Parametric and stochastic equations, or: to be or not to be intrusive? Part I: the plain vanilla Galerkin case*

Loïc Giraldi[c], Alexander Litvinenko[a], Dishi Liu[b], Hermann G. Matthies[†] [a], and Anthony Nouy[c]

[a]Institute of Scientific Computing, Technische Universität Braunschweig
[b]$C^2A^2S^2E$, German Aerospace Center (DLR), Braunschweig
[c]Institut de Recherche en Génie Civil et Mécanique,
École Centrale de Nantes

4th February 2013

Abstract

Parametric problems and Galerkin approximation. ...

1 Introduction

Many problems depend on parameters. See [4] for a synopsis on our approach to such parametric problems. [1] and [2].

While not such an existential question as for *Hamlet*, the question is still of considerable importance on the choice of methods.

Some methods obviously non-intrusive (MC, collocation). Some look like they are intrusive (Galerkin and variants like PGD, SR1U).

A method for a parametric problem is considered intrusive, if one has to modify the original software to solve the parametric problem. Question hinges on what kind of interface one has to the software.

Intrusive or not is not associated to a method, but rather is a software-engineering question. The difference is coupled or un-coupled.

*This work was partly supported by the Deutsche Forschungsgemeinschaft (DFG)

[†]Corresponding author:TU Braunschweig, D-38092 Braunschweig, Germany, e-mail:wire@tu-bs.de

2 Parametric Problems

The problem of parameters and Galerkin

To be more specific, let us consider the following situation: we are investigating some physical system which is modelled by an equation for its state:

$$A(p;u) = f(p), \tag{1}$$

where $u \in \mathcal{U}$ describes the state of the system lying in a Hilbert space $\mathcal{U}$ (for the sake of simplicity), A is an operator modelling the physics of the system, and $f \in \mathcal{U}^*$ is some external influence (action / excitation / loading). The model depends on some parameters $p \in \mathcal{P}$. In many cases Eq. (1) is the abstract formulation of a partial differential equation. But for the sake of simplicity we shall assume here that we are dealing with a model on a finite-dimensional space $\mathcal{U}$, e.g. a partial differential equation after discretisation.

Assume that for all $p \in \mathcal{P}$, Eq. (1) is a well-posed problem. This means that A as a mapping $u \mapsto A(p,u)$ is continuously invertible, which implies that this map is injective. If it is also differentiable w.r.t. u, this means that the derivative $\mathrm{D}_u A$ is non-singular and also continuously invertible. Now one may invoke the implicit function theorem, which assures one that it is possible to define the state u as a function of p — at least locally. Let this 'solution' be denoted by $u^*(p)$, such that for all $p: \; A(p;u^*(p)) = f(p)$.

Furthermore assume that we are also given an iterative solver — convergent for all values of p — which generates successive iterates for $k = 0, \dots,$

$$u^{(k+1)}(p) = S(k,p,u^{(k)}(p), R^{(k)}(p)), \quad \text{with } u^{(k)}(p) \to u^*(p), \tag{2}$$

where S is one cycle of the solver, k the iteration counter, $u^{(0)}$ some starting vector, and $R^{(k)}$ the residuum of Eq. (1)

$$R^{(k)} := R(p,u^{(k)}) := f(p) - A(p;u^{(k)}). \tag{3}$$

In the iteration in Eq. (2) we may set $u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$ with

$$\Delta u^{(k)} := S(u^{(k)}, R(u^{(k)})) - u^{(k)}, \quad \text{and usually} \tag{4}$$

$$P_k(\Delta u^{(k)}) = R^{(k)}, \tag{5}$$

so that in Eq. (2): $S(u^{(k)}) = u^{(k)} + P_k^{-1}(R(u^{(k)}))$. Here P_k is some pre-conditioner, which may depend on p, the iteration counter k, and on the current iterate $u^{(k)}$; e.g. in *Newton's method* $P_k = \mathrm{D}_u A(p;u^{(k)})$. In any case, we assume that for all k and p, the map P_k is linear and non-singular. The iteration corresponding to a normal solve for a particular value of p then is given in algorithm 2.1.

As this is assumed to be a convergent iteration, one has that $\|\Delta u^{(k+1)}(p)\| \leq \varrho(p)\,\|\Delta u^{(k)}(p)\|$ (at least for k large enough), with $\varrho(p) < 1$. For the convergence analysis to follow later we will assume that the convergence factors or Lipschitz constants $\varrho(p)$ are uniformly bounded for all values of $p \in \mathcal{P}$ by a constant strictly less than unity, i.e.

