LEAST-SQUARES PROPER GENERALISED DECOMPOSITIONS FOR ELLIPTIC SYSTEMS

THOMAS L. D. CROFT† AND TIMOTHY N. PHILLIPS†

Abstract. Proper generalised decompositions (PGDs) are a family of methods for efficiently solving high-dimensional PDEs. Convergence of PGD algorithms can be proven provided that the weak form of the PDE can be recast as the minimisation of some energy functional. A large number of elliptic problems, such as the Stokes problem, cannot be guaranteed to converge when employing a Galerkin PGD. Least-squares methods are derived from the minimisation of the residual of the differential operator under a carefully selected norm. This provides an ‘artificial’ energy functional with which convergence of least-squares PGDs can be proven for all elliptic problems. In this paper robust least-squares PGD algorithms for the Poisson and Stokes problems are constructed and a comparison of the efficiency of different formulations is given.

Key words. least-squares methods, proper generalized decomposition

AMS subject classifications. 65N12, 65N30

1. Introduction. The main concept underpinning all PGD algorithms (see Ammar et al. [3], for example) is the approximation of the solution, \( u \), to a \( d \)-dimensional PDE by the separated representation:

\[
    u(x_1, \ldots, x_d) \approx \sum_{j=1}^{J} F_{1,j}(x_1) \times \cdots \times F_{d,j}(x_d),
\]

where \( J \) is known as the rank of the PGD approximation with the idea being that, as \( J \to \infty \), the separated representation approaches the true solution. The importance of this separated representation in the numerical analysis of high-dimensional problems was first noted by Beylkin and Mohlenkamp [7]. In the case of the PGD it can drastically reduce the number of unknowns required to solve high-dimensional PDEs. Indeed, given an \( N \)-node discretisation, the number of unknowns in a standard mesh based approximation of the solution to a \( d \)-dimensional PDE is \( N^d \) whereas for the PGD one would have \( N \times J \times d \) unknowns. In other words the complexity of a mesh based approximation scales exponentially with increasing dimension whereas the PGD scales linearly; clearly a vast improvement.

There are a variety of PGD algorithms (see Nouy [24]) but in this paper we only consider the simplest PGD algorithm: the progressive PGD. This algorithm consists of iteratively finding the ‘best’ rank-one separated representation of the true solution to the given PDE, \( F_{1,j}(x_1) \times \cdots \times F_{d,j}(x_d) \), for each \( j = 1, \ldots, J \). These rank-one separated representations are known as the PGD modes. Previously calculated PGD modes are simply moved to the right hand side of the equation at which point the next PGD mode is calculated in the subsequent iteration. For Galerkin PGDs the ‘best’ PGD mode is simply the one which satisfies Galerkin orthogonality with an appropriate choice of test function. This leads to a non-linear system which can be solved using an alternating directions fixed point linearisation. We direct the interested reader to an in-depth description of the algorithm for \( d = 3 \) by Chinesta et al. [16].

†School of Mathematics, Cardiff University, Cardiff, Wales, UK. (CroftT@cf.ac.uk, PhillipsTN@cf.ac.uk).
Le Bris et al. [21] have recently associated PGDs with greedy algorithms, the likes of which have been studied by Temlyakov [29]. The ‘best’ choice of PGD mode for these greedy algorithms is the one which minimises a certain functional which is equivalent to the weak solution of the PDE. For strongly coercive variational problems, such as the Poisson equation, this could be its associated Rayleigh-Ritz setting. The advantage of considering a PGD in this way is that it allows one to prove convergence. Unfortunately, this means that for variational problems which are not strongly coercive in which no equivalent minimisation principle exists, such as the Stokes problem, it is not possible to prove convergence of an associated Galerkin PGD algorithm. Furthermore, for problems like the saddle-point Stokes problem, stability conditions such as the LBB condition cannot be guaranteed to hold in the PGD framework. This provides the motivation for this paper in which we focus on PGDs based on least-squares methods instead of Galerkin PGDs.

The main idea behind least-squares methods is the minimisation of the residual of a differential operator in a carefully selected norm. This provides one with an ‘artificial’ Rayleigh-Ritz type setting. Provided that certain coercivity estimates hold on the chosen norms, this can be used to prove convergence of an associated greedy algorithm. The convexity of the least-squares formulation has the added benefit that it bypasses stability conditions such as the LBB condition. This is similar to the notion of the minimal residual PGD (e.g. [14, 24]). However, we use the terminology ‘least-squares PGD’ to highlight the fact that we construct PGD algorithms based on rigorously defined least-squares principles. We shall investigate questions such as: ‘Which choice of norm leads to the most efficient least-squares PGDs?’ and ‘How do least-squares PGDs compare with Galerkin PGDs?’.

This paper is structured as follows: In Section 2 we introduce the abstract formulation of least-squares methods with some additional comments on their inclusion in the PGD. In Section 3 we provide outlines of two different proofs of convergence of least-squares PGDs in the abstract form of Section 2. In Sections 4 and 5 we construct a variety of least-squares methods for the Poisson equation and Stokes problem, respectively. In these sections we also present numerical results for each of the least-squares PGD algorithms. Conclusions and future work are discussed in Section 6.

1.1. Notation. In what follows we have used the notation that \( \| \cdot \| \), and \( \| \cdot \|_{k, \Gamma} \) denote the norms on \( H^k(\Omega) \) and \( H^k(\Gamma) \) respectively, and similarly for the inner products \( \langle \cdot, \cdot \rangle \) and \( \langle \cdot, \cdot \rangle_{k, \Gamma} \).

