioner as Algorithm 2 and for the constraint mean-based preconditioner as Algorithm 3. Finally, the constraint hierarchical Gauss–Seidel preconditioner is listed as Algorithm 4. It can be viewed as an extension of Algorithm 3, because the solves with stochastic Galerkin matrices (5.22) are used also in this preconditioner, but in addition the right-hand sides are updated using an idea inspired by Gauss–Seidel method in a for-loop over the degree of the gPC basis. Moreover, as proposed in [32], the matrix-vector multiplications, used in the setup of the right-hand sides, can be truncated in the sense that in the summations, $t = 1, \ldots, n_\xi$ is replaced by $t \in \mathcal{I}_t$, where $\mathcal{I}_t = \{1, \ldots, n_t\}$ with $n_t = \binom{m_\xi + p_t}{p_t}$ for some $p_t \le p$, and in particular with $p_t = 0$ the chGS preconditioner (Algorithm 4) reduces to the cMB preconditioner (Algorithm 3). We also note that, since the initial guess is zero in Algorithm 4, the multiplications by$\mathcal{F}_1$ and $\mathcal{F}_{d+1}$ vanish from (5.23)–(5.24).

Algorithm 3 Constraint mean-based preconditioner (cMB)

The preconditioner $\mathcal{M}_{\text{cMB}} : \left(\bar{R}^{v_{\text{Re}}}, \bar{R}^{v_{\text{Im}}}, \bar{R}^{\lambda_{\text{Re}}}, \bar{R}^{\lambda_{\text{Im}}}\right) \longmapsto \left(\bar{V}^{v_{\text{Re}}}, \bar{V}^{v_{\text{Im}}}, \bar{V}^{\lambda_{\text{Re}}}, \bar{V}^{\lambda_{\text{Im}}}\right)$ is defined as

$$\mathcal{M}_{\text{cMB}} \begin{bmatrix} \bar{V}^{v_{\text{Re}}} \\ \bar{V}^{v_{\text{Im}}} \\ \bar{V}^{\lambda_{\text{Re}}} \\ \bar{V}^{\lambda_{\text{Im}}} \end{bmatrix} = \begin{bmatrix} \bar{R}^{v_{\text{Re}}} \\ \bar{R}^{v_{\text{Im}}} \\ \bar{R}^{\lambda_{\text{Re}}} \\ \bar{R}^{\lambda_{\text{Im}}} \end{bmatrix}. \tag{5.21}$$

Above

$$\mathcal{M}_{\text{cMB}} = \begin{bmatrix} \widetilde{A} & & -M_\sigma w_{\text{Re}}^{(0)} & M_\sigma w_{\text{Im}}^{(0)} \\ & & -M_\sigma w_{\text{Im}}^{(0)} & -M_\sigma w_{\text{Re}}^{(0)} \\ -w_{\text{Re}}^{(0)T} & 0_{n_x \times 1} & 0_{2\times 2} & \\ 0_{n_x \times 1} & -w_{\text{Im}}^{(0)T} & & \end{bmatrix}, \tag{5.22}$$

with $w_{\text{Re}}^{(0)}$, $w_{\text{Im}}^{(0)}$, and $\widetilde{A}$ set as in Algorithm 2.

5.2.1. Updating the constraint preconditioner. It is also possible to update the application of the constraint mean-based preconditioner in order to incorporate the latest approximations of the eigenvector mean coefficients represented by the vectors $w_{\text{Re/Im}}^{(n)}$ in (5.22). Suppose the inverse of the matrix $\mathcal{M}_{\text{cMB}}$ from (5.22) for $n = 0$ is available, e.g., as the LU decomposition. That is, we have the preconditioner for the initial step of the Newton iteration, and let us denote its application by X^{-1}. Specifically, $X^{-1} = U^{-1}L^{-1}$, where L and U are the factors of $\mathcal{M}_{\text{cMB}}$. Next, let us consider two matrices Y and Z such that

$$YZ^T = \begin{bmatrix} 0_{2n_x \times 2n_x} & & -M_\sigma \Delta w_{\text{Re}}^{(n)} & M_\sigma \Delta w_{\text{Im}}^{(n)} \\ & & -M_\sigma \Delta w_{\text{Im}}^{(n)} & -M_\sigma \Delta w_{\text{Re}}^{(n)} \\ -\Delta w_{\text{Re}}^{(n)} & 0_{n_x \times 1} & 0_{2\times 2} & \\ 0_{n_x \times 1} & -\Delta w_{\text{Im}}^{(n)} & & \end{bmatrix}, \tag{5.25}$$

where

$$\Delta w_{\text{Re}}^{(n)} = w_{\text{Re}}^{(n)} - w_{\text{Re}}^{(0)}, \qquad \Delta w_{\text{Im}}^{(n)} = w_{\text{Im}}^{(n)} - w_{\text{Im}}^{(0)}.$$

Specifically, YZ^T is the rank 4 update of the preconditioner at step n of the Newton iteration, and the matrices Y and Z can be computed using a sparse SVD, which is available, e.g., in [20]. In implementation, using MATLAB notation with $YZ^T = \texttt{YZt}$, we compute $[\texttt{U}, \texttt{S}, \texttt{V}] = \texttt{svds}(\texttt{YZt}, 4)$ and set $Y = \texttt{U} * \texttt{S}$, $Z^T = \texttt{V}'$. Finally, a solve $\mathcal{M}_{\text{cMB}} v = u$ at step n of Newton iteration may be facilitated using the Sherman–Morrison–Woodbury formula (see, e.g., [14], or [22, section 3.8]) as

$$v = \left(X + YZ^T\right)^{-1} u = \left(X^{-1} - X^{-1}Y\left(I + Z^T X^{-1} Y\right)^{-1} Z^T X^{-1}\right) u.$$

Algorithm 4 Constraint hierarchical Gauss–Seidel preconditioner (chGS)

The preconditioner $M_{\text{chGS}} : \left(\bar{R}^{v_{\text{Re}}}, \bar{R}^{v_{\text{Im}}}, \bar{R}^{\lambda_{\text{Re}}}, \bar{R}^{\lambda_{\text{Im}}}\right) \longmapsto \left(\bar{V}^{v_{\text{Re}}}, \bar{V}^{v_{\text{Im}}}, \bar{V}^{\lambda_{\text{Re}}}, \bar{V}^{\lambda_{\text{Im}}}\right)$ is defined as follows.

1: Set the initial solution $\left(\bar{V}^{v_{\text{Re}}}, \bar{V}^{v_{\text{Im}}}, \bar{V}^{\lambda_{\text{Re}}}, \bar{V}^{\lambda_{\text{Im}}}\right)$ to zero and update as:

2: Solve

$$\mathcal{M}_{\text{cMB}} \begin{pmatrix} V_1^{v_{\text{Re}}} \\ V_1^{v_{\text{Im}}} \\ V_1^{\lambda_{\text{Re}}} \\ V_1^{\lambda_{\text{Im}}} \end{pmatrix} = \begin{bmatrix} R_1^{v_{\text{Re}}} \\ R_1^{v_{\text{Im}}} \\ R_1^{\lambda_{\text{Re}}} \\ R_1^{\lambda_{\text{Im}}} \end{bmatrix} - \mathcal{F}_1 \begin{pmatrix} V_{(2:n_\xi)}^{v_{\text{Re}}} \\ V_{(2:n_\xi)}^{v_{\text{Im}}} \\ V_{(2:n_\xi)}^{\lambda_{\text{Re}}} \\ V_{(2:n_\xi)}^{\lambda_{\text{Im}}} \end{pmatrix}, \tag{5.23}$$

where $\mathcal{M}_{\text{cMB}}$ is set as in Algorithm 3, and

$$\mathcal{F}_1 \begin{pmatrix} V_1^{v_{\text{Re}}} \\ V_1^{v_{\text{Im}}} \\ V_1^{\lambda_{\text{Re}}} \\ V_1^{\lambda_{\text{Im}}} \end{pmatrix} = \sum_{t \in \mathcal{I}_t} \begin{bmatrix} \mathcal{F}_1^A \left[h_{t,(2:n_\xi)(1)}\right] \\ \mathcal{F}_1^B \left[h_{t,(2:n_\xi)(1)}\right] \\ -(v_{\text{Re},t}^{(n)})^T V_{(2:n_\xi)}^{v_{\text{Re}}} \left[h_{t,(2:n_\xi)(1)}\right] \\ -(v_{\text{Im},t}^{(n)})^T V_{(2:n_\xi)}^{v_{\text{Im}}} \left[h_{t,(2:n_\xi)(1)}\right] \end{bmatrix},$$

$$\mathcal{F}_1^A = \left(\left(K_t - \lambda_{\text{Re},t}^{(n)} M_\sigma\right) V_{(2:n_\xi)}^{v_{\text{Re}}} + \lambda_{\text{Im},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{v_{\text{Im}}} - v_{\text{Re},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{\lambda_{\text{Re}}} + v_{\text{Im},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{\lambda_{\text{Im}}}\right),$$

$$\mathcal{F}_1^B = \left(\left(K_t - \lambda_{\text{Re},t}^{(n)} M_\sigma\right) V_{(2:n_\xi)}^{v_{\text{Im}}} - \lambda_{\text{Im},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{v_{\text{Re}}} - v_{\text{Re},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{\lambda_{\text{Im}}} - v_{\text{Im},t}^{(n)} M_\sigma V_{(2:n_\xi)}^{\lambda_{\text{Re}}}\right),$$

and $v_{\text{Re},t}^{(n)}$, $v_{\text{Im},t}^{(n)}$, the tth gPC coefficients of eigenvector $v^{(n)}$ at step n of Algorithm 1.