Algorithm 2.1 Iteration of Eq. (2)

```
                                        ▷ %comment: Start with some initial guess $u^{(0)}$%
  $k \leftarrow 0$
  while no convergence do
                                        ▷ %comment: the global iteration loop%
      Compute $\Delta u^{(k)}$ according to Eq. (4) or Eq. (5)
      $u^{(k+1)} \leftarrow u^{(k)} + \Delta u^{(k)}$
      $k \leftarrow k + 1$
  end while
```

$\varrho(p) \leq \varrho^* < 1$. Another way of saying this is that for all $u, v \in \mathcal{U}$, $p \in \mathcal{P}$, $k \geq 0$ the iterator S in Eq. (2) is uniformly Lipschitz continuous with Lipschitz constant $\varrho^* < 1$, i.e. a strict contraction:

$$\|S(k,p,u,R(u)) - S(k,p,v,R(v))\| \leq \varrho^* \, \|u - v\|. \tag{6}$$

3 Galerkin approximation of parametric dependence

The dependence of u on the parameters p one would like to approximate $u^*(p)$ in the following fashion:

$$u^*(p) \approx u(p) = \sum_{\alpha \in \mathcal{A}} u_\alpha \psi_\alpha(p), \tag{7}$$

where $u_\alpha \in \mathcal{U}$ are vector coefficients to be determined, and ψ_α are some linearly independent functions, whose linear combinations form the Galerkin subspace $\mathcal{Q}_{\mathcal{A}} := \text{span}\{\psi_\alpha\}_{\alpha \in \mathcal{A}} \subset \mathbb{R}^{\mathcal{P}}$, and $\mathcal{A}$ is some finite set of indices of cardinality $|\mathcal{A}|$. Often the set $\mathcal{A}$ has no canonical order, but for the purpose of computation later we will assume that some particular ordering has been chosen.

If we take the *ansatz* Eq. (7) and insert it into Eq. (1), the residuum Eq. (3) will usually not vanish for all p, as the finite set of functions $\{\psi_\alpha\}_{\alpha \in \mathcal{A}}$ can not match all possible parametric variations of $u(p)$.

3.1 The Galerkin equations for the residual

The *Galerkin* method — also called the method of weighted residuals — determines the unknown coefficients u_α in Eq. (7) by requiring that for all $\beta \in \mathcal{A}$

$$\boldsymbol{G}_p(\xi_\beta(\cdot) R_{\mathcal{A}}(\cdot)) = 0, \tag{8}$$

where the residuum from the ansatz Eq. (7) is $R_{\mathcal{A}}(p) = f(p) - A(p; \sum_{\alpha \in \mathcal{A}} u_\alpha \psi_\alpha(p))$, and $\{\xi_\beta\}_{\beta \in \mathcal{A}}$ is a set of linearly independent functions of p. The linear Galerkin projector $\boldsymbol{G}_p$ is defined by requiring that for any scalar functions $\phi(p), \varphi(p)$ and vector $v \in \mathcal{U}$ one has

$$\boldsymbol{G}_p(\phi(\cdot)\,(\varphi(\cdot) \otimes v)) = \boldsymbol{G}_p(\phi(\cdot)\, v \, \varphi(\cdot)) = \langle \phi, \varphi \rangle_p \, v, \tag{9}$$

where $\langle\cdot,\cdot\rangle_p$ is some duality pairing on a subspace of the scalar functions $\mathbb{R}^{\mathcal{P}}$, and from this $\boldsymbol{G}_p$ can be extended by linearity. In case $\mathcal{P}$ is a measure space with measure μ, then that pairing often is $\langle\phi,\varphi\rangle_p = \int_{\mathcal{P}} \phi(p)\varphi(p)\,\mu(\mathrm{d}p)$, and if $\mu(\mathcal{P}) = 1$, such that $\mathcal{P}$ may be considered as a probability space with expectation operator $\mathbb{E}\left(\phi\right) = \int_{\mathcal{P}} \phi(p)\,\mu(\mathrm{d}p)$, then $\langle\phi,\varphi\rangle_p = \mathbb{E}\left(\phi\varphi\right)$. A bit more general is to allow $\langle\phi,\varphi\rangle_p = \int_{\mathcal{P}\times\mathcal{P}} \varkappa(p,q)\phi(p)\varphi(q)\,\mu(\mathrm{d}p)\mu(\mathrm{d}q)$, where $\varkappa$ is a symmetric positive definite kernel. What is important for what is to follow, and what we want to assume from now on, is that the pairing is given by some integral.

The set $\{\psi_\alpha\}_{\alpha\in\mathcal{A}}$ determines the Galerkin subspace $\mathcal{Q}_{\mathcal{A}} := \mathrm{span}\{\psi_\alpha\}_{\alpha\in\mathcal{A}} \subset \mathbb{R}^{\mathcal{P}}$, whereas the set $\{\xi_\beta\}_{\beta\in\mathcal{A}}$ determines the projection onto that subspace. The subspace $\mathcal{Q}_{\mathcal{A}}$ determines the approximation properties, whereas the projection is important for the stability of the procedure, as the projection is orthogonal to $\mathcal{R}_{\mathcal{A}} := \mathrm{span}\{\xi_\beta\}_{\beta\in\mathcal{A}}$.

Often one takes $\xi_\beta = \psi_\beta$, this is then sometimes called the *Bubnov-Galerkin* method, whereas in the general case $\xi_\beta \neq \psi_\beta$ one speaks of the *Petrov-Galerkin* method.

Explicitly writing down Eq. (8), one obtains for all β

$$\boldsymbol{G}_p(\xi_\beta(\cdot)(f(p) - A(p; \sum_{\alpha\in\mathcal{A}} u_\alpha\psi_\alpha(p)))) = 0. \tag{10}$$

It is important to recognise that Eq. (10) is a — usually coupled — system of equations for the unknown u_α. These equations look sufficiently different from Eq. (1), so that the common wisdom is that the solution of Eq. (10) requires new software and new methods, and that the solver Eq. (2) is of no use here. As a change or re-write of the existing software seems to be necessary, the resulting methods are often labeled intrusive.