2. Abstract Least-Squares Formulation. Consider the following abstract boundary value problem

\[
\begin{align*}
L u &= f \quad \text{in } \Omega, \\
R u &= g \quad \text{on } \Gamma = \partial \Omega,
\end{align*}
\]

where \( L \) is a linear elliptic partial differential operator, \( R \) is a trace operator and \( f \) and \( g \) are given functions. We further assume that \( L \) is a first-order differential operator since we can recast any higher order problem into equivalent systems of first-order differential equations and in the least-squares method we need to do this in order to construct a practical method. We elaborate on this later.

Now if we also assume the above boundary value problem is well-posed and that there exists a homeomorphism \( \{ L, R \} : X \rightarrow Y \times Z \) where \( X = X(\Omega), Y = Y(\Omega) \) and \( Z = Z(\Gamma) \) are some underlying Hilbert spaces with norms \( \| \cdot \|_X, \| \cdot \|_Y \) and \( \| \cdot \|_Z, \)
respectively, then there exist constants $C_1, C_2 > 0$ such that:

\begin{equation}
C_1\|u\|_X \leq \|Lu\|_Y + \|Ru\|_Z \leq C_2\|u\|_X, \quad \forall u \in X.
\end{equation}

If we let $\tilde{u}$ denote the unique solution of (2.1)-(2.2) then using the inequality (2.3) we can write

\begin{equation}
C_1\|u - \tilde{u}\|_X \leq \|Lu - f\|_Y + \|Ru - g\|_Z \leq C_2\|u - \tilde{u}\|_X, \quad \forall u \in X.
\end{equation}

This norm equivalence between the error in the $X$-norm and the residual in the differential equation in the $Y \times Z$-norm is termed the coercivity estimate (or a priori estimate) and it is the key ingredient in the analysis of least-squares methods. This is due to the fact that (2.4) implies that if we had a sequence of functions $u_n \in X$ such that $\|Lu_n - f\|_Y \to 0$ and $\|Ru_n - g\|_Z \to 0$ as $n \to \infty$ then $\|u_n - \tilde{u}\|_X \to 0$ as $n \to \infty$ and vice versa. This means that the sequence $u_n$ converges to the true solution in the $X$-norm. Therefore, minimisation of the convex functional:

\begin{equation}
J(u) = \|Lu - f\|_Y^2 + \|Ru - g\|_Z^2, \quad \forall u \in X,
\end{equation}

yields the unique solution $\tilde{u}$ to the boundary value problem (2.1)-(2.2). In fact this functional $J$ is the previously mentioned quadratic least-squares functional and it has been proven (see e.g. [10]) that the unique minimiser of $J$ is the unique solution $\tilde{u}$. We are then able to derive the Euler-Lagrange equation associated with the minimisation of (2.5): Find $u \in X$ such that:

\begin{equation}
\lim_{\epsilon \to 0} \frac{d}{d\epsilon} J(u + \epsilon v) = 0, \quad \forall v \in X
\end{equation}

The minimisation process leads to the following variational formulation: Find $u \in X$ such that:

\begin{equation}
A(u, v) = L(v), \quad \forall v \in X
\end{equation}

where

\begin{equation}
A(u, v) = \langle Lu, Lv \rangle_Y + \langle Ru, Rv \rangle_Z, \quad L(v) = \langle f, Lv \rangle_Y + \langle g, Rv \rangle_Z.
\end{equation}

and $\langle \cdot, \cdot \rangle_Y$ and $\langle \cdot, \cdot \rangle_Z$ denote the $Y$ and $Z$ inner products, respectively. We can see from this that least-squares methods always yield symmetric linear systems. This is a major advantage of least-squares methods since it means one is able to use robust iterative solvers such as the conjugate gradient method for problems which may not yield symmetric systems in the Galerkin formulation of the same problem.

At the beginning of this section we made the assumption that Hilbert spaces $X$, $Y$ and $Z$ exist and that they provide a homeomorphism $\{L, R\} : X \to Y \times Z$. The difficulty lies in choosing suitable Hilbert spaces so that the problem is well-defined and such a homeomorphism exists. One way of verifying these assumptions is to employ the theory of Agmon, Douglis and Nirenberg (ADN theory) [1,2] the elements of which are most relevant to least-squares methods are summarised by Aziz et al. [6]. This is a particularly powerful tool since it reduces the verification of continuous estimates to the verification of some algebraic constraints. One can then use this to obtain the lower bound in the coercivity estimate (2.3) whereas the upper bound comes simply from the continuity of the operators $L$ and $R$. We will not detail the ADN theory in this paper but all estimates that we present for the Poisson and Stokes problems can be verified in this way. We shall now highlight some practical considerations when employing least-squares methods.
2.1. Practical Issues. We mentioned earlier that for least-squares formulations to be a practical method for approximating the solution of differential equations it is necessary to first recast the problem as a first-order system. The reason for this is clear if we assume that, in the above material, \( L \) is a second-order differential operator and \( Y = L^2(\Omega) \). The variational formulation (2.6) then requires one to evaluate the inner product \( \langle Lu, Lv \rangle_0 \). Notice that the differential operator appears in both arguments of the inner product and hence, unlike in Galerkin formulations, we cannot weaken the required differentiability on the space \( X \) via Green's first integral identity. This means that we would need to provide a conforming discrete space \( X_h \subset X \) that is \( C^1 \)-continuous. This is a property that is not satisfied by standard linear finite/spectral element approximations over element edges. However, while there are finite element methods that can be constructed which are \( C^1 \)-continuous on element edges, they tend to be difficult to work with and are impractical [28]. It is often the case, for example in Galerkin formulations of fourth-order problems such as the equations governing plate bending, that non-conforming finite element methods are used instead such that \( X_h \not\subset X \). However, in least-squares methods it becomes unclear how the norm equivalence in the coercivity estimate (2.4) is affected if non-conforming discrete spaces are used [10]. It is for this reason that we must recast the problem into a system of first-order differential equations so that \( L \) is a first-order differential operator.