3: **for** $d = 1, \ldots, p-1$ **do**

4: Set $\ell = (n_\ell + 1 : n_u)$, where $n_\ell = \binom{n_\xi + d - 1}{d-1}$ and $n_u = \binom{n_\xi + d}{d}$.

5: Solve

$$\mathcal{M}_{\text{cMB}} \begin{pmatrix} V_{(\ell)}^{v_{\text{Re}}} \\ V_{(\ell)}^{v_{\text{Im}}} \\ V_{(\ell)}^{\lambda_{\text{Re}}} \\ V_{(\ell)}^{\lambda_{\text{Im}}} \end{pmatrix} = \begin{bmatrix} R_{(\ell)}^{v_{\text{Re}}} \\ R_{(\ell)}^{v_{\text{Im}}} \\ R_{(\ell)}^{\lambda_{\text{Re}}} \\ R_{(\ell)}^{\lambda_{\text{Im}}} \end{bmatrix} - \mathcal{E}_{d+1} \begin{pmatrix} V_{(1:n_\ell)}^{v_{\text{Re}}} \\ V_{(1:n_\ell)}^{v_{\text{Im}}} \\ V_{(1:n_\ell)}^{\lambda_{\text{Re}}} \\ V_{(1:n_\ell)}^{\lambda_{\text{Im}}} \end{pmatrix} - \mathcal{F}_{d+1} \begin{pmatrix} V_{(n_u+1:n_\xi)}^{v_{\text{Re}}} \\ V_{(n_u+1:n_\xi)}^{v_{\text{Im}}} \\ V_{(n_u+1:n_\xi)}^{\lambda_{\text{Re}}} \\ V_{(n_u+1:n_\xi)}^{\lambda_{\text{Im}}} \end{pmatrix}, \tag{5.24}$$

where

$$\mathcal{E}_{d+1} \begin{pmatrix} V_{(1:n_\ell)}^{v_{\text{Re}}} \\ V_{(1:n_\ell)}^{v_{\text{Im}}} \\ V_{(1:n_\ell)}^{\lambda_{\text{Re}}} \\ V_{(1:n_\ell)}^{\lambda_{\text{Im}}} \end{pmatrix} = \sum_{t \in \mathcal{I}_t} \begin{bmatrix} \mathcal{E}_{d+1}^A \left[h_{t,(1:n_\ell)(\ell)}\right] \\ \mathcal{E}_{d+1}^B \left[h_{t,(1:n_\ell)(\ell)}\right] \\ -(v_{\text{Re},t}^{(n)})^T V_{(1:n_\ell)}^{v_{\text{Re}}} \left[h_{t,(1:n_\ell)(\ell)}\right] \\ -(v_{\text{Im},t}^{(n)})^T V_{(1:n_\ell)}^{v_{\text{Im}}} \left[h_{t,(1:n_\ell)(\ell)}\right] \end{bmatrix},$$

Algorithm 4 Constraint hierarchical Gauss–Seidel preconditioner (chGS), continued

$$\mathcal{F}_{d+1}\begin{pmatrix} V^{v_{\mathrm{Re}}}_{(n_u+1:n_\xi)} \\ V^{v_{\mathrm{Im}}}_{(n_u+1:n_\xi)} \\ V^{\lambda_{\mathrm{Re}}}_{(n_u+1:n_\xi)} \\ V^{\lambda_{\mathrm{Im}}}_{(n_u+1:n_\xi)} \end{pmatrix} = \sum_{t\in\mathcal{I}_t} \begin{bmatrix} \mathcal{F}^A_{d+1}\left[h_{t,(n_u+1:n_\xi)(\ell)}\right] \\ \mathcal{F}^B_{d+1}\left[h_{t,(n_u+1:n_\xi)(\ell)}\right] \\ -(v^{(n)}_{\mathrm{Re},t})^T V^{v_{\mathrm{Re}}}_{(n_u+1:n_\xi)}\left[h_{t,(n_u+1:n_\xi)(\ell)}\right] \\ -(v^{(n)}_{\mathrm{Im},t})^T V^{v_{\mathrm{Im}}}_{(n_u+1:n_\xi)}\left[h_{t,(n_u+1:n_\xi)(\ell)}\right] \end{bmatrix},$$

$$\mathcal{E}^A_{d+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Re}}}_{(1:n_\ell)} + \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Im}}}_{(1:n_\ell)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(1:n_\ell)} + v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(1:n_\ell)}\right),$$

$$\mathcal{E}^B_{d+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Im}}}_{(1:n_\ell)} - \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Re}}}_{(1:n_\ell)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(1:n_\ell)} - v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(1:n_\ell)}\right),$$

$$\mathcal{F}^A_{d+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Re}}}_{(n_u+1:n_\xi)} + \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Im}}}_{(n_u+1:n_\xi)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(n_u+1:n_\xi)} + v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(n_u+1:n_\xi)}\right),$$

$$\mathcal{F}^B_{d+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Im}}}_{(n_u+1:n_\xi)} - \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Re}}}_{(n_u+1:n_\xi)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(n_u+1:n_\xi)} - v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(n_u+1:n_\xi)}\right).$$

6: **end for**
7: Set $\ell = (n_u + 1 : n_\xi)$.
8: Solve

$$\mathcal{M}_{\mathrm{cMB}}\begin{pmatrix} V^{v_{\mathrm{Re}}}_{(\ell)} \\ V^{v_{\mathrm{Im}}}_{(\ell)} \\ V^{\lambda_{\mathrm{Re}}}_{(\ell)} \\ V^{\lambda_{\mathrm{Im}}}_{(\ell)} \end{pmatrix} = \begin{bmatrix} R^{v_{\mathrm{Re}}}_{(\ell)} \\ R^{v_{\mathrm{Im}}}_{(\ell)} \\ R^{\lambda_{\mathrm{Re}}}_{(\ell)} \\ R^{\lambda_{\mathrm{Im}}}_{(\ell)} \end{bmatrix} - \mathcal{E}_{p+1}\begin{pmatrix} V^{v_{\mathrm{Re}}}_{(1:n_u)} \\ V^{v_{\mathrm{Im}}}_{(1:n_u)} \\ V^{\lambda_{\mathrm{Re}}}_{(1:n_u)} \\ V^{\lambda_{\mathrm{Im}}}_{(1:n_u)} \end{pmatrix},$$

where

$$\mathcal{E}_{p+1}\begin{pmatrix} V^{v_{\mathrm{Re}}}_{(1:n_u)} \\ V^{v_{\mathrm{Im}}}_{(1:n_u)} \\ V^{\lambda_{\mathrm{Re}}}_{(1:n_u)} \\ V^{\lambda_{\mathrm{Im}}}_{(1:n_u)} \end{pmatrix} = \sum_{t\in\mathcal{I}_t} \begin{bmatrix} \mathcal{E}^A_{p+1}\left[h_{t,(1:n_u)(\ell)}\right] \\ \mathcal{E}^B_{p+1}\left[h_{t,(1:n_u)(\ell)}\right] \\ -(v^{(n)}_{\mathrm{Re},t})^T V^{v_{\mathrm{Re}}}_{(1:n_u)}\left[h_{t,(1:n_u)(\ell)}\right] \\ -(v^{(n)}_{\mathrm{Im},t})^T V^{v_{\mathrm{Im}}}_{(1:n_u)}\left[h_{t,(1:n_u)(\ell)}\right] \end{bmatrix},$$

$$\mathcal{E}^A_{p+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Re}}}_{(1:n_u)} + \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Im}}}_{(1:n_u)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(1:n_u)} + v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(1:n_u)}\right),$$

$$\mathcal{E}^B_{p+1} = \left(\left(K_t - \lambda^{(n)}_{\mathrm{Re},t} M_\sigma\right) V^{v_{\mathrm{Im}}}_{(1:n_u)} - \lambda^{(n)}_{\mathrm{Im},t} M_\sigma V^{v_{\mathrm{Re}}}_{(1:n_u)} - v^{(n)}_{\mathrm{Re},t} M_\sigma V^{\lambda_{\mathrm{Im}}}_{(1:n_u)} - v^{(n)}_{\mathrm{Im},t} M_\sigma V^{\lambda_{\mathrm{Re}}}_{(1:n_u)}\right).$$

9: **for** $d = p-1, \ldots, 1$ **do**
10: Set $\ell = (n_\ell + 1 : n_u)$, where $n_\ell = \binom{n_\xi+d-1}{d-1}$ and $n_u = \binom{n_\xi+d}{d}$.
11: Solve (5.24).
12: **end for**
13: Solve (5.23).