As a remark, observe that if one chooses $\xi_\beta(p) = \delta_\beta(p) = \delta(p-p_\beta)$ — the *delta-function* associated to the duality pairing $\langle\cdot,\cdot\rangle_p$ (i.e. $\langle\phi,\delta_\beta\rangle_p = \phi(p_\beta)$) — where the p_β are distinct points in $\mathcal{P}$ in Eq. (13), this becomes for all β:

$$f(p_\beta) - A(p_\beta; \sum_{\alpha\in\mathcal{A}} u_\alpha\psi_\alpha(p_\beta)) = f(p_\beta) - A(p_\beta; u_\beta) = 0, \tag{11}$$

where the last of these equalities holds only in case the basis $\{\psi_\alpha\}$ satisfies the *Kronecker-δ* property $\psi_\alpha(p_\beta) = \delta_{\alpha,\beta}$ — the Kronecker-δ — as then $u_\beta = u(p_\beta)$. In this latter case the equations are uncoupled and have for each p_β the form Eq. (1) — we have recovered the *collocation* method which independently for each p_β computes u_β, using the solver Eq. (2). Such a method then obviously is non-intrusive, as the original software may be used. Thus this is often the method of choice, as often there is considerable investment in the software which performs Eq. (2), which one would like to re-use. Unfortunately this choice is very rigid as regards the subspace $\mathcal{Q}_{\mathcal{A}}$ and the projection orthogonal to $\mathcal{R}_{\mathcal{A}}$.

We believe that this is a *false* alternative, and that the distinction is not between *intrusive or non-intrusive*, but between *coupled or uncoupled.* Furthermore and more importantly, we want to show that also in the more general case of a coupled system like in Eq. (10) the original solver Eq. (2) may be put to good use. This will be achieved by making Eq. (2) the starting point, instead of Eq. (1) or Eq. (3). Such coupled iterations also arise for example from multi-physics problem, and there these coupled iterations can also be solved by wht is called a *partitioned* approach, see e.g. [3], which is the equivalent of non-intrusive here. Quite a number of different variants of global partitioned iterations are possible [3], we only look at the simples variant, as the point is here only to dispel the myth about intrusiveness.

3.2 The fixed-point Galerkin equations

Whatever the starting point, we would still like to achieve the same result. So before continuing, let us show

Proposition 3.1. *Projecting the $\mathcal{A}$-residual Eq. (8) or the fixed point equation attached to the iteration Eq. (2), $u = S(u, R_{\mathcal{A}})$ gives the same conditions.*

Proof. For any $\beta \in \mathcal{A}$, we have

$$\begin{aligned} 0 = \boldsymbol{G}_p(\xi_\beta\,(S(u,R_{\mathcal{A}}) - u)) = \boldsymbol{G}_p(\xi_\beta\,((u + P^{-1}R_{\mathcal{A}}) - u)) = \boldsymbol{G}_p(\xi_\beta\, P^{-1}R_{\mathcal{A}}) = \\ 0 = P^{-1}\boldsymbol{G}_p(\xi_\beta\, R_{\mathcal{A}}) \quad \Leftrightarrow \quad 0 = \boldsymbol{G}_p(\xi_\beta\, R_{\mathcal{A}}), \end{aligned} \tag{12}$$

on noting that for any linear map L one has $\boldsymbol{G}_p(\phi(\cdot)\, Lv\, \varphi(\cdot)) = \langle \phi, \varphi\rangle_p\, Lv$, and by remembering that P and P^{-1} are non-singular. □

This means that instead of the residual Eq. (3) we may just as well project the iteration Eq. (5): for all β

$$\boldsymbol{G}_p(\xi_\beta(\cdot)\, u^{(k+1)}) = \boldsymbol{G}_p(\xi_\beta(\cdot)\,(u^{(k)} + \Delta u^{(k)})) = \boldsymbol{G}_p(\xi_\beta(\cdot)\,(u^{(k)} + P_k^{-1}R_{\mathcal{A}}^{(k)})). \tag{13}$$

Expanding $u^{(k)}(p) = \sum_\alpha u_\alpha^{(k)}\psi_\alpha(p)$ in Eq. (13), that becomes a coupled iteration equation for the u_α:

$$\forall\beta:\ \boldsymbol{G}_p(\xi_\beta(\cdot)\,\sum_\alpha u_\alpha^{(k+1)}\psi_\alpha(p)) = \boldsymbol{G}_p(\xi_\beta(\cdot)\,(\sum_\alpha u_\alpha^{(k)}\psi_\alpha(p) + P_k^{-1}R_{\mathcal{A}}^{(k)})). \tag{14}$$