Secondly, we also require that the differentiability of the spaces \( X \) and \( Y \) do not exceed 1 and 0, respectively. Indeed, if, for example, \( X = H^2(\Omega) \) and/or \( Y = H^1(\Omega) \) then a conforming discrete space \( X_h \subset X \) would again need to be \( C^1 \)-continuous due to the order of derivatives that appear in the \( H^2 \) and \( H^1 \)-norms.

Systems that do not suffer from these two practical issues are called homogeneous elliptic (a more formal definition can be found in the ADN theory [2]). For non-homogeneous elliptic systems, the least-squares method can still be made practical by extending the coercivity estimate to hold for Sobolev spaces with negative index. Unfortunately, this has the side-effect of introducing negative norms into the least-squares functional. These negative norms are problematic to work with. Indeed, the minus one norm given by (see e.g. [10]):

\[
\|f\|_{-1} = \sup_{u \in H^1_0(\Omega)} \frac{\langle f, u \rangle_0}{\|u\|_1},
\]
does not lend itself to being computed easily using a finite/spectral element method.

Besides these negative norms we also have problematic trace norms (i.e. the \( Z \)-norm). Indeed, trace norms on fractional Sobolev spaces are defined by (see e.g. [10]):

\[
\|u\|_{s-1/2} = \inf_{v \in H^s_0(\Omega)} \|v\|_s,
\]
where \( H^s_0(\Omega) = \{ v \in H^s(\Omega) : v = g \text{ on } \Gamma \} \). This norm is not easily computed using finite/spectral element methods. To make these problematic norms more practical we replace them by specially weighted \( L^2 \)-norms. The fundamental idea which allows us to do this is that in finite dimensional spaces all norms are equivalent so that when the problem is discretised the continuous coercivity estimate from Theorem 2.5 is somehow preserved. More formally, consider the norm generating operators \( S_Y \) and \( S_Z \) such that we can rewrite the energy norm, \( \||·|| \), in terms of \( L^2 \)-norms:

\[
\|u\| := \|Lu\|_Y + \|Ru\|_Z = \|S_Y \circ Lu\|_0 + \|S_Z \circ Ru\|_{0,\Gamma}.
\]
We can compute discrete approximations for the norm generating operators such that we have the following discrete energy norm:

$$\| u_h \| := \| S_Y^h \circ L^h u_h \|_0 + \| S_Z^h \circ R^h u_h \|_{0,\Gamma}.$$ 

The choice of approximations for the discrete norm generating operators can lead to two distinct cases. Firstly we can retain the norm equivalence in the discrete energy norm, so that there exists a constant $c > 0$ such that:

$$c \| u_h \|_X \leq \| u_h \|_h \leq c \| u_h \|_X.$$ 

This is the most desirable situation since the coercivity estimate is preserved and optimal rates of convergence in $h$ can be achieved [10]. Secondly we can obtain only quasi-norm equivalence in the discrete energy norm such that the constants in the norm equivalence now depend on the mesh parameter $h$, i.e. we have that:

$$c(h) \| u_h \|_X \leq \| u_h \|_h \leq c(h) \| u_h \|_X.$$ 

The dependence on $h$ means that it becomes unclear, as $h \to 0$, how well the discrete energy norm $\| \cdot \|_h$ represents the true energy norm $\| \cdot \|$. In particular, this means that, in general, optimal rates of convergence in $h$ are not guaranteed and also it can lead to high condition numbers [10]. It is also unclear in the context of the PGD how this discrete norm equivalence is affected by the rank, $J$, of the PGD approximation.

To simplify things we shall only consider Dirichlet boundary conditions on the whole of $\Gamma$ in this paper. The reason for this being that we can impose these boundary conditions strongly and hence can avoid difficulties introduced by approximating the norm generating operators $S_Z$. To this end we define the space, $X_g$, whose elements satisfy the boundary conditions

$$X_g = \{ u \in X : Ru = g \text{ on } \Gamma \},$$

hence $\| Ru - g \|_Z = 0$ for $u \in X_g$. This leads to the simplified quadratic least squares functional:

$$J(u) = \| Lu - f \|_Y^2, \quad \forall u \in X_g.$$ 

We are then able to derive the Euler-Lagrange equations associated with the minimisation of (2.7): Find $u \in X_g$ such that:

$$A(u, v) = L(v), \quad \forall v \in X_0$$

where

$$A(u, v) = \langle Lu, Lv \rangle_Y, \quad L(v) = \langle f, Lv \rangle_Y.$$ 

A final practical issue we must mention is an issue related to the implementation of least-squares methods into the PGD. The issue is that if we have a problem defined in high-dimensional space then a first-order reformulation of such a problem has a much larger number of dependent variables. For example, given a 100-dimensional Poisson equation, a Div-Grad type formulation of this problem would have 101 dependent variables. This means that the number of unknowns in a least-squares PGD algorithm no longer grows linearly as the dimension increases. Indeed, consider a Galerkin PGD formulation of a problem in $d$-dimensional space which has $N \times J \times d$ unknowns. A
least-squares formulation of the same problem would instead have \(N \times J \times d \times d\alpha\) unknowns, where \(d\alpha\) represents the rate at which the number of dependent variables increase with the dimension \(d\). This means that for least-squares PGD algorithms we will obtain, at best, a quadratic rate of increase of the number of unknowns as the dimension increases. While this is certainly worse than for Galerkin PGD algorithms it is still a vast improvement over the exponential rate of increase one would obtain with a standard mesh based approach.

We now turn our attention to convergence of least-squares PGD algorithms. One of the key reasons that we are interested in using least-squares formulations in conjunction with the PGD is that they provide us with a minimisation principle of an artificial energy functional. This enables us to define associated greedy algorithms for which convergence can be proved.