6. Bifurcation analysis of Navier–Stokes equations with stochastic viscosity. Here, we follow the setup from [30] and assume that the viscosityν is given by a stochastic expansion

$$\nu(x,\xi) = \sum_{\ell=1}^{n_\nu} \nu_\ell(x)\,\psi_\ell(\xi), \tag{6.1}$$

where $\{\nu_\ell(x)\}$ is a set of given deterministic spatial functions. We note that there are several possible interpretations of such a setup [24, 30]. Assuming fixed geometry, the stochastic viscosity is equivalent to the Reynolds number being stochastic and, for example, to a scenario when the volume of fluid moving into a channel is uncertain. Consider the discretization of (2.3) by a div-stable mixed finite element method, and let the bases for the velocity and pressure spaces be denoted by $\{\phi_i\}_{i=1}^{n_u}$ and $\{\varphi_j\}_{i=1}^{n_p}$, respectively. We further assume that we have a discrete approximation of the steady-state solution of the stochastic counterpart of (2.3), given as[3]

$$\begin{aligned}
\vec{u}(x,\xi) &\approx \sum_{k=1}^{n_\xi}\sum_{i=1}^{n_u} u_{ik}\phi_i(x)\psi_k(\xi) = \sum_{k=1}^{n_\xi} \vec{u}_k(x)\psi_k(\xi),\\
p(x,\xi) &\approx \sum_{k=1}^{n_\xi}\sum_{j=1}^{n_p} p_{jk}\varphi_j(x)\psi_k(\xi) = \sum_{k=1}^{n_\xi} p_k(x)\psi_k(\xi).
\end{aligned}$$

We are interested in a stochastic counterpart of the generalized eigenvalue problem (2.2), which we write as

$$\mathcal{J}(\xi)v = \lambda M_\sigma v, \tag{6.2}$$

whereM_σ is the deterministic (shifted) mass matrix from (2.5), and $\mathcal{J}(\xi)$ is the stochastic Jacobian matrix operator given by the stochastic expansion

$$\mathcal{J}(\xi) = \sum_{\ell=1}^{\widehat{n}} \mathcal{J}_\ell\psi_\ell(\xi). \tag{6.3}$$

The expansion is built from matrices $\mathcal{J}_\ell \in \mathbb{R}^{n_x\times n_x}$, $\ell = 1,\dots,\widehat{n}$, such that

$$\mathcal{J}_1 = \begin{bmatrix} F_1 & B^T \\ B & 0 \end{bmatrix}, \qquad \mathcal{J}_\ell = \begin{bmatrix} F_\ell & 0 \\ 0 & 0 \end{bmatrix}, \qquad \ell = 2,\dots,\widehat{n},$$

where $\widehat{n} = \max(n_\nu, n_\xi)$, and F_ℓ is a sum of the vector-Laplacian matrixA_ℓ, the vector-convection matrixN_ℓ, and the Newton derivative matrix W_ℓ,

$$F_\ell = A_\ell + N_\ell + W_\ell,$$

[3]Techniques for computing these approximations were studied in [18, 30].

where

$$A_\ell = [a_{\ell,ab}]\,, \qquad a_{\ell,ab} = \int_D \nu_\ell(x)\,\nabla\phi_b : \nabla\phi_a, \qquad \ell = 1,\dots,n_\nu,$$
$$N_\ell = [n_{\ell,ab}]\,, \qquad n_{\ell,ab} = \int_D (\vec{u}_\ell \cdot \nabla\phi_b)\cdot\phi_a, \qquad \ell = 1,\dots,n_\xi,$$
$$W_\ell = [w_{\ell,ab}]\,, \qquad w_{\ell,ab} = \int_D (\phi_b \cdot \nabla\vec{u}_\ell)\cdot\phi_a, \qquad \ell = 1,\dots,n_\xi,$$

and if $n_\nu > n_\xi$, we set $N_\ell = W_\ell = 0$ for $\ell = n_{\xi+1},\dots,n_\nu$, and if $n_\nu < n_\xi$, we set $A_\ell = 0$ for $\ell = n_{\nu+1},\dots,n_\xi$. In the numerical experiments, we use the stochastic Galerkin method from [30] to calculate the terms $\vec{u}_\ell$ for the construction of the matrices N_ℓ. The divergence matrix B is defined as

$$B = [b_{cd}]\,, \qquad b_{cd} = -\int_D \varphi_c\,(\nabla\cdot\phi_d)\,,$$

and the velocity mass matrix G is defined as

$$G = [g_{ab}]\,, \qquad g_{ab} = \int_D \phi_b\,\phi_a.$$

7. Numerical experiments. We implemented the method in MATLAB using the IFISS 3.5 package [6], and we tested the algorithms using two benchmark problems: flow around an obstacle, and an expansion flow around a symmetric step. The stochastic Galerkin methods were used to solve both the Navier–Stokes problem (see [18, 30] for full description) and the eigenvalue problem (3.3), which was solved using the inexact Newton iteration from section 5. The sampling methods (Monte Carlo and stochastic collocation) entail generating a set of sample viscosities from (6.1), and for each sample solving a deterministic steady-state Navier–Stokes equation followed by a solution of a deterministic eigenvalue problem (4.2) with a matrix operator corresponding to sampled Jacobian matrix operator (6.3), where the deterministic eigenvalue problems at sample points were solved using function `eigs` in MATLAB. For the solution of the Navier–Stokes equation, in both sampling and stochastic Galerkin methods, we used a hybrid strategy in which an initial approximation was obtained from the solution of the stochastic Stokes problem, after which several steps of Picard iteration were used to improve the solution, followed by Newton iteration. A convergent iteration stopped when the Euclidean norm of the algebraic residual was smaller than 10^{-8}; see [30] for more details. Also, when replacing the mass matrix$M$ by the shifted mass matrixM_σ (see (2.4) and (2.5)), we set $\sigma = -10^{-2}$ as in [5]. The 300 eigenvalues with the largest real part of the deterministic eigenvalue problem with mean viscosity ν_1 for each of the two examples are displayed in Figure 7.1.

7.1. Flow around an obstacle. For the first example, we considered flow around an obstacle in a similar setup as studied in [30]. The domain of the channel and the discretization are shown in Figure 7.2. The spatial discretization used a stretched grid with 1008 $\boldsymbol{Q}_2-\boldsymbol{Q}_1$ finite elements. We note that these elements are referred to as *Taylor–Hood* in the literature. There were 8416 velocity and 1096 pressure degrees of freedom. The viscosity$\nu(x,\xi)$ was taken to be

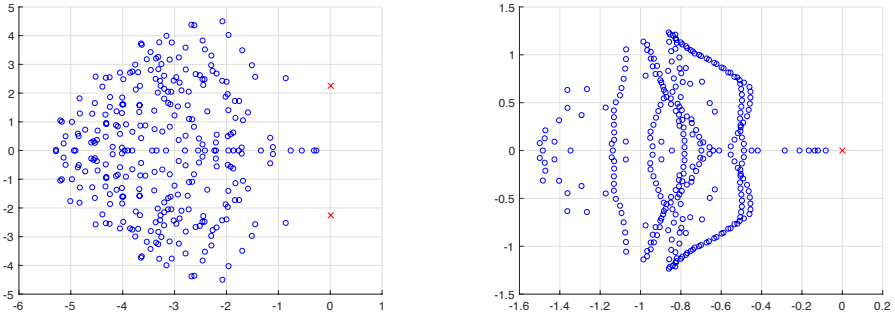

Figure 7.1. *The* 300 *eigenvalues with the largest real part of the matrices* K_1 *for the two examples: flow around an obstacle (left), and expansion flow around a symmetric step (right). The rightmost eigenvalues are indicated by a red cross.*

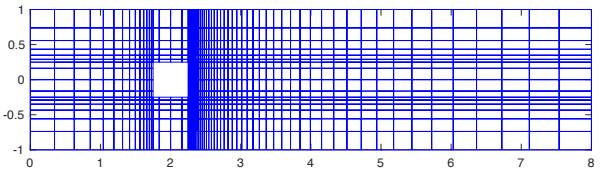

Figure 7.2. *Finite element mesh for the flow around an obstacle problem.*

a truncated lognormal process transformed from the underlying Gaussian process [11]. That is, $\psi_\ell(\xi)$, $\ell = 1, \ldots, n_\nu$, is a set of Hermite polynomials and, denoting the coefficients of the Karhunen–Loève expansion of the Gaussian process by$g_j(x)$ and $\eta_j = \xi_j - g_j$, $j = 1, \ldots, m_\xi$, the coefficients in expansion (6.1) were computed as

$$\nu_\ell(x) = \frac{\mathbb{E}\left[\psi_\ell(\eta)\right]}{\mathbb{E}\left[\psi_\ell^2(\eta)\right]} \exp\left[g_0 + \frac{1}{2}\sum_{j=1}^{m_\xi} \left(g_j(x)\right)^2\right].$$

The covariance function of the Gaussian field, for points $X_1 = (x_1, y_1)$ and $X_2 = (x_2, y_2)$ in$D$, was chosen to be

$$C\left(X_1, X_2\right) = \sigma_g^2 \exp\left(-\frac{|x_2 - x_1|}{L_x} - \frac{|y_2 - y_1|}{L_y}\right), \tag{7.1}$$

where L_x and L_y are the correlation lengths of the random variables ξ_i, $i = 1, \ldots, m_\xi$, in the x and y directions, respectively, and σ_g is the standard deviation of the Gaussian random field. The correlation lengths were set to be equal to 25% of the width and height of the domain. The coefficient of variation CoV of the lognormal field, defined as $CoV = \sigma_\nu/\nu_1$, where σ_ν is the standard deviation and ν_1 is the mean viscosity, was 1% or 10%. The viscosity (6.1) was parameterized using $m_\xi = 2$ random variables. According to [21], in order to guarantee a complete representation of the lognormal process by (6.1), the degree of polynomial expansion of $\nu(x, \xi)$ should be twice the degree of the expansion of the solution. We followed the same strategy here. Therefore, the values of n_ξ and n_ν are (cf., e.g. [12, p. 87] or [33, section 5.2]) $n_\xi = \frac{(m_\xi + p)!}{m_\xi! p!}$, $n_\nu = \frac{(m_\xi + 2p)!}{m_\xi!(2p)!}$. For the gPC expansion of eigenvalues/eigenvectors (3.5), the