As before, the choice $\xi_\beta(p) = \delta_\beta(p) = \delta(p - p_\beta)$ leads to the collocation method. Here we stay with the more general case, and Eq. (13) may now be written as

$$\forall\beta:\ \sum_\alpha \boldsymbol{M}_{\beta,\alpha}u_\alpha^{(k+1)} = \sum_\alpha \boldsymbol{M}_{\beta,\alpha}u_\alpha^{(k)} + \boldsymbol{G}_p(\xi_\beta\, P_k^{-1}R_{\mathcal{A}}^{(k)}), \tag{15}$$

where $\boldsymbol{M}_{\beta,\alpha} := \langle \xi_\beta, \psi_\alpha\rangle_p$. If the coefficients are arranged column-wise in a matrix $\mathbf{u} = [\dots, u_\alpha, \dots]$ — and similarly the $\boldsymbol{G}_p(\xi_\alpha \dots)$ as $\mathbf{G}_p(\mathbf{u})$ — and the $\boldsymbol{M}_{\beta,\alpha}$ are viewed as entries of a matrix $\mathbf{M} \in \mathbb{R}^{\mathcal{A}\times\mathcal{A}}$, Eq. (15) may be compactly written as $\mathbf{u}^{(k+1)}\mathbf{M}^T = \mathbf{u}^{(k+1)}\mathbf{M}^T + \mathbf{G}_p$. With the Kronecker product $\mathbf{L} \otimes_K \mathbf{E}$ of a matrix $\mathbf{L} \in \mathbb{R}^{\mathcal{A}\times\mathcal{A}}$ and a matrix $\mathbf{E}$ on $\mathcal{U}$ defined by its action on $\mathbf{u} \in \mathcal{U}^{\mathcal{A}}$ by $(\mathbf{L} \otimes_K \mathbf{E})\,\mathbf{u} := \mathbf{E}\,\mathbf{u}\,\mathbf{L}^T$, and observing that $(\mathbf{L} \otimes_K \mathbf{E})^{-1} = (\mathbf{L}^{-1} \otimes_K \mathbf{E}^{-1})$, we may write the iteration Eq. (15) now as

$$\begin{aligned} (\mathbf{M} \otimes_K \mathbf{I})\,\mathbf{u}^{(k+1)} &= (\mathbf{M} \otimes_K \mathbf{I})\,\mathbf{u}^{(k)} + \mathbf{G}_p(\mathbf{u}^{(k)}), && (16) \\ \Rightarrow \mathbf{u}^{(k+1)} &= (\mathbf{M} \otimes_K \mathbf{I})^{-1}\,\Big((\mathbf{M} \otimes_K \mathbf{I})\,\mathbf{u}^{(k)} + \mathbf{G}_p(\mathbf{u}^{(k)})\Big) = \mathbf{u}^{(k)} + \mathbf{G}_p(\mathbf{u}^{(k)})\,\mathbf{M}^{-T} && (17) \\ &= \mathbf{u}^{(k)} + (\mathbf{M} \otimes_K \mathbf{I})^{-1}\,\mathbf{G}_p(\mathbf{u}^{(k)}) = \mathbf{u}^{(k)} + \boldsymbol{\Delta}_p(\mathbf{u}^{(k)}) = \mathbf{S}_p(\mathbf{u}^{(k)}), && (18) \end{aligned}$$

where we have implicitly defined new functions $\mathbf{\Delta}_p$ and $\mathbf{S}_p$ in Eq. (18), which will be needed in Subsection 3.3.

It is apparent that the computation will be much simplified if the 'ansatz'-functions $\{\psi_\alpha\}_{\alpha\in\mathcal{A}}$ and the test-functions for the projection $\{\xi_\beta\}_{\beta\in\mathcal{A}}$ are chosen bi-orthogonal, i.e. if one has for all $\alpha, \beta \in \mathcal{A}$ that $\boldsymbol{M}_{\beta,\alpha} = \delta_{\beta,\alpha}$, i.e. $\mathbf{M} = \mathbf{I}$, which shall be assumed from now on.

Eq. (18) or its equivalent form Eq. (17) are already a possible way of performing the iteration. The practical, non-intrusive, computation of the terms in Eq. (18) still has to be considered, but we may formulate the corresponding algorithm and investigate its convergence.

Algorithm 3.1 Block Jacobi iteration of Eq. (17) or Eq. (18)

```
                                   ▷ %comment: Start with some initial guess $\mathbf{u}^{(0)}$%
$k \leftarrow 0$
while no convergence do
                                   ▷ %comment: the global iteration loop%
    Compute $\mathbf{\Delta}_p(\mathbf{u}^{(k)})$ according to Eq. (18)
    $\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \mathbf{\Delta}_p(\mathbf{u}^{(k)}) \quad [= \mathbf{S}_p(\mathbf{u}^{(k)})]$
    $k \leftarrow k + 1$
end while
```

Although the underlying iteration Eq. (2) in Algorithm 2.1 may be of any kind — e.g. Newton's method — when one views Eq. (18) with regard to the block structure imposed by the $\mathbf{u} = [\dots, u_\beta, \dots]$, it is a — maybe nonlinear — *block Jacobi* iteration, meaning that the right hand sides can be evaluated for all β by using only $\mathbf{u}^{(k)}$ of iteration k.

3.3 Convergence of coupled iterations

Here we want to show that the map $\mathbf{S}_p$ in Eq. (18) satisfies a Lipschitz conditions with the same constant as Eq. (6). This will need some more theoretical considerations.