3. Convergence of Least-Squares PGD Algorithms. We aim to prove that least-squares PGD algorithms converge for all problems which fit into the abstract theory covered in the last section (i.e. all linear ADN elliptic problems). We shall consider a proof based on minimisation of energies by Cancès et al. [13] as well as a proof based on a functional Eckart-Young theorem by Falcó and Nouy [18].

3.1. Energy Minimisation. The fundamental issue that prevents us from providing a proof of the convergence of a Galerkin PGD algorithm for problems which are not strongly coercive is the absence of an energy minimisation principle that could be used to define an associated greedy algorithm. The least-squares PGD algorithm has overcome this by providing us with an artificial energy functional:

\[
\mathcal{J}(u) = \|Lu - f\|_Y^2.
\]

To ensure that we can include least-squares PGD problems into the general theoretical setting outlined in [13], in which we are able to prove convergence of the associated greedy algorithms, this artificial energy functional must satisfy two conditions:

1. \(\mathcal{J}\) is strongly convex for \(\| \cdot \|_X\) i.e. there exists a constant \(\alpha > 0\) such that for \(t \in [0, 1]\):

\[
\mathcal{J}(tu + (1 - t)v) \leq t\mathcal{J}(u) + (1 - t)\mathcal{J}(v) - \frac{\alpha}{2} t(1 - t)\|u - v\|_X^2, \quad \forall u, v \in X.
\]

We then say that \(\mathcal{J}\) is \(\alpha\)-convex [20].

2. \(\mathcal{J}\) is differentiable and its Fréchet derivative is Lipschitz continuous i.e. there exists a constant \(L \geq 0\) such that

\[
\|\mathcal{J}'(u) - \mathcal{J}'(v)\|_X \leq L\|u - v\|_X, \quad \forall u, v \in X,
\]

where \(\mathcal{J}'\) denotes the Fréchet derivative of \(\mathcal{J}\).

Lemma 3.1. The least-squares functional (3.1) satisfies both the above conditions.

Proof. The key ingredient to proving that these two conditions hold for the least-squares functional is the coercivity relation arising from the ADN theory

\[
C_1\|u\|_X \leq \|Lu\|_Y \leq C_2\|u\|_X, \quad \forall u \in X.
\]

Indeed, since we know that

\[
\|u - v\|_X^2 \geq \frac{1}{C_2^2}\|Lu - Lv\|_Y^2 = \frac{1}{C_2^2}\|Lu -Lv\|_Y^2,
\]

\[
\mathcal{J}(u) = \|Lu - f\|_Y^2.
\]

To ensure that we can include least-squares PGD problems into the general theoretical setting outlined in [13], in which we are able to prove convergence of the associated greedy algorithms, this artificial energy functional must satisfy two conditions:

1. \(\mathcal{J}\) is strongly convex for \(\| \cdot \|_X\) i.e. there exists a constant \(\alpha > 0\) such that for \(t \in [0, 1]\):

\[
\mathcal{J}(tu + (1 - t)v) \leq t\mathcal{J}(u) + (1 - t)\mathcal{J}(v) - \frac{\alpha}{2} t(1 - t)\|u - v\|_X^2, \quad \forall u, v \in X.
\]

We then say that \(\mathcal{J}\) is \(\alpha\)-convex [20].

2. \(\mathcal{J}\) is differentiable and its Fréchet derivative is Lipschitz continuous i.e. there exists a constant \(L \geq 0\) such that

\[
\|\mathcal{J}'(u) - \mathcal{J}'(v)\|_X \leq L\|u - v\|_X, \quad \forall u, v \in X,
\]

where \(\mathcal{J}'\) denotes the Fréchet derivative of \(\mathcal{J}\).

Lemma 3.1. The least-squares functional (3.1) satisfies both the above conditions.

Proof. The key ingredient to proving that these two conditions hold for the least-squares functional is the coercivity relation arising from the ADN theory

\[
C_1\|u\|_X \leq \|Lu\|_Y \leq C_2\|u\|_X, \quad \forall u \in X.
\]

Indeed, since we know that

\[
\|u - v\|_X^2 \geq \frac{1}{C_2^2}\|Lu - Lv\|_Y^2 = \frac{1}{C_2^2}\|Lu -Lv\|_Y^2,
\]
then proving strong convexity amounts to proving that for \( t \in [0, 1] \):

\[
J(tu + (1-t)v) \leq tJ(u) + (1-t)J(v) - \frac{\alpha}{2C_2^2} t(1-t)\|Lu - Lv\|_Y^2, \quad \forall u, v \in X.
\]

Indeed, if we consider the left hand side of (3.3):

\[
J(tu + (1-t)v) = \|tLu + (1-t)Lv - f\|_Y^2 \\
= t^2\|Lu\|_Y^2 + (1-t)^2\|Lv\|_Y^2 + \|f\|_Y^2 - 2t(Lu, f)_Y \\
- 2(1-t)(Lv, f)_Y + 2t(1-t)(Lu, Lv)_Y \\
= t(\|Lu\|_Y^2 - 2(Lu, f)_Y + \|f\|_Y^2) \\
+ (1-t)(\|Lv\|_Y^2 - 2(Lv, f)_Y + \|f\|_Y^2) \\
- t(1-t)(\|Lu\|_Y^2 - 2(Lu, Lv)_Y + \|Lv\|_Y^2) \\
= t\|Lu - f\|_Y^2 + (1-t)\|Lv - f\|_Y^2 - t(1-t)\|Lu - Lv\|_Y^2
\]

which is the right-hand side of (3.3) with \( \alpha = 2C_2^2 \). Hence \( J \) is \( 2C_2^2 \)-convex.