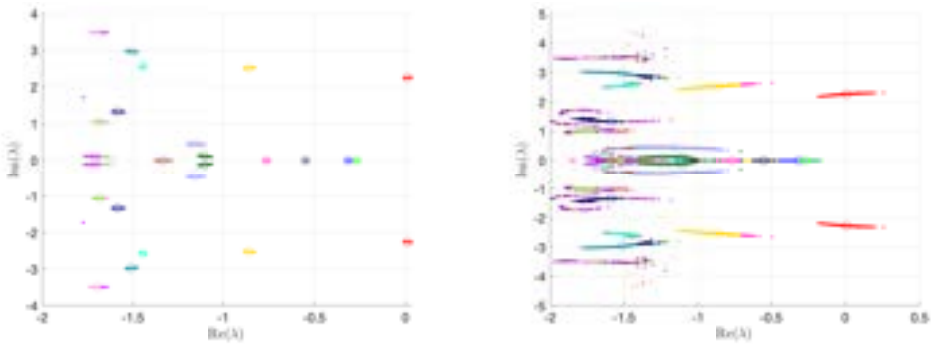

Figure 7.3. *Monte Carlo samples of* 25 *eigenvalues with the largest real part for the flow around an obstacle with* $CoV = 1\%$ *(left) and* $CoV = 10\%$ *(right). The eigenvalues of the mean problem are indicated by circles.*

maximal degree of gPC expansion is $p = 3$, so then $n_\xi = 10$ and $n_\nu = 28$. We used 1×10^3 samples for the Monte Carlo method and Smolyak sparse grid with Gauss–Hermite quadrature points and grid level4 for collocation; see, e.g., [17] for discussion of quadrature rules. With these settings, the size of $\{H_\ell\}_{\ell=1}^{n_\nu}$ in (5.3) was $10 \times 10 \times 28$ with 203 nonzeros, and there were $n_q = 29$ points in the sparse grid. As a consequence, the size of the stochastic Galerkin matrices is $n_\xi(n_u + n_p) = 95120$, the matrix associated with the Jacobian is fully block dense, and the mass matrix is block diagonal, but we note that these matrices are never formed in implementation. For the solution of the Navier–Stokes problem we used the hybrid strategy with 6 steps of Picard iteration followed by at most 15 steps of Newton iteration. We used mean viscosity $\nu_1 = 5.36193 \times 10^{-3}$, which corresponds to Reynolds number $Re = 373$, and the rightmost eigenvalue pair is $0.0085 \pm 2.2551i$; see the left panel in Figure 7.1. Figure 7.3 displays Monte Carlo realizations of the 25 eigenvalues with the largest real part for the values $CoV = 1\%$ and $CoV = 10\%$. It can be seen that the rightmost eigenvalue is relatively less sensitive to perturbation compared to the other eigenvalues, and since its real part is well separated from the rest of the spectrum, it can be easily identified in all runs of a sampling method. Figure 7.4 displays the probability density function (pdf) estimates of the rightmost eigenvalue with the positive imaginary part obtained directly by Monte Carlo, the stochastic collocation, and stochastic Galerkin methods, for which the estimates were obtained using MATLAB function `ksdensity` (in 2D) for sampled gPC expansions. Figure 7.5 shows plots of the estimated pdf of the real part of the rightmost eigenvalue. In both figures we can see a good agreement of the plots in the left column corresponding to $CoV = 1\%$ and in the right column corresponding to $CoV = 10\%$. In Table 7.1 we tabulate the coefficients of the gPC expansion of the rightmost eigenvalue with positive imaginary part computed using the stochastic collocation and Galerkin methods. A good agreement of coefficients can be seen, in particular, for coefficients with value much larger than zero, specifically with $k = 1, 2, 4, 6, 7$, and 9. Finally, in Table 7.2 we examine the inexact line-search Newton iteration from Algorithm 1. For the line-search method, we set $\rho = 0.9$ for the backtracking and $c = 0.25$. The initial guess is set using the rightmost eigenvalue and corresponding eigenvector of the eigenvalue problem (5.15) concatenated by zeros. The nonlinear iteration terminates when the norm of the residual $\|\widehat{r}_n\|_2 < 10^{-10}$. The linear systems in line 4 of Algorithm 1 are

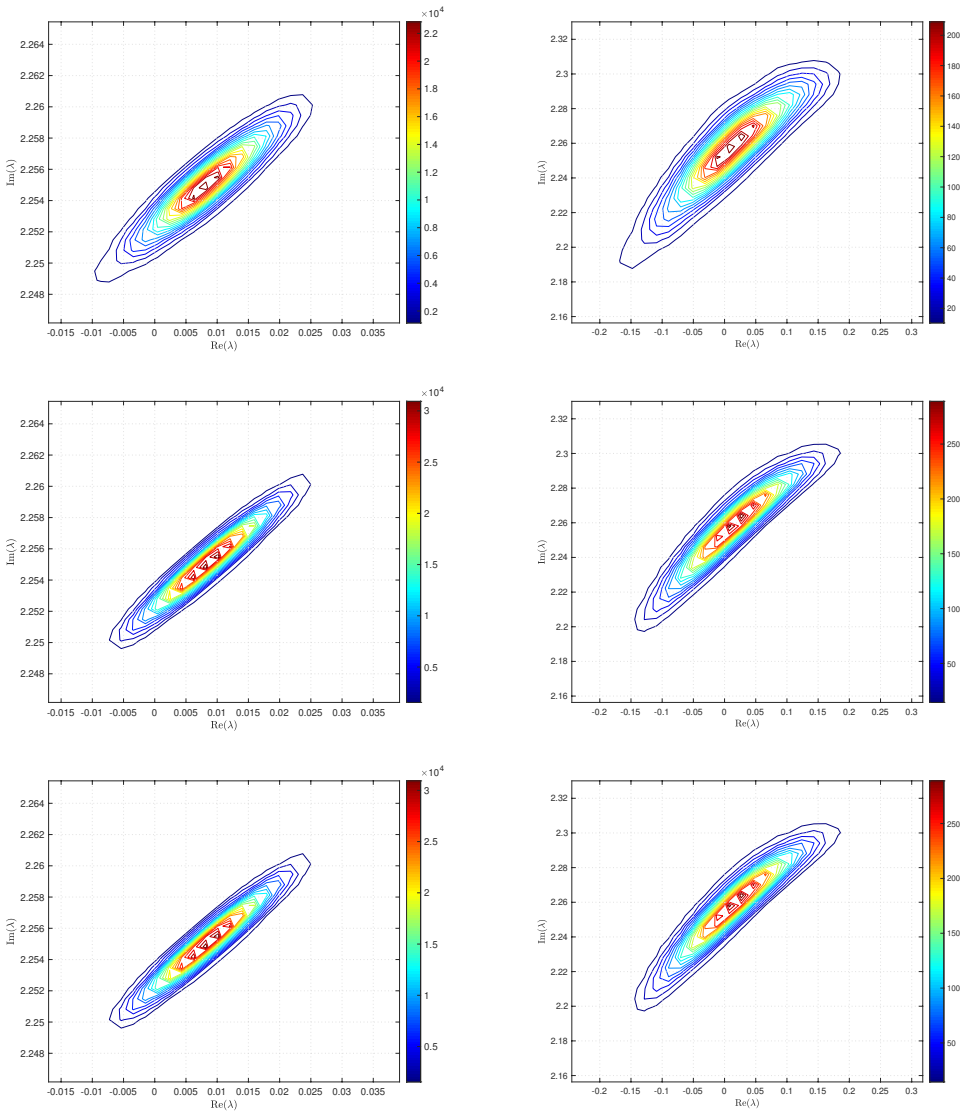

Figure 7.4. *Plots of the pdf estimate of the rightmost eigenvalue with positive imaginary part obtained using Monte Carlo (top), stochastic collocation (middle), and stochastic Galerkin method (bottom) for the flow around an obstacle with CoV = 1% (left) and CoV = 10% (right).*

solved using GMRES with the mean-based preconditioner (Algorithm 2), constraint mean-based preconditioner (Algorithm 3), and its updated variant discussed in section 5.2.1, and the constraint hierarchical Gauss–Seidel preconditioner (Algorithm 4–5), which was used without truncation of the matrix-vector multiplications and also with truncation, setting $p_t = 2$, as discussed in section 5.2. For the mean-based preconditioner we used $\epsilon_{\mathrm{Re}} = \epsilon_{\mathrm{Im}} = 0.97$, which worked best in our experience, and $\epsilon_{\mathrm{Re}} = \epsilon_{\mathrm{Im}} = 1$ otherwise. For the constraint mean-based preconditioner the matrix $\mathcal{M}_{\mathrm{cMB}}$ from (5.22) was factored using LU decomposition, and the updated variant from section 5.2.1 was used also in the constraint hierarchical Gauss–Seidel

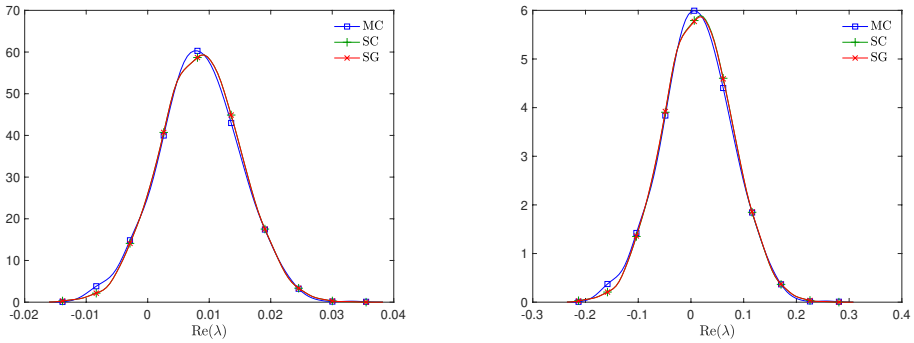

Figure 7.5. *Plots of the pdf estimate of the real part of the rightmost eigenvalue obtained using Monte Carlo (MC), stochastic collocation (SC), and stochastic Galerkin method (SG) for the flow around an obstacle with $CoV = 1\%$ (left) and $CoV = 10\%$ (right).*

preconditioner. First, we note that the performance of the algorithm with the mean-based preconditioner was very sensitive to the choice of ϵ_{Re} and ϵ_{Im}, and it can be seen that it is quite sensitive also to CoV. On the other hand, the performance of all variants of the constraint preconditioners appear to be far less sensitive, and we see only a mild increase in numbers of both nonlinear and GMRES iterations. Next, we see that updating the constraint mean-based preconditioner reduces the numbers of GMRES iterations, in particular in the latter steps of the nonlinear method. Finally, we see that using the constraint hierarchical Gauss–Seidel preconditioner further decreases the number of GMRES iterations; for smaller CoV it may be suitable to truncate the matrix-vector multiplications without any change in iteration counts, and even though we see with $CoV = 10\%$ an increase in number of nonlinear steps, the overall number of GMRES iterations remains smaller than when the two variants of the constraint mean-based preconditioner were used.