As already stated, we assume that $\langle\cdot,\cdot\rangle_p$ is a duality pairing; for the sake of simplicity we will additionally assume that this is actually an inner product on some Hilbert space $\mathcal{Q} \subseteq \mathbb{R}^{\mathcal{P}}$, such that $\mathcal{Q}_\mathcal{A} \subseteq \mathcal{Q}$ and $\mathcal{R}_\mathcal{A} \subseteq \mathcal{Q}$. We also assume that a certain *inf-sup* condition is satisfied, namely

$$\inf_{\xi\in\mathcal{R}_\mathcal{A}} \sup_{\psi\in\mathcal{Q}_\mathcal{A}} \langle\xi,\psi\rangle_\mathcal{Q} \geq c > 0. \tag{19}$$

This implies that the matrix $\mathbf{M}$ in Eq. (15) is non-singular, and thus the expressions in Eq. (17) and Eq. (18) are well defined. Here we assume that $\mathbf{M}$ is the identity. As is well known, then the projection along $\mathcal{R}_\mathcal{A}^\perp$ given by:

Proposition 3.2. *For $\phi \in \mathcal{Q}$ find $\psi \in \mathcal{Q}_\mathcal{A}$ such that*

$$\forall\xi \in \mathcal{R}_\mathcal{A} : \quad \langle\xi,\psi\rangle_\mathcal{Q} = \langle\xi,\phi\rangle_\mathcal{Q} \tag{20}$$

has under the inf-sup-condition of Eq. (19) a unique solution satisfying $\|\psi\|_\mathcal{Q} \leq \frac{1}{c}\,\|\phi\|_\mathcal{Q}$.

In fact, this is nothing else than a simple version of the well-known *Babuška-Lions-Nečas* generalisation of the *Lax-Milgram* lemma. Observe that this projection is the essential ingredient in the definition of the map $\mathbf{G}_p : \mathcal{Q} \otimes \mathcal{U} \to \mathcal{U}^{\mathcal{A}}$ defined in Proposition 3.1 and Eq. (9) and after Eq. (15), and this will be used later in Lemma 3.3.

We equip the tensor product $\mathcal{Q} \otimes \mathcal{U}$ and its closed subspace $\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U}$ with the usual Hilbert space structure, such that for all $(\phi \otimes u), (\varphi \otimes v) \in \mathcal{Q} \otimes \mathcal{U}$ we have

$$\langle (\phi \otimes u), (\varphi \otimes v) \rangle_{\mathcal{Q}\otimes\mathcal{U}} := \langle \phi, \varphi \rangle_p \, \langle u, v \rangle_{\mathcal{U}}, \tag{21}$$

and denote the completion w.r.t. the corresponding norm again with the same symbol.

The mapping $\mathcal{U}^{\mathcal{A}} \ni \mathbf{u} = [\ldots, u_\alpha, \ldots] \mapsto \sum_\alpha u_\alpha \psi_\alpha(\cdot) \in \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \subseteq \mathcal{Q} \otimes \mathcal{U}$ is by design injective and may thus be used to induce a norm and inner product on $\mathcal{U}^{\mathcal{A}}$ via $\|\mathbf{u}\|_{\mathcal{U}^{\mathcal{A}}} := \| \sum_\alpha u_\alpha \psi_\alpha(\cdot) \|_{\mathcal{Q}\otimes\mathcal{U}}$, making it an isometrical injection $J : \mathcal{U}^{\mathcal{A}} \to \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U}$, so that for the simple elements $\mathbf{v} = [\ldots, 0, v_\beta, 0, \ldots], \mathbf{u} = [\ldots, 0, u_\alpha, 0, \ldots] \in \mathcal{U}^{\mathcal{A}}$

$$\langle \mathbf{v}, \mathbf{u} \rangle_{\mathcal{U}^{\mathcal{A}}} = \langle J\mathbf{v}, J\mathbf{u} \rangle_{\mathcal{Q}\otimes\mathcal{U}} = \langle (\psi_\beta \otimes v_\beta), (\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{Q}\otimes\mathcal{U}} = \langle \psi_\beta, \psi_\alpha \rangle_p \, \langle v_\beta, u_\alpha \rangle_{\mathcal{U}}, \tag{22}$$

then extended to the whole space by linearity.

Lemma 3.3. *The restricted mapping* $\mathbf{G}_p : \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \ni \sum_\alpha u_\alpha \psi_\alpha \mapsto [\ldots, u_\alpha, \ldots]$ *defined via Eq. (9) and after Eq. (15) coincides with the adjoint* J^* *on the range* $\operatorname{im} J = J(\mathcal{U}^{\mathcal{A}})$*. It thus has norm or Lipschitz constant less than unity on* $\operatorname{im} J$*. Otherwise the norm and hence Lipschitz constant of* $\mathbf{G}_p$ *on all of* $\mathcal{Q} \otimes \mathcal{U}$ *is bounded by* $1/c$*.*