For the second part of the proof we do not need to evaluate explicitly the Fréchet derivative \( J' \). Instead we use the fact that the functional derivative, which is exactly the Euler-Lagrange equation associated with the minimisation of \( J \), is equal to the \( X \)-inner product of its Fréchet derivative with a test function. More precisely we know that:

\[
\langle J'(u), w \rangle_X = (Lw, Lu - f)_Y, \quad \forall w \in X.
\]

For all \( u, v, w \in X \) we have:

\[
|\langle J'(u) - J'(v), w \rangle_X| = |\langle J'(u), w \rangle_X - \langle J'(v), w \rangle_X| \\
= |(Lw, Lu - f)_Y - (Lw, Lv - f)_Y| \\
= |(Lw, Lu - Lv)_Y|,
\]

and by Cauchy-Schwarz we have that

\[
|(Lw, Lu - Lv)_Y| \leq \|Lw\|_Y \|Lu - Lv\|_Y \\
\leq C_2^2 \|w\|_X \|u - v\|_X,
\]

using the coercivity relation (3.2). Hence we have:

\[
|\langle J'(u) - J'(v), w \rangle_X| \leq C_2^2 \|w\|_X \|u - v\|_X, \quad \forall u, v, w \in X.
\]

In particular, taking \( w = J'(u) - J'(v) \) yields:

\[
\|J'(u) - J'(v)\|_X \leq C_2^2 \|u - v\|_X.
\]

Therefore \( J' \) is Lipschitz continuous.

\( \square \)

**Remark 3.1.** Note that the above proof can be extended trivially to cover the least-squares functional with weakly imposed boundary conditions

\[
J(u) = \|Lu - f\|_Y^2 + \|Ru - g\|_Z^2.
\]

There are two additional conditions on the involved functional spaces that must also be satisfied in order for the proof of convergence given in [13] to hold. If we let \( \Sigma \) denote the set of all rank-one tensors then the following conditions must be satisfied:
1. Span(\(\Sigma\)) is a dense subset of \(X\) for \(\| \cdot \|_X\).

2. \(\Sigma\) is weakly closed in \((X, \| \cdot \|_X)\).

The ADN Theory supplies us with a functional space \(X\) that is simply a Sobolev space depending on the set of indices defining the principal part of the differential operator \(L\). As a result these two conditions will hold for a least-squares formulated problem. Indeed, a proof of this for the simple case of \(H^1\) spaces can be found in the paper by Cancès et al. [13] in the context of a high-dimensional Poisson equation.

The four conditions are therefore satisfied by a least-squares formulated problem. This means that the convergence of the greedy algorithm associated with any least-squares PGD algorithm is guaranteed since it is covered by the general proof provided by Cancès et al. [13]

3.2. Functional Eckart-Young Approach. A proof of convergence for least-squares PGD algorithms was also given by Falcó and Nouy [18] based on their generalised Eckart-Young theorem approach. Given that the variational problem derived from the Euler-Lagrange equation associated with the minimisation of the quadratic least-squares functional (2.8) can be written equivalently with:

\[
A(u, v) = \langle L^* Lu, v \rangle_X, \quad L(v) = \langle L^* f, v \rangle_X,
\]

then we can introduce the inner product \(\langle \cdot, \cdot \rangle_L\) induced by the operator \(L^*L\):

\[
\langle u, v \rangle_L = \langle L^* Lu, v \rangle_X = \langle Lu, Lv \rangle_Y,
\]

and associated norm:

\[
\| u \|_L = \sqrt{\langle u, u \rangle_L}
\]

Now since \(\| u \|_L^2 = \| Lu \|_Y^2\) then norm equivalence between \(\| \cdot \|_L\) and \(\| \cdot \|_X\) follows directly from the coercivity estimate (2.3). Hence under the assumption that \(\Sigma\) is weakly closed in \((X, \| \cdot \|_X)\) we have that it is also weakly closed in \((X, \| \cdot \|_L)\) since equivalent norms induce the same weak topology. For a given \(z \in X\) we then use the \(L\)-norm to define an associated rank-one projector \(\Pi_L(z)\) with which we can define the optimal progressive rank-J separated representation of the solution \(u = L^{-1} f\) by:

\[
u_J = \sum_{j=1}^{J} u^{(j)}, \quad u^{(j)} \in \Pi_L(u - u_{j-1}).\]

The generalised Eckart-Young Theorem in [18] then ensures that this sequence converges as \(J \rightarrow \infty\). The additional assumption that \(\text{Span}(\Sigma)\) is a dense subset of \(X\) for \(\| \cdot \|_X\) ensures that it converges to the solution \(u = L^{-1} f\).

Remark 3.2. As before this proof can be trivially extended to cover the case where we have weakly imposed boundary conditions so that we have:

\[
A(u, v) = \langle Lu, Lv \rangle_Y + \langle R u, R v \rangle_Z.
\]

Note that the two additional assumptions we made were exactly the assumptions on the involved spaces in the paper by Cancès et al. [13] that we mentioned above.

3.3. Rate of Convergence. While there are no theoretical results of convergence rates for PGD algorithms specific to least-squares formulations, it has been noted by Nouy [24] that PGD algorithms based on least-squares formulations converge slower than their Galerkin counterparts. It was also noted that the rate of
convergence could be improved by weighting the norms in the quadratic least-squares functional. To investigate this further we shall firstly consider least-squares formulations of the Poisson equation so we can compare rates of convergence with earlier results on the Galerkin formulation. We will not use the Stokes problem for this comparison since we cannot guarantee LBB stability in the Galerkin formulation.

4. Least-Squares Formulation of the Poisson Equation. We begin by noting that there is nothing to gain practically from solving a least-squares formulation of the Poisson equation. This is because it already possesses a natural Rayleigh-Ritz setting and hence there are no benefits to be gained in constructing an artificial one by residual minimisation. The reason we have chosen to do so is to be able to compare convergence rates of PGD algorithms based on least-squares and Galerkin formulations of equivalent problems. The aim is to provide a better understanding of how the rates of convergence differ in both formulations and, in particular, how they can be improved.