7.2. Expansion flow around a symmetric step. For the second example, we considered an expansion flow around a symmetric step. The domain and its discretization are shown in Figure 7.6. The spatial discretization used a uniform grid with 976 $\boldsymbol{Q}_2-\boldsymbol{P}_{-1}$ finite elements, which provided a stable discretization for the rectangular grid; see [8, p. 139]. There were 8338 velocity and 2928 pressure degrees of freedom. For the viscosity we considered a random field with affine dependence on the random variables ξ given as

$$\nu(x,\xi) = \nu_1 + \sigma_\nu \sum_{\ell=2}^{n_\nu} \nu_\ell(x)\,\xi_{\ell-1}, \tag{7.2}$$

where ν_1 is the mean and $\sigma_\nu = CoV \cdot \nu_1$ the standard deviation of the viscosity, $n_\nu = m_\xi + 1$, and $\nu_{\ell+1} = \sqrt{3\lambda_\ell} v_\ell(x)$ with $\{(\lambda_\ell, v_\ell(x))\}_{\ell=1}^{m_\xi}$ are the eigenpairs of the eigenvalue problem associated with the covariance kernel of the random field. As in the previous example, we used the values $CoV = 1\%$ and 10%. We considered the same form of the covariance kernel as in (7.1),

$$C(X_1, X_2) = \exp\left(-\frac{|x_2 - x_1|}{L_x} - \frac{|y_2 - y_1|}{L_y}\right),$$

Table 7.1

The 10 coefficients of the gPC expansion of the rightmost eigenvalue with positive complex part for the flow around an obstacle problem with $CoV = 1\%$ and 10% computed using stochastic collocation (SC), and stochastic Galerkin method (SG). Here d is the polynomial degree and k is the index of basis function in expansion (3.5).

d	k	SC	SG
		$CoV = 1\%$	
0	1	8.5726E-03 + 2.2551E + 00 i	8.5726E-03 + 2.2551E + 00 i
1	2	−6.5686E-03 − 2.2643E-03 i	−6.5686E-03 − 2.2643E-03 i
	3	1.1181E-16 − 2.0817E-14 i	2.6512E-17 + 8.3094E-17 i
2	4	−1.1802E-06 − 2.4274E-05 i	−1.2055e-06 − 2.4200E-05 i
	5	3.8351E-15 − 4.4964E-15 i	8.9732E-20 − 2.0565E-19 i
	6	−3.3393E-06 + 4.0603E-05 i	−3.3527E-06 + 4.0641E-05 i
3	7	−1.0635E-07 + 4.1735E-07 i	−8.5671E-08 + 3.5926E-07 i
	8	7.8095E-16 + 6.1617E-15 i	−4.3191E-22 − 8.3970E-21 i
	9	−4.6791E-07 + 5.1602E-08 i	−4.4762E-07 − 6.0766E-09 i
	10	2.2155E-15 + 4.6907E-15 i	1.2691E-15 + 2.9181E-16 i
		$CoV = 10\%$	
0	1	1.3420E-02 + 2.2577E + 00 i	1.3419E-02 + 2.2576E + 00 i
1	2	−6.6200E-02 − 2.2034E-02 i	−6.6243E-02 − 2.2018E-02 i
	3	1.6011E-15 − 1.0297E-14 i	1.1672E-15 + 8.8396E-16 i
2	4	−2.2415E-04 − 2.5416E-03 i	−1.0889E-04 − 2.4178E-03 i
	5	8.5869E-17 − 1.0547E-15 i	1.1865E-17 + 6.5559E-17 i
	6	−2.7323E-04 + 4.1219E-03 i	−2.1977E-04 + 4.1437E-03 i
3	7	−4.8106E-05 + 3.556E-04 i	1.3510E-04 + 9.1486E-05 i
	8	2.8365E-15 + 6.1062E-15 i	8.0683E-19 + 5.3753E-18 i
	9	−4.5696E-04 + 2.7795E-06 i	−4.1149E-04 − 1.8160E-04 i
	10	1.7408E-15 + 1.3101E-14 i	1.3975E-15 + 3.5152E-16 i

Table 7.2

The number of GMRES iterations in each step of inexact line-search Newton method (Algorithm 1) for computing the rightmost eigenvalue and corresponding eigenvectors of the flow around an obstacle problem with $CoV = 1\%$ (left) and 10% (right) and with the stopping criteria $\|r_n\|_2 < 10^{-10}$ and different choices of preconditioners: mean-based (MB) from Algorithm 2, constraint mean-based (cMB) from Algorithm 3 and its updated variant (cMB_u) from section 5.2.1, and the constraint hierarchical Gauss–Seidel preconditioner (chGS) from Algorithm 4–5 and also with truncation, setting $p_t = 2$ ($chGS_2$).

	$CoV = 1\%$					$CoV = 10\%$				
step	MB	cMB	cMB_u	chGS	$chGS_2$	MB	cMB	cMB_u	chGS	$chGS_2$
1	2	1	1	1	1	7	1	1	1	1
2	2	1	1	1	1	6	3	2	3	3
3	6	3	2	1	1	13	4	4	3	4
4	9	6	3	2	2	10	8	7	3	4
5	15	10	6	3	3	15	16	13	4	5
6						14			8	8
7						25				
8						32				
9						67				

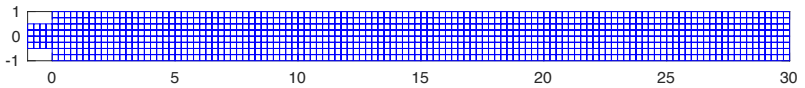

Figure 7.6. *Finite element mesh for the expansion flow around a symmetric step.*

and the correlation lengths were set to 12.5% of the width and 25% of the height of the domain. We assume that the random variables $\{\xi_\ell\}_{\ell=1}^{m_\xi}$ follow a uniform distribution over $(-1, 1)$. We note that (7.2) can be viewed as a special case of (6.1), which consists of only linear terms ofξ. For the parametrization of viscosity by (7.2) we used the same stochastic dimension m_ξ and degree of polynomial expansion$p$ as in the previous example: $m_\xi = 2$ and $p = 3$, so then $n_\xi = 10$ and $n_\nu = m_\xi + 1 = 3$. We used 1×10^3 samples for the Monte Carlo method and Smolyak sparse grid with Gauss–Legendre quadrature points and grid level4 for collocation. With these settings, the size of $\{H_\ell\}_{\ell=1}^{n_\nu}$ in (5.3) was $10 \times 10 \times 3$ with 34 nonzeros, and there were $n_q = 29$ points on the sparse grid. As a consequence, the size of the stochastic Galerkin matrices is 112660, and the matrix associated with the Jacobian has in this case a block-sparse structure; see, e.g., [17, p. 88]. For the solution of the Navier–Stokes problem we used the hybrid strategy with 20 steps of Picard iteration followed by at most 20 steps of Newton iteration. We used mean viscosity $\nu_1 = 4.5455 \times 10^{-3}$, which corresponds to Reynolds number $Re = 220$, and the rightmost eigenvalue is 5.7963×10^{-4} (the second largest eigenvalue is -8.2273×10^{-2}) see the right panel in Figure 7.1. Figure 7.7 displays Monte Carlo realizations of the 25 eigenvalues with the largest real part. As in the previous example, it can be seen that the rightmost eigenvalue is relatively less sensitive to perturbation compared to the other eigenvalues, and it can be easily identified in all runs of a sampling method. Figure 7.8 displays the probability density function (pdf) estimates of the rightmost eigenvalue obtained directly by Monte Carlo, the stochastic collocation, and stochastic Galerkin methods, for which the estimates were obtained using MATLAB function `ksdensity` for sampled gPC expansions. We can see a good agreement of the plots in the left column corresponding to $CoV = 1\%$ and in the right column corresponding to $CoV = 10\%$. In Table 7.3 we tabulate the coefficients of the gPC expansion of the rightmost eigenvalue computed using the stochastic collocation and stochastic Galerkin methods. A good agreement of coefficients can be seen, in particular, for coefficients with larger values. Finally, we examine the inexact line-search Newton iteration from Algorithm 1. For the line-search method, we used the same setup as before with $\rho = 0.9$ and $c = 0.25$. The initial guess is set using the rightmost eigenvalue and corresponding eigenvector of the eigenvalue problem (5.15), just with no imaginary part, concatenated by zeros. The nonlinear iteration terminates when the norm of the residual $\|\widehat{r}_n\|_2 < 10^{-10}$. The linear systems in line 4 of Algorithm 1 are solved using the right-preconditioned GMRES method as in the complex case. However, since the eigenvalue is real, the generalized eigenvalue problem as written in (5.5) has the (usual) form given by (3.3), and all algorithms formulated in this paper can be adapted by simply dropping the components corresponding to the imaginary part of the eigenvalue problem, for example, the constraints mean-based preconditioner (Algorithm 3), and specifically (5.22) reduces to

$$\mathcal{M}_{\text{cMB}} = \begin{bmatrix} K_1 - \epsilon_{\text{Re}}\mu_{\text{Re}}M_\sigma & -M_\sigma w_{\text{Re}}^{(0)} \\ -w_{\text{Re}}^{(0)T} & 0 \end{bmatrix}.$$