Proof. The bound of $1/c$ on the whole space follows from Proposition 3.2. As J has norm one, so does its adjoint J^*. On im J, by definition $\mathbf{G}_p(\psi_\alpha \otimes u_\alpha) = \mathbf{u}$ and hence we have

$$\langle \mathbf{v}, J^*(\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{U}^{\mathcal{A}}} = \langle J\mathbf{v}, (\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{Q}\otimes\mathcal{U}} = \langle (\psi_\beta \otimes v_\beta), (\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{Q}\otimes\mathcal{U}},$$

which means that for all such $\mathbf{v}, \mathbf{u}$ we have $\langle \mathbf{v}, J^*(\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{U}^{\mathcal{A}}} = \langle \mathbf{v}, \mathbf{u} \rangle_{\mathcal{U}^{\mathcal{A}}}$ and also $\langle \mathbf{v}, \mathbf{u} \rangle_{\mathcal{U}^{\mathcal{A}}} = \langle \mathbf{v}, \mathbf{G}_p(\psi_\alpha \otimes u_\alpha) \rangle_{\mathcal{U}^{\mathcal{A}}}$, hence the two maps coincide on $\operatorname{im} J$: $J^* = \mathbf{G}_p$. The result then follows by linear extension and due to the isometry of J. □

This result also holds in the more general case when $\mathbf{M}$ is not the identity matrix, only then the statement is for the map $(\mathbf{M} \otimes_K \mathbf{I})^{-1} \circ \mathbf{G}_p$, cf. Eq. (17)

What needs to be shown to prove convergence, is that the map $\mathbf{S}_p$ in Eq. (18) is contractive. To this end observe that $\mathbf{S}_p : \mathcal{U}^{\mathcal{A}} \to \mathcal{U}^{\mathcal{A}}$ can be factored in the following way:

$$\mathbf{S}_p = \mathbf{G}_p \circ \tilde{S} \circ \iota \circ J, \tag{23}$$

$$\mathbf{S}_p : \mathcal{U}^{\mathcal{A}} \overset{J}{\to} \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \overset{\iota}{\hookrightarrow} \mathcal{Q} \otimes \mathcal{U} \overset{\tilde{S}}{\to} \mathcal{Q} \otimes \mathcal{U} \overset{\mathbf{G}_p}{\to} \mathcal{U}^{\mathcal{A}}, \tag{24}$$

where $\iota : \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \hookrightarrow \mathcal{Q} \otimes \mathcal{U}$ is simply the injection of a subspace, and $\tilde{S}$ is defined via the solver map S in Eq. (2) by

$$\tilde{S} : \mathcal{Q} \otimes \mathcal{U} \ni u(\cdot) \mapsto S(\cdot, u(\cdot), R(\cdot, u(\cdot))) \in \mathcal{Q} \otimes \mathcal{U}. \tag{25}$$

The maps J and ι in Eq. (24) are isometries, the Lipschitz constant of $\mathbf{G}_p$ has been determined in Lemma 3.3, so what is left is to look at $\tilde{S}$:

Theorem 3.4. *The map $\tilde{S}$ from Eq. (25) has the same Lipschitz constant ϱ^* as the map S in Eq. (2), cf. Eq. (6).*

Proof. We now use the assumption that the inner product on $\mathcal{Q}$ is given by an integral, for the sake of simplicity we just consider the case $\langle \phi, \varphi \rangle_p = \int_{\mathcal{P}} \phi(p)\varphi(p)\,\mu(\mathrm{d}p)$. In that case $\mathcal{Q} = L_2(\mathcal{P}, \mu; \mathbb{R})$, and the Hilbert tensor product $\mathcal{Q} \otimes \mathcal{U}$ is isometrically isomorphic to $L_2(\mathcal{P}, \mu; \mathcal{U})$. Hence with Eq. (6) for all $u(\cdot), v(\cdot) \in L_2(\mathcal{P}, \mu; \mathcal{U})$

$$\begin{aligned}\|\tilde{S}(u(\cdot)) - \tilde{S}(v(\cdot))\|^2_{L_2(\mathcal{P},\mu;\mathcal{U})} &= \int_{\mathcal{P}} \|S(p, u(p), R(p, u(p))) - S(p, v(p), R(p, v(p)))\|^2_{\mathcal{U}}\, \mu(\mathrm{d}p) \\ &\le (\varrho^*)^2 \int_{\mathcal{P}} \|u(p) - v(p)\|^2_{\mathcal{U}}\, \mu(\mathrm{d}p) = (\varrho^*)^2 \|u(\cdot) - v(\cdot)\|^2_{L_2(\mathcal{P},\mu;\mathcal{U})},\end{aligned}$$

and the proof is concluded by taking square roots. □

This immediately leads to

Corollary 3.5. *The map $\mathbf{S}_p$ from Eq. (18) has Lipschitz constant $L = \varrho^*/c$.*

Proof. This follows from the decomposition Eq. (23), Lemma 3.3, and the fact that J and ι are isometries. □