Consider the Poisson problem:

\[
\begin{align*}
-\nabla^2 \phi &= f & \text{in } \Omega, \\
\phi &= g & \text{on } \Gamma.
\end{align*}
\]

(4.1)

To begin applying least-squares formulations we must recast this as a first-order system. We shall consider two first-order systems: one which is homogeneous elliptic and one which is not.

4.1. Div-Grad System. The simplest way to recast (4.1) in the form of a first-order system is to introduce the vector \( \mathbf{u} = -\nabla \phi \) which leads to the following Div-Grad system equivalent to (4.1):

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= f & \text{in } \Omega, \\
\mathbf{u} + \nabla \phi &= 0 & \text{in } \Omega, \\
\phi &= g & \text{on } \Gamma.
\end{align*}
\]

(4.2, 4.3, 4.4)

The following estimate for this formulation can be derived from the ADN theory (e.g. see Bochev and Gunzburger [10]):

\[
\| \phi \|_{q+2} + \| \mathbf{u} \|_{q+1} \leq C_q (\| \nabla \cdot \mathbf{u} \|_q + \| \nabla \phi + \mathbf{u} \|_{q+1}),
\]

for some constant \( C_q > 0 \). This estimate holds for all \( q \geq 0 \) but, since this system is non-homogeneous elliptic, the required differentiability on the involved function spaces is impractical for all \( q \geq 0 \). However, Bochev and Gunzburger [10] proved that the estimate can be extended to all \( q \in \mathbb{R} \). This result enables us to choose \( q = -1 \) yielding the following estimate:

\[
\| \phi \|_1 + \| \mathbf{u} \|_0 \leq C_{-1} (\| \nabla \cdot \mathbf{u} \|_{-1} + \| \nabla \phi + \mathbf{u} \|_0).
\]

(4.5)

This has overcome the practical issue of differentiability but has introduced a new problem in the form of the negative norm.

4.2. Dealing with the Negative Norm. The difficulties in using negative norms (as well as trace norms) were stated in Section 2.1. As mentioned in that section we deal with such norms by using an approximation of their discrete norm generating operator. The norm generating operator for the \( H^{-1} \)-norm is given by
\( S_Y = (-\Delta)^{-1/2} \) where \((-\Delta)^{-1}\) denotes the inverse operator of the Poisson equation with homogeneous boundary conditions [10]. In other words we have that:

\[
(4.6) \quad \|\psi\|_{-1}^2 = \|(-\Delta)^{-1/2}\psi\|_0^2, \quad \forall \psi \in H^{-1}(\Omega).
\]

We consider the following two simple approximations of the discrete norm generating operator \( S_Y^h \) (see e.g. [8]):

- \( \text{The identity operator } S_Y^h = I \).
- \( \text{A mesh parameter-scaled identity operator } S_Y^h = hI. \)

The first of these approximations is equivalent to simply replacing the \( H^{-1}\)-norm by an \( L^2\)-norm yielding the following least-squares functional:

\[
J_1(\phi, u) = \|\nabla \cdot u - f\|_0^2 + \|\nabla \phi + u\|_0^2,
\]

whereas the second approximation yields the following weighted functional:

\[
J_2(\phi, u) = h^2\|\nabla \cdot u - f\|_0^2 + \|\nabla \phi + u\|_0^2.
\]

Both these approximations have the advantage that they are very simple to implement but they also both lead to discrete norms which are only quasi-equivalent. This means that it becomes unclear, as we refine our approximation space, how well the estimate (4.5) is preserved. For the first of these approximations this also means that we are unable to provide a proof of an optimal rate of convergence in \( h \). For the second approximation optimal rates of convergence can still be proven using carefully constructed duality arguments [10]. However, an undesirable consequence is that the condition number of the involved linear systems is increased [10]. An additional disadvantage of both these approximations is that it is unclear how the rank of the PGD approximation affects the discrete norm-equivalence.

There is a third way of approximating the discrete norm generating operator that was first considered by Bramble et al. [11]. This involves considering the inner-product generating operator \( S_Y^h = (-\Delta)^{-1} \) defined by:

\[
\langle \psi, \phi \rangle_{-1} = \langle (-\Delta)^{-1/2}\psi, (-\Delta)^{-1/2}\phi \rangle_0 = \langle (-\Delta)^{-1}\psi, \phi \rangle_0, \quad \forall \psi, \phi \in H^{-1}(\Omega).
\]

One then uses the discrete approximation \( (S_Y^h)^h = h^2I + K^h \) where \( K^h \) is a spectrally equivalent approximation of the Galerkin solution operator for \(-\Delta\) [10]. Note that, in the literature, it is often the case that this is stated as approximating the discrete norm generating operator by \( S_Y^h = hI + (K^h)^{1/2} \) (e.g. [8, 10]). This should be thought of as an abuse of notation since this approximation would actually introduce an additional unwanted term, \( 2h\langle (K^h)^{1/2}(\nabla \cdot u - f), \nabla \cdot u - f \rangle_0 \), into the least-squares functional. Unlike the other two methods, this approximation retains norm-equivalence in the discrete norms and hence optimal rates of convergence in \( h \) follow directly. This comes at the cost of being a more expensive approximation to implement. The difficulty lies in calculating a suitable, self-adjoint, operator \( K^h \). As mentioned earlier this operator must be a spectrally equivalent approximation of the Galerkin solution operator for \(-\Delta\). In other words, if we let \( G^h : H^{-1}(\Omega) \to H_0^{1,h}(\Omega) \), where \( G^h\psi = u^h \) if and only if

\[
\langle \nabla u^h, \nabla v^h \rangle_0 = \langle \psi, v^h \rangle_0, \quad \forall v^h \in H_0^{1,h}(\Omega).
\]