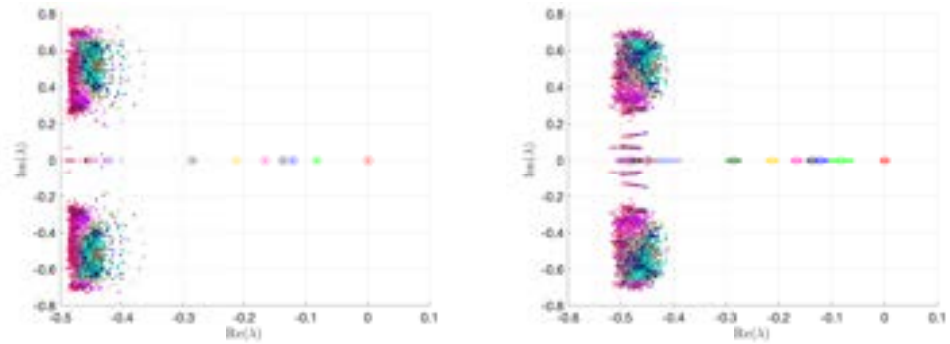

Figure 7.7. *Monte Carlo samples of* 25 *eigenvalues with the largest real part for the flow around an obstacle with* $CoV = 1\%$ *(left) and* $CoV = 10\%$ *(right). The eigenvalues of the mean problem are indicated by circles.*

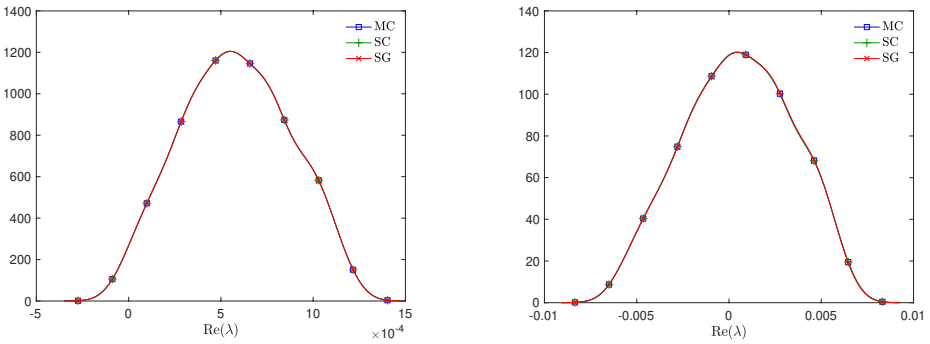

Figure 7.8. *Plots of the pdf estimate of the real part of the rightmost eigenvalue obtained using Monte Carlo (MC), stochastic collocation (SC), and stochastic Galerkin method (SG) for the expansion flow around a symmetric step with* $CoV = 1\%$ *(left) and* $CoV = 10\%$ *(right).*

and in the mean-based preconditioner (Algorithm 2) we also modified (5.19) as

$$\mathcal{M}_{\text{cMB}} = \begin{bmatrix} K_1 - \epsilon_{\text{Re}}\mu_{\text{Re}} I & 0 \\ 0 & w_{\text{Re}}^{(0)T} \left(K_1 - \epsilon_{\text{Re}}\mu_{\text{Re}} I\right)^{-1} w_{\text{Re}}^{(0)} \end{bmatrix},$$

that is, we used I instead of M_σ in the shift of the matrix K_1. We also adapted the constraint hierarchical Gauss–Seidel preconditioner (Algorithm 4–5), which was used as before without truncation of the matrix-vector multiplications and also with truncation, setting $p_t = 2$, as discussed in section 5.2. For the mean-based preconditioner we used $\epsilon_{\text{Re}} = 0.97$, but the preconditioner appeared to be far less sensitive to a specific value ofϵ_{Re}, and $\epsilon_{\text{Re}} = 1$ otherwise. We note that this way the algorithms are still formulated for a generalized nonsymmetric eigenvalue problem unlike in [19], where we studied symmetric problems, and in implementation we used a Cholesky factorization of the mass matrix in order to formulate a standard eigenvalue problem. From the results in Table 7.4 we see that for all preconditioners the overall number of nonlinear steps and GMRES iterations increases for larger CoV; although, all variants of the constraint preconditioner outperform the mean-based preconditioner and the total number of iterations remains relatively small. Next, the performance with constraint

Table 7.3

The 10 coefficients of the gPC expansion of the rightmost eigenvalue for the expansion flow around a symmetric step problem with $CoV = 1\%$ and 10% computed using stochastic collocation (SC), and stochastic Galerkin method (SG). Here d is the polynomial degree and k is the index of basis function in expansion (3.5).

d	k	SC	SG	SC	SG
		$CoV = 1\%$		$CoV = 10\%$	
0	1	5.7873E-04	5.7873E-04	4.8948E-04	4.8927E-04
1	2	−1.5948E-04	−1.5948E-04	−1.5890E-03	−1.5877E-03
	3	−2.3689E-04	−2.3689E-04	−2.3619E-03	−2.3605E-03
2	4	−2.4179E-07	−2.6041E-07	−2.4472E-05	−2.6501E-05
	5	−8.2562E-07	−8.7937E-07	−8.3136E-05	−8.8911E-05
	6	−5.6059E-07	−5.9203E-07	−5.6429E-05	−5.9831E-05
3	7	7.7918E-10	8.2134E-10	5.7810E-07	8.5057E-07
	8	2.5941E-09	3.9327E-09	2.8439E-06	4.022E-06
	9	3.8788E-09	5.5168E-09	4.0315E-06	5.6217E-06
	10	1.3002E-09	2.2685E-09	1.6668E-06	2.3171E-06

Table 7.4

The number of GMRES iterations in each step of inexact line-search Newton method (Algorithm 1) for computing the rightmost eigenvalue and corresponding eigenvectors of the expansion flow around a symmetric step problem with $CoV = 1\%$ (left) and 10% (right) and with the stopping criteria $\|r_n\|_2 < 10^{-10}$ and different choices of preconditioners: mean-based (MB) from Algorithm 2, constraint mean-based (cMB) from Algorithm 3 and its updated variant (cMB_u) from section 5.2.1, and the constraint hierarchical Gauss–Seidel preconditioner (chGS) from Algorithm 4–5 and also with truncation, setting $p_t = 2$ ($chGS_2$).

	$CoV = 1\%$					$CoV = 10\%$				
step	MB	cMB	cMB_u	chGS	$chGS_2$	MB	cMB	cMB_u	chGS	$chGS_2$
1	19	4	4	2	2	23	6	6	3	3
2	17	4	4	3	3	20	6	6	4	4
3	15	3	3	3	3	19	6	6	4	4
4						15	5	5	4	4
5						14	5	5	3	3
6						23	8	8	5	5

preconditioners seems not to improve with the updating discussed in section 5.2.1, which is likely since the numbers of iterations are already low. Finally, using the constraint hierarchical Gauss–Seidel preconditioner reduces the number of GMRES iterations, which is slightly more significant for larger values of CoV. The computational cost of the preconditioner may be reduced by using the truncation of the matrix-vector multiplications; specifically we see that the overall iteration counts with and without the truncation are the same.

8. Conclusion.

We studied inexact stochastic Galerkin methods for linear stability analysis of dynamical systems with parametric uncertainty and a specific application: the Navier–Stokes equation with stochastic viscosity. The model leads to a generalized eigenvalue problem with a symmetric mass matrix and nonsymmetric stiffness matrix, which was given by an affine expansion obtained from a stochastic expansion of the viscosity. For the assessment of linear stability we were interested in characterizing the rightmost eigenvalue using the generalized

polynomial chaos expansion. Since the eigenvalue of interest may be complex, we considered separated representation of the real and imaginary parts of the associated eigenpair. The algorithms for solving the eigenvalue problem were formulated on the basis of line-search Newton iteration, in which the associated linear systems were solved using the right-preconditioned GMRES method. We proposed several preconditioners: the mean-based preconditioner, the constraint mean-based preconditioner, and the constraint hierarchical Gauss–Seidel preconditioner. For the two constraint preconditioners we also proposed an updated version, which adapts the preconditioners to the linear system using the Sherman–Morrison–Woodbury formula after each step of Newton iteration. We studied two model problems: one when the rightmost eigenvalue is given by a complex conjugate pair, and another when the eigenvalue is real. The overall iteration count of GMRES with the constraint preconditioners was smaller compared to the mean-based preconditioner, and the constraint preconditioners were also less sensitive to the value of CoV. Also we found that updating the constraint preconditioner after each step of Newton iteration and using the off-diagonal blocks in the action of the constraint hierarchical Gauss–Seidel preconditioner may further decrease the overall iteration count, in particular when the rightmost eigenvalue is complex. Finally, for both problems the probability density function estimates of the rightmost eigenvalue closely matched the estimates obtained using the stochastic collocation and also with the direct Monte Carlo simulation.