Let $\mathcal{R}_{\mathcal{A}}^{\perp}$ be the orthogonal complement of $\mathcal{R}_{\mathcal{A}}$, then $\mathbf{G}_p$ has nullspace $\ker \mathbf{G}_p = \mathcal{R}_{\mathcal{A}}^{\perp} \otimes \mathcal{U}$, and $\mathcal{Q} \otimes \mathcal{U} = \ker \mathbf{G}_p \oplus (\ker \mathbf{G}_p)^{\perp} = (\mathcal{R}_{\mathcal{A}}^{\perp} \otimes \mathcal{U}) \oplus (\mathcal{R}_{\mathcal{A}} \otimes \mathcal{U})$. If this orthogonal decomposition were $\mathcal{Q} \otimes \mathcal{U} = (\mathcal{Q}_{\mathcal{A}}^{\perp} \otimes \mathcal{U}) \oplus (\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U})$, one could just replace $\mathbf{G}_p$ by $J^* \circ \iota^*$ in the factorisation Eq. (23), as on $\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U}$ it coincides with J^* according to Lemma 3.3. This is achieved by

Theorem 3.6. *If one chooses $\mathcal{R}_{\mathcal{A}} (= \mathrm{span}\{\xi_\beta\}_{\beta \in \mathcal{A}})$ equal to $\mathcal{Q}_{\mathcal{A}} (= \mathrm{span}\{\psi_\alpha\}_{\alpha \in \mathcal{A}})$, the map $\mathbf{S}_p$ from Eq. (18) has Lipschitz constant $L = \varrho^*$, and is thus a contraction with the same factor as the solver S in Eq. (2), i.e. the algorithm 3.1 converges with the same linear speed of convergence as algorithm 2.1.*

Proof. In case $\mathcal{R}_{\mathcal{A}} = \mathcal{Q}_{\mathcal{A}}$, we have that $\ker \mathbf{G}_p = \mathcal{Q}_{\mathcal{A}}^{\perp} \otimes \mathcal{U} = (\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U})^{\perp}$, the whole space can be decomposed as $\mathcal{Q} \otimes \mathcal{U} = \ker \mathbf{G}_p \oplus (\ker \mathbf{G}_p)^{\perp} = (\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U})^{\perp} \oplus (\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U})$, and on $\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U}$ according to Lemma 3.3 $\mathbf{G}_p$ coincides with J^*. Hence in this case the factorisation Eq. (24) is the same as

$$\begin{aligned}&\mathbf{S}_p = J^* \circ \iota^* \circ \tilde{S} \circ \iota \circ J, \\ &\mathbf{S}_p : \mathcal{U}^{\mathcal{A}} \xrightarrow{J} \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \overset{\iota}{\hookrightarrow} \mathcal{Q} \otimes \mathcal{U} \xrightarrow{\tilde{S}} \mathcal{Q} \otimes \mathcal{U} \xrightarrow{\iota^*} \mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} \xrightarrow{J^*} \mathcal{U}^{\mathcal{A}},\end{aligned}$$

where $\mathbf{G}_p$ has been replaced by $J^* \circ \iota^*$. Both J and ι are isometrical injections, hence their adjoints J^* and ι^* have the same Lipschitz constant equal to unity — they are or may be seen as equivalent to orthogonal projections, ι^* is the orthogonal projection from $\mathcal{Q} \otimes \mathcal{U}$ on $\mathcal{Q}_{\mathcal{A}} \otimes \mathcal{U} = (\ker \mathbf{G}_p)^{\perp}$. Hence, due to the above factorisation, $\mathbf{S}_p$ and $\tilde{S}$ have the same Lipschitz constant. Effectively, with the Bubnov-Galerkin choice $\mathcal{R}_{\mathcal{A}} = \mathcal{Q}_{\mathcal{A}}$, the constant c in Eq. (19) in Proposition 3.2 is $c = 1$, as the projection is orthogonal. Invoking now Corollary 3.5 finishes the proof. □

Observe that this only holds for the linear convergence speed, in case algorithm 2.1 has super-linear convergence, this can not be necessarily matched by algorithm 3.1, and more sophisticated algorithms are necessary, e.g. [3].

4 Non-intrusive residual

One may observe that the term $\boldsymbol{G}_p(\xi_\alpha\, P_k^{-1} R_{\mathcal{A}}^{(k)})$ in Eq. (15) is the preconditioned residual for that iteration. Let us recall that the Galerkin projector was defined by $\boldsymbol{G}_p(\phi(\cdot)\, v\, \varphi(\cdot)) = \langle \phi, \varphi \rangle_p\, v$, and assume that the duality pairing on the scalar functions is given by some kind of integral with measure μ,

$$\langle \phi, \varphi \rangle_p = \int_{\mathcal{P}} \phi(p)\varphi(p)\, \mu(\mathrm{d}p). \tag{26}$$

This integral is assumed to have some approximate numerical quadrature formula

$$\int_{\mathcal{P}} \phi(p)\, \mu(\mathrm{d}p) \approx \sum_z w_z \phi(p_z), \tag{27}$$

where the integrand is evaluated at the quadrature points p_z and the w_z are appropriate weights.