Then we need to find an operator \( K^h \) that is spectrally equivalent to \( G^h \) i.e. there exists some constant \( c \) for which [10]:

\[
c^{-1}(G^h\psi^h, v^h)_0 \leq (K^h v^h, v^h)_0 \leq c(G^h v^h, v^h)_0, \quad \forall v^h \in H_0^{1,h}(\Omega).
\]
This is a property that is satisfied by any good preconditioner of the Poisson problem. This can be expensive to construct in standard implementations of least-squares methods but unfortunately the problem is even more expensive for least-squares PGD algorithms. To see why this is the case, consider the 2D domain \( \Omega = \Omega_x \times \Omega_y \). We know from [13] that \( H^1(\Omega) \neq H^1(\Omega_x) \otimes H^1(\Omega_y) \) and hence we also have the same property for its dual, \( H^{-1}(\Omega) \neq H^{-1}(\Omega_x) \otimes H^{-1}(\Omega_y) \). This means that the Galerkin solution operator \( G^h \) cannot be expanded as a finite sum of tensorised operators, i.e. there exists no operators \( G^h_{x,j} : H^{-1}(\Omega_x) \to H^1_h(\Omega_x) \) and \( G^h_{y,j} : H^{-1}(\Omega_y) \to H^1_h(\Omega_y) \) such that \( G^h = \sum_{j=1}^J G^h_{x,j} \otimes G^h_{y,j} \). This important point was made by Cancès et al. [14] in the context of inverting Riesz operators for use in dual residual minimisation in the PGD. For our purpose it means that we cannot find suitable preconditioners \( K_{x,j}^h \) and \( K_{y,j}^h \), such that \( K^h = \sum_{j=1}^J K_{x,j}^h \otimes K_{y,j}^h \), for use in the alternating steps of our fixed point linearisation without incurring a great deal of expense. For this reason we believe this third way of treating the \( H^{-1} \)-norm to be impractical in the context of the PGD and hence we shall not consider it further.

Returning our attention to the Div-Grad system, we are able to derive the Euler-Lagrange equations associated with the minimisation of the two quadratic least-squares functionals, \( J_k(\phi, \mathbf{u}) \), \( k = 1, 2 \): Find \( \mathbf{v} = (\phi, \mathbf{u})^T \in H^1_0(\Omega) \times (L^2(\Omega))^2 \) such that:

\[
A_k(\mathbf{v}, \mathbf{v}^*) = L_k(\mathbf{v}^*), \quad \forall \mathbf{v}^* = (\phi^*, \mathbf{u}^*)^T \in H^1_0(\Omega) \times (L^2(\Omega))^2
\]

for \( k = 1, 2 \), where:

\[
A_1(\mathbf{v}, \mathbf{v}^*) = (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{u}^*)_0 + (\nabla \phi + \mathbf{u}, \nabla \phi^* + \mathbf{u}^*)_0,
\]

\[
A_2(\mathbf{v}, \mathbf{v}^*) = h^2(\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{u}^*)_0 + (\nabla \phi + \mathbf{u}, \nabla \phi^* + \mathbf{u}^*)_0,
\]

and

\[
L_1(\mathbf{v}) = \langle \mathbf{f}, \nabla \cdot \mathbf{u}^* \rangle_0, \quad L_2(\mathbf{v}) = h^2 \langle \mathbf{f}, \nabla \cdot \mathbf{u}^* \rangle_0,
\]

Note that, while the dependent variable \( \mathbf{u} \in (L^2(\Omega))^2 \) in the continuous least-squares estimate (4.5), we will still use a \( C^0 \)-continuous discrete space for \( \mathbf{u} \) in practice since we need to evaluate the divergence of this quantity which is undefined for elements of \( L^2 \).

### 4.3. Extended Div-Grad System.

The next first-order formulation of the Poisson equation that we shall consider is the extended Div-Grad system. This is essentially the same as the Div-Grad system (4.2)-(4.4) with the inclusion of an additional redundant equation and boundary condition [15]:

\[
\nabla \cdot \mathbf{u} = f \quad \text{in} \quad \Omega,
\]

\[
\mathbf{u} + \nabla \phi = 0 \quad \text{in} \quad \Omega,
\]

\[
\nabla \times \mathbf{u} = 0 \quad \text{in} \quad \Omega,
\]

\[
\phi = 0 \quad \text{on} \quad \Gamma,
\]

\[
\mathbf{n} \times \mathbf{u} = 0 \quad \text{on} \quad \Gamma,
\]

where \( \mathbf{n} \) denotes the outward unit normal to \( \Gamma \). The additional boundary condition holds since, from (4.8), we have that \( \mathbf{n} \times \mathbf{u} = -\mathbf{n} \times \nabla \phi = 0 \) since the boundary condition on \( \phi \) implies that its tangential derivatives vanish on the boundary. Note
that for simplicity of notation we have made the Dirichlet boundary condition on \( \phi \) homogeneous. For the non-homogeneous case, \( \phi = g \) on \( \Gamma \), the additional boundary condition (4.11) should be replaced by \( \mathbf{n} \times \mathbf{u} = -\mathbf{n} \times \nabla g \).

The additional equation (4.9) is derived by taking the curl of (4.8) and using the identity \( \nabla \times \nabla \phi = 0 \). Note that the curl operator is only defined in \( \mathbb{R}^2 \) and \( \mathbb{R}^3 \). For higher-dimensional Poisson equations one should instead consider the exterior derivation \( d_{k+1} \) in the differential de Rham complex [5], where \( d_k = \nabla \), so that \( \text{Ker}(d_{k+1}) = \text{Im}(\nabla) \) and hence \( d_{k+1} \mathbf{u} = -d_{k+1} \nabla \phi = 0 \).