Appendix A. Computations in inexact Newton iteration. The main component of a Krylov subspace method, such as GMRES, is the computation of a matrix-vector product. First, we note that the algorithms make use of the identity

$$(V \otimes W)\,\mathrm{vec}\,(U) = \mathrm{vec}\left(WUV^T\right). \tag{A.1}$$

Let us write a product with Jacobian matrix from (5.13) as

$$\widehat{DJ}(\bar{v}_{\mathrm{Re}}^{(n)}, \bar{v}_{\mathrm{Im}}^{(n)}, \bar{\lambda}_{\mathrm{Re}}^{(n)}, \bar{\lambda}_{\mathrm{Im}}^{(n)}) \begin{bmatrix} \delta\bar{v}_{\mathrm{Re}} \\ \delta\bar{v}_{\mathrm{Im}} \\ \delta\bar{\lambda}_{\mathrm{Re}} \\ \delta\bar{\lambda}_{\mathrm{Im}} \end{bmatrix},$$

where

$$\widehat{DJ}(\bar{v}_{\mathrm{Re}}^{(n)}, \bar{v}_{\mathrm{Im}}^{(n)}, \bar{\lambda}_{\mathrm{Re}}^{(n)}, \bar{\lambda}_{\mathrm{Im}}^{(n)}) = \begin{bmatrix} A_{\mathrm{Re}} & A_{\mathrm{Im}} & B_{\mathrm{Re}} & B_{\mathrm{Im}} \\ -\frac{1}{2}C_{\mathrm{Re}} & -\frac{1}{2}C_{\mathrm{Im}} & 0 & 0 \end{bmatrix}, \tag{A.2}$$

with A_{Re}, A_{Im}, B_{Re}, B_{Im}, and C_{Re}, C_{Im} denoting the matrices in (5.9)–(5.12). Then, using (5.17) and (A.1), the matrix-vector product entails evaluating

$$\begin{aligned} A_{\mathrm{Re}}\delta\bar{v}_{\mathrm{Re}} &= \begin{bmatrix} \mathbb{E}\left[\Psi\Psi^T \otimes K\right] - \mathbb{E}[(\bar{\lambda}_{\mathrm{Re}}^{(n)T}\Psi)\Psi\Psi^T \otimes M_\sigma] \\ -\mathbb{E}[(\bar{\lambda}_{\mathrm{Im}}^{(n)T}\Psi)\Psi\Psi^T \otimes M_\sigma] \end{bmatrix} \delta\bar{v}_{\mathrm{Re}} \\ &= \begin{bmatrix} \mathrm{vec}\left(\sum_{\ell=1}^{n_\nu} K_\ell \delta\bar{V}_{\mathrm{Re}} H_\ell^T\right) - \mathrm{vec}\left(\sum_{\ell=1}^{n_\xi} \lambda_{\mathrm{Re},\ell}^{(n)} M_\sigma \delta\bar{V}_{\mathrm{Re}} H_\ell^T\right) \\ -\mathrm{vec}\left(\sum_{\ell=1}^{n_\xi} \lambda_{\mathrm{Im},\ell}^{(n)} M_\sigma \delta\bar{V}_{\mathrm{Re}} H_\ell^T\right) \end{bmatrix}, \end{aligned} \tag{A.3}$$

$$
\begin{aligned}
(\text{A.4}) \quad A_{\text{Im}}\delta\bar{v}_{\text{Im}} &= \begin{bmatrix} \mathbb{E}[(\bar{\lambda}_{\text{Im}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma] \\ \mathbb{E}[\Psi\Psi^T\otimes K]-\mathbb{E}[(\bar{\lambda}_{\text{Re}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma] \end{bmatrix}\delta\bar{v}_{\text{Im}} \\
&= \begin{bmatrix} \text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Im},\ell}^{(n)}M_\sigma\delta\bar{V}_{\text{Re}}H_\ell^T\right) \\ \text{vec}\left(\sum_{\ell=1}^{n_\nu}K_\ell\delta\bar{V}_{\text{Re}}H_\ell^T\right)-\text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Re},\ell}^{(n)}M_\sigma\delta\bar{V}_{\text{Re}}H_\ell^T\right) \end{bmatrix}, \\
(\text{A.5}) \quad B_{\text{Re}}\delta\bar{\lambda}_{\text{Re}} &= \begin{bmatrix} -\mathbb{E}[\Psi^T\otimes(\Psi\Psi^T\otimes M_\sigma)]\bar{v}_{\text{Re}}^{(n)} \\ -\mathbb{E}[\Psi^T\otimes(\Psi\Psi^T\otimes M_\sigma)]\bar{v}_{\text{Im}}^{(n)} \end{bmatrix}\delta\bar{\lambda}_{\text{Re}} = \begin{bmatrix} -\text{vec}\left(\sum_{\ell=1}^{n_\xi}\delta\lambda_{\text{Re},\ell}M_\sigma\bar{V}_{\text{Re}}^{(n)}H_\ell^T\right) \\ -\text{vec}\left(\sum_{\ell=1}^{n_\xi}\delta\lambda_{\text{Re},\ell}M_\sigma\bar{V}_{\text{Im}}^{(n)}H_\ell^T\right) \end{bmatrix}, \\
(\text{A.6}) \quad B_{\text{Im}}\bar{\lambda}_{\text{Im}} &= \begin{bmatrix} \mathbb{E}[\Psi^T\otimes(\Psi\Psi^T\otimes M_\sigma)]\bar{v}_{\text{Im}}^{(n)} \\ -\mathbb{E}[\Psi^T\otimes(\Psi\Psi^T\otimes M_\sigma)]\bar{v}_{\text{Re}}^{(n)} \end{bmatrix}\delta\bar{\lambda}_{\text{Im}} = \begin{bmatrix} \text{vec}\left(\sum_{\ell=1}^{n_\xi}\delta\lambda_{\text{Im},\ell}M_\sigma\bar{V}_{\text{Im}}^{(n)}H_\ell^T\right) \\ -\text{vec}\left(\sum_{\ell=1}^{n_\xi}\delta\lambda_{\text{Im},\ell}M_\sigma\bar{V}_{\text{Re}}^{(n)}H_\ell^T\right) \end{bmatrix}, \\
(\text{A.7}) \quad -\frac{1}{2}C_{\text{Re}}\delta\bar{v}_{\text{Re}} &= -\begin{bmatrix} \mathbb{E}[\Psi\otimes(\bar{v}_{\text{Re}}^{(n)T}\Psi\Psi^T\otimes I_{n_x})] \\ 0 \end{bmatrix}\delta\bar{v}_{\text{Re}} = -\begin{bmatrix} \begin{bmatrix} \bar{v}_{\text{Re}}^{(n)T}(H_1\otimes I_{n_x})\delta\bar{v}_{\text{Re}} \\ \vdots \\ \bar{v}_{\text{Re}}^{(n)T}(H_{n_\xi}\otimes I_{n_x})\delta\bar{v}_{\text{Re}} \end{bmatrix} \\ 0 \end{bmatrix}, \\
(\text{A.8}) \quad -\frac{1}{2}C_{\text{Im}}\delta\bar{v}_{\text{Im}} &= -\begin{bmatrix} 0 \\ \mathbb{E}[\Psi\otimes(\bar{v}_{\text{Im}}^{(n)T}\Psi\Psi^T\otimes I_{n_x})] \end{bmatrix}\delta\bar{v}_{\text{Im}} = -\begin{bmatrix} 0 \\ \begin{bmatrix} \bar{v}_{\text{Im}}^{(n)T}(H_1\otimes I_{n_x})\delta\bar{v}_{\text{Im}} \\ \vdots \\ \bar{v}_{\text{Im}}^{(n)T}(H_{n_\xi}\otimes I_{n_x})\delta\bar{v}_{\text{Im}} \end{bmatrix} \end{bmatrix},
\end{aligned}
$$

and the right-hand side of (5.13) is evaluated using

$$
\begin{aligned}
F^{(n)} &= \begin{bmatrix} \mathbb{E}[\Psi\Psi^T\otimes K]\bar{v}_{\text{Re}}^{(n)}-\mathbb{E}[(\bar{\lambda}_{\text{Re}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma]\bar{v}_{\text{Re}}^{(n)}+\mathbb{E}[(\bar{\lambda}_{\text{Im}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma]\bar{v}_{\text{Im}}^{(n)} \\ \mathbb{E}[\Psi\Psi^T\otimes K]\bar{v}_{\text{Im}}^{(n)}-\mathbb{E}[(\bar{\lambda}_{\text{Im}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma]\bar{v}_{\text{Re}}^{(n)}-\mathbb{E}[(\bar{\lambda}_{\text{Re}}^{(n)T}\Psi)\Psi\Psi^T\otimes M_\sigma]\bar{v}_{\text{Im}}^{(n)} \end{bmatrix} \\
&= \begin{bmatrix} \text{vec}\left(\sum_{\ell=1}^{n_\nu}K_\ell\delta\bar{V}_{\text{Re}}^{(n)}H_\ell^T\right)-\text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Re},\ell}^{(n)}M_\sigma\bar{V}_{\text{Re}}^{(n)}H_\ell^T\right)+\text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Im},\ell}^{(n)}M_\sigma\bar{V}_{\text{Im}}^{(n)}H_\ell^T\right) \\ \text{vec}\left(\sum_{\ell=1}^{n_\nu}K_\ell\delta\bar{V}_{\text{Im}}^{(n)}H_\ell^T\right)-\text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Im},\ell}^{(n)}M_\sigma\bar{V}_{\text{Re}}^{(n)}H_\ell^T\right)-\text{vec}\left(\sum_{\ell=1}^{n_\xi}\lambda_{\text{Re},\ell}^{(n)}M_\sigma\bar{V}_{\text{Im}}^{(n)}H_\ell^T\right) \end{bmatrix}
\end{aligned}
$$