With this approximation the term $\boldsymbol{G}_p(\xi_\beta\, P_k^{-1} R_{\mathcal{A}}^{(k)})$ in Eq. (15) becomes practically computable, giving

$$\boldsymbol{G}_p(\xi_\beta\, P_k^{-1} R_{\mathcal{A}}^{(k)}) \approx \Delta_q u_\beta^{(k)} := \sum_z w_z \xi_\beta(p_z)\, \Delta u_z^{(k)}, \quad \text{where} \tag{28}$$

$$\Delta u_z^{(k)} := P_k^{-1}(p_z, u^{(k)}(p_z)) R_{\mathcal{A}}^{(k)}(p_z, u^{(k)}(p_z)) = P_k^{-1}(p_z) \left(f(p_z) - A(p_z; u^{(k)}(p_z)) \right) \tag{29}$$

is the preconditioned residuum evaluated at p_z, and $u^{(k)}(p_z) = \sum_\alpha u_\alpha^{(k)} \psi_\alpha(p_z)$. This is indeed the only interface needed to the original equation, something which can be easily evaluated *non-intrusively* as the iteration increment $\Delta u_z^{(k)}$ in Eq. (29) in case the current state is given as $u^{(k)}(p_z)$.

5 Iteration

See [3].

Rewriting Eq. (15) for the case of a bi-orthogonal system, the coupling between different α's on the left hand side disappears — the equations are generally still coupled through the term $R_{\mathcal{A}}^{(k)}$ — and one obtains with the numerical integration and Eq. (28)

$$\forall \beta : \; u_\beta^{(k+1)} = u_\beta^{(k)} + \Delta_q u_\beta^{(k)}. \tag{30}$$

Algorithm 5.1 Block Jacobi iteration of Eq. (30)

```
                                    ▷ %comment: Start with some initial guess $[\dots, u_\beta^{(0)}, \dots]$%
$k \leftarrow 0$
while no convergence do
                                    ▷ %comment: the global iteration loop%
    for each $\beta$ do
                                    ▷ %comment: pick a number $J$ of inner iterations%
        for $j \leftarrow 1, \dots, J$ do
            Compute $\Delta_q u_\beta^{(k)}$ according to Eq. (28)
            $u_\beta^{(k)} \leftarrow u_\beta^{(k)} + \Delta_q u_\beta^{(k)}$
        end for
    end for
    $k \leftarrow k + 1$
end while
```

Usually the nonlinear *block Gauss-Seidel* variant for the same set-up will converge faster. To describe it, the abbreviations

$$r_z^{(k)}(v_j) := f(p_z) - A(p_z; \sum_{\beta<\alpha} u_\beta^{(k+1)}\psi_\beta(p_z) + v_j\psi_\alpha(p_z) + \sum_{\beta>\alpha} u_\beta^{(k)}\psi_\beta(p_z))) \quad (31)$$

$$\text{and } \Delta_{GS} u_\alpha^{(k)}(v_j) := \sum_z w_z \boldsymbol{G}_p(\xi_\alpha(p_z)\, P_k^{-1}(p_z) r_z^{(k)}(v_j)) \quad (32)$$

are quite useful, where v_j is the as yet unspecified coefficient of the function ψ_α. The relation $\alpha < \beta$ and similar refers to the chosen ordering on $\mathcal{A}$ alluded to before. Please observe that $\Delta_{GS} u_\alpha^{(k)}(v_j)$ is a component of a projected preconditioned residual which may be computed *non-intrusively* by sampling preconditioned residuals.

6 Conclusion

The problem of parameters

References

[1] H. G. Matthies, *Uncertainty quantification with stochastic finite elements*, Encyclopaedia of Computational Mechanics (E. Stein, R. de Borst, and T. J. R. Hughes, eds.), John Wiley & Sons, Chichester, 2007, doi:10.1002/0470091355.ecm071.

[2] H. G. Matthies and A. Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Computer Methods in Applied Mechanics and Engineering **194** (2005), no. 12-16, 1295–1331. MR MR2121216 (2005j:65146)

Algorithm 5.2 Block Gauss-Seidel iteration of Eq. (30)

```
                              ▷ %comment: Start with some initial guess $[\dots, u_\alpha^{(0)}, \dots]$%
$k \leftarrow 0$
while no convergence do
                              ▷ %comment: the global iteration loop%
    for each $\alpha$ in chosen order do
                              ▷ %comment: the block Gauss-Seidel sweep%
        $v_0 \leftarrow u_\alpha^{(k)}$
                              ▷ %comment: pick a number $J$ of inner iterations%
        for $j \leftarrow 1, \dots, J$ do
            Compute $\Delta_{GS} u_\alpha^{(k)}(v_{j-1})$ according to Eq. (32)
            $v_j \leftarrow v_{j-1} + \Delta_{GS} u_\alpha^{(k)}(v_{j-1})$
        end for
        $u_\alpha^{(k+1)} \leftarrow v_J$
    end for
    $k \leftarrow k + 1$
end while
```

[3] H. G. Matthies, R. Niekamp, and J. Steindorf, *Algorithms for strong coupling procedures*, Computer Methods in Applied Mechanics and Engineering **195** (2006), no. 17-18, 2028–2049, Available from: `http://dx.doi.org/10.1016/j.cma.2004.11.032`, `doi:10.1016/j.cma.2004.11.032`. MR 2202913 (2006h:74023)

[4] H. G. Matthies, A. Litvinenko, O. Pajonk, B. V. Rosić, and E. Zander, *Parametric and uncertainty computations with tensor product representations*, Uncertainty Quantification in Scientific Computing (Berlin) (A. Dienstfrey and R. Boisvert, eds.), IFIP Advances in Information and Communication Technology, vol. 377, Springer, 2012, pp. 139–150.