and

$$G^{(n)} = \begin{bmatrix} \mathbb{E}[\Psi \otimes ((\bar{v}_{\mathrm{Re}}^{(n)T}(\Psi\Psi^T \otimes I_{n_x})\bar{v}_{\mathrm{Re}}^{(n)}) - 1)] \\ \\ \mathbb{E}[\Psi \otimes ((\bar{v}_{\mathrm{Im}}^{(n)T}(\Psi\Psi^T \otimes I_{n_x})\bar{v}_{\mathrm{Im}}^{(n)}) - 1)] \end{bmatrix},$$

where, using $\star$ for either Re or Im, the ith row of$G^{(n)}$ is

$$\begin{aligned} \left[G^{(n)}\right]_i &= \mathbb{E}[\psi_i(\bar{v}_\star^{(n)T}(\Psi\Psi^T \otimes I_{n_x})\bar{v}_\star^{(n)}) - \psi_i], \\ &= \bar{v}_\star^{(n)T}\mathbb{E}[\psi_i\Psi\Psi^T \otimes I_{n_x}]\bar{v}_\star^{(n)} - \delta_{1i}, \end{aligned}$$

and the first term above is evaluated as

$$\bar{v}_\star^{(n)T}\mathbb{E}[\psi_i\Psi\Psi^T \otimes I_{n_x}]\bar{v}_\star^{(n)} = \bar{v}_\star^{(n)T}(H_i \otimes I_{n_x})\bar{v}_\star^{(n)},$$

or, denoting the trace operator bytr, this term can be also evaluated as

$$\bar{v}_\star^{(n)T}\mathbb{E}[\psi_i\Psi\Psi^T \otimes I_{n_x}]\bar{v}_\star^{(n)} = \mathrm{tr}(\bar{V}_\star^{(n)}H_i\bar{V}_\star^{(n)T}) = \mathrm{tr}(\bar{V}_\star^{(n)T}\bar{V}_\star^{(n)}H_i).$$

Appendix B. Parameters used in numerical experiments. In addition to the description in section 7, we provide in Table B.1 an overview of the main parameters used in the numerical experiments. Besides that, we used the following settings in both experiments: the gPC parameters $m_\xi = 2$, $p = 3$, $n_\xi = 10$; for the sampling methods, $n_{MC} = 1 \times 10^3$, $n_q = 29$; for the inexact Newton iteration, $\rho = 0.9$, $c = 0.25$, stopping criterion $\|\widehat{r}_n\|_2 < 10^{-10}$; for the preconditioners, $\epsilon_{\mathrm{Re}} = \epsilon_{\mathrm{Im}} = 0.97$ (the mean-based preconditioner) and $\epsilon_{\mathrm{Re}} = \epsilon_{\mathrm{Im}} = 1$ (otherwise).

Table B.1
The main parameters used in the numerical experiments.

	Section 7.1	Section 7.2
problem	Flow around an obstacle	Expansion flow around a symmetric step
FEM	$\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$	$\boldsymbol{Q}_2$– $\boldsymbol{P}_{-1}$
nelem/n_u/n_p	1008/8416/1096	976/8338/2928
Re	373	220
λ	$0.0085 \pm 2.2551i$	5.7963×10^{-4}
n_ν	28	3
quadrature (in SC)	Gauss–Hermite	Gauss–Legendre
Solving the Navier–Stokes problem (see [30] for details):		
max Picard steps	6	20
max Newton steps	15	20
nltol	10^{-8}	10^{-8}

Acknowledgment. We would like to thank the anonymous reviewers for their insightful suggestions. Bedřich Sousedík would also like to thank Professors Pavel Burda, Vladimír Janovský, Howard Elman, and Jaroslav Novotný for many discussions about bifurcation analysis and Navier–Stokes equations that inspired this work.

REFERENCES

[1] R. Andreev and C. Schwab, *Sparse tensor approximation of parametric eigenvalue problems*, in Numerical Analysis of Multiscale Problems, Lect. Notes Comput. Sci. Engrg. 83, Springer, Berlin, 2012, pp. 203–241, https://doi.org/10.1007/978-3-642-22061-6˙7.

[2] A. Bespalov, D. Loghin and R. Youngnoi, *Truncation preconditioners for stochastic Galerkin finite element discretizations*, SIAM J. Sci. Comput., 43 (2021), pp. S92–S116, https://doi.org/10.1137/20M1345645.

[3] K. A. Cliffe, T. J. Garratt, and A. Spence, *Eigenvalues of block matrices arising from problems in fluid mechanics*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1310–1318, https://doi.org/10.1137/S0895479892233230.

[4] K. A. Cliffe, A. Spence, and S. J. Taverner, *The numerical analysis of bifurcation problems with application to fluid mechanics*, Acta Numer., 9 (2000), pp. 39–131.

[5] H. C. Elman, K. Meerbergen, A. Spence, and M. Wu, *Lyapunov inverse iteration for identifying Hopf bifurcations in models of incompressible flow*, SIAM J. Sci. Comput., 34 (2012), pp. A1584–A1606, https://doi.org/10.1137/110827600.

[6] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: A computational laboratory for investigating incompressible flow problems*, SIAM Rev., 56 (2014), pp. 261–273, https://doi.org/10.1137/120891393.

[7] H. C. Elman and D. J. Silvester, *Collocation methods for exploring perturbations in linear stability analysis*, SIAM J. Sci. Comput., 40 (2018), pp. A2667–A2693, https://doi.org/10.1137/17M1117689.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, 2nd ed., Numer. Math. Sci. Comput., Oxford University Press, New York, 2014.

[9] H. C. Elman and T. Su, *Low-rank solution methods for stochastic eigenvalue problems*, SIAM J. Sci. Comput., 41 (2019), pp. A2657–A2680, https://doi.org/10.1137/18M122100X.

[10] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 317–339, https://doi.org/10.1051/m2an/2011045.

[11] R. G. Ghanem, *The nonlinear Gaussian spectrum of log-normal stochastic processes and variables*, J. Appl. Mech., 66 (1999), pp. 964–973, https://doi.org/10.1115/1.2791806.

[12] R. G. Ghanem and P. D. Spanos (1991). *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York; revised edition by Dover, 2003.

[13] W. J. F. Govaerts, *Numerical Methods for Bifurcations of Dynamical Equilibria*, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719543.

[14] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239, https://doi.org/10.1137/1031049.

[15] H. Hakula and M. Laaksonen, *Multiparametric shell eigenvalue problems*, Comput. Methods Appl. Mech. Eng., 343 (2019), pp. 721–745, https://doi.org/10.1016/j.cma.2018.09.011.

[16] W. Kerner, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85, https://doi.org/10.1016/0021-9991(89)90200-3.

[17] O. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Sci. Comput., Springer, 2010.

[18] K. Lee, H. C. Elman, and B. Sousedík, *A low-rank solver for the Navier–Stokes equations with uncertain viscosity*, SIAM/ASA J. Uncertain. Quantif., 7 (2019), pp. 1275–1300, https://doi.org/10.1137/17M1151912.

[19] K. Lee and B. Sousedík, *Inexact methods for symmetric stochastic eigenvalue problems*, SIAM/ASA J. Uncertain. Quantif., 6 (2018), pp. 1744–1776, https://doi.org/10.1137/18M1176026.

[20] The Mathworks, Inc., *MATLAB version 9.10.0.1684407 (R2021a) Update 3*, Natick, MA, 2021.

[21] H. G. Matthies and A. Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Methods Appl. Mech. Eng., 194 (2005), pp. 1295–1331, https://doi.org/10.1016/j.cma.2004.05.027.
[22] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, PA, 2000.
[23] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[24] C. E. Powell and D. J. Silvester, *Preconditioning steady-state Navier–Stokes equations with random data*, SIAM J. Sci. Comput., 34 (2012), pp. A2482–A2506, https://doi.org/10.1137/120870578.
[25] E. Sarrouy, O. Dessombz, and J.-J. Sinou, *Stochastic analysis of the eigenvalue problem for mechanical systems using polynomial chaos expansion—application to a finite element rotor*, J. Vib. Acoust., 134 (2012), pp. 1–12.
[26] E. Sarrouy, O. Dessombz, and J.-J. Sinou, *Piecewise polynomial chaos expansion with an application to brake squeal of a linear brake system*, J. Sound Vib., 332 (2013), pp. 577–594, https://doi.org/10.1016/j.jsv.2012.09.009.
[27] P. J. Schmid and D. S. Henningson, *Stability and Transition in Shear Flows*, Springer, New York, 2001.
[28] B. E. Sonday, R. D. Berry, H. N. Najm, and B. J. Debusschere, *Eigenvalues of the Jacobian of a Galerkin-projected uncertain ODE system*, SIAM J. Sci. Comput., 33 (2011), pp. 1212–1233, https://doi.org/10.1137/100785922.
[29] B. Sousedík and H. C. Elman, *Inverse subspace iteration for spectral stochastic finite element methods*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 163–189, https://doi.org/10.1137/140999359.
[30] B. Sousedík and H. C. Elman, *Stochastic Galerkin methods for the steady-state Navier-Stokes equations*, J. Comput. Phys., 316 (2016), pp. 435–452, https://doi.org/10.1016/j.jcp.2016.04.013.
[31] B. Sousedík, H. C. Elman, K. Lee, and R. Price, *On surrogate learning for linear stability assessment of Navier-Stokes equations with stochastic viscosity*, Appl. Math., first online, https://doi.org/10.21136/AM.2022.0046-21; preprint available at https://arxiv.org/abs/2103.00622.
[32] B. Sousedík and R. G. Ghanem, *Truncated hierarchical preconditioning for the stochastic Galerkin FEM*, Int. J. Uncertain. Quantif., 4 (2014), pp. 333–348, https://doi.org/10.1615/Int.J.UncertaintyQuantification.2014007353.
[33] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.
[34] D. Xiu and G. E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644, https://doi.org/10.1137/S1064827501387826.