The following estimate for this extended formulation can be derived from the ADN theory (e.g. see Bochev and Gunzburger [10]):

\[
\|\phi\|_1 + \|\mathbf{u}\|_1 \leq C_0(\|\nabla \cdot \mathbf{u}\|_0 + \|\mathbf{u} + \nabla \phi\|_0 + \|\nabla \times \mathbf{u}\|_0),
\]

for some constant \( C_0 > 0 \). Note that this system is homogeneous elliptic since the right hand side of the above estimate only involves \( L^2 \)-norms. This leads to the following quadratic least-squares functional:

\[
J_3(\phi, \mathbf{u}) = \|\nabla \cdot \mathbf{u} - f\|_0^2 + \|\mathbf{u} + \nabla \phi\|_0^2 + \|\nabla \times \mathbf{u}\|_0^2.
\]

We are then able to derive the Euler-Lagrange equations associated with the minimisation of (4.12): Find \( \mathbf{u} = (\phi, \mathbf{u})^T \in H^1_0(\Omega) \times H^1_\times(\Omega) \) such that

\[
A_3(\mathbf{u}, \mathbf{v}^*) = L_3(\mathbf{v}^*), \quad \forall \mathbf{v}^* = (\phi^*, \mathbf{u}^*)^T \in H^1_0(\Omega) \times H^1_\times(\Omega),
\]

where

\[
H^1_\times(\Omega) = \{ \mathbf{u} \in H^1(\Omega) \mid \mathbf{n} \times \mathbf{u} = 0 \quad \text{on} \quad \Gamma \},
\]

and

\[
A_3(\mathbf{u}, \mathbf{v}^*) = (\nabla \cdot \mathbf{u}, \mathbf{n} \cdot \mathbf{v}^*)_0 + (\mathbf{u} + \nabla \phi, \mathbf{u}^* + \nabla \phi^*)_0 + (\nabla \times \mathbf{u}, \nabla \times \mathbf{u}^*)_0,
\]

\[
L_3(\mathbf{v}^*) = (f, \nabla \cdot \mathbf{v}^*)_0.
\]

In summary, we have considered three different least-squares methods for the Poisson problem. Two of these least-squares methods were obtained from the non-homogeneous elliptic Div-Grad system which are associated with minimisation of the functionals \( J_1(\phi, \mathbf{u}) \) and \( J_2(\phi, \mathbf{u}) \), the difference between the two methods being the choice of weighting of the norms. The third least-squares method was based on the homogeneous elliptic extended Div-Grad system associated with the minimisation of the functional \( J_3(\phi, \mathbf{u}) \) given above. We shall now explain how these methods are implemented into the PGD framework and then compare the performance of the different methods, with each other as well as with a standard Galerkin PGD, using some examples.

4.4. Implementation into the PGD. Consider a first-order formulation of the 2D Poisson equation on the rectangular domain \( \Omega = [a, b] \times [c, d] \). In the PGD algorithm we seek a reduced basis separated representation of the dependent variables:

\[
u(x, y) \approx \sum_{j=1}^J X_j^u(x)Y_j^v(y) =: u_J(x, y), \quad \psi(x, y) \approx \sum_{j=1}^J X_j^\psi(x)Y_j^\psi(y) =: \psi_J(x, y),
\]

\[
\phi(x, y) \approx \sum_{j=1}^J X_j^\phi(x)Y_j^\phi(y) =: \phi_J(x, y).
\]
Although, least-squares methods are most commonly applied in conjunction with a finite element discretisation, we use spectral element methods which have been applied to least-squares formulations by Proot and Gerritsma [25–27]. To this end, we divide \([a, b]\) into \(K_x\) elements, \([a_{k_x-1}, a_{k_x}], k_x = 1, \ldots, K_x\), and divide \([c, d]\) into \(K_y\) elements, \([c_{k_y-1}, c_{k_y}], k_y = 1, \ldots, K_y\). The PGD basis functions \(X_j^{(\bullet)}(x)\), \(Y_j^{(\bullet)}(y)\) for \((\bullet) = \{\phi, u, v\}\) are then piecewise polynomials given by:

\[
X_j^{(\bullet)}(x) = \begin{cases} \sum_{i=0}^{N} \alpha_{j,i,k_x} h_i(k_x)(x), & \text{if } x \in [a_{k_x-1}, a_{k_x}], \\ 0, & \text{otherwise,} \end{cases}
\]

\[
Y_j^{(\bullet)}(y) = \begin{cases} \sum_{i=0}^{N} \beta_{j,i,k_y} h_i(k_y)(y), & \text{if } y \in [c_{k_y-1}, c_{k_y}], \\ 0, & \text{otherwise,} \end{cases}
\]

where \(h_{i,k}, i = 0, \ldots, N\), are the Lagrange interpolating polynomials on the \(k^{th}\) element.

The algorithm then proceeds in much the same way as for the Galerkin progressive PGD except that instead of imposing Galerkin orthogonality in order to calculate the next PGD mode we employ the Euler-Lagrange equation associated with the minimisation of our chosen quadratic least-squares functional. This still leads to a nonlinear system which is solved via an alternating directions fixed point algorithm.

### 4.5. Numerical Results.

**Example 4.1.** (Infinite Rank Solution): Consider the Poisson equation (4.1) on the domain \(\Omega = [-1, 1]^2\) with homogeneous Dirichlet boundary conditions and source term

\[
f(x, y) = 4\pi^2(x^2(1-y^2)^2 + y^2(1-x^2)^2) \sin(\pi(1-x^2)(1-y^2)) \\
+ 2\pi((1-x^2) + (1-y^2)) \cos(\pi(1-x^2)(1-y^2)).
\]

This problem has the exact solution \(\phi = \sin(\pi(1-x^2)(1-y^2))\) for the primary dependent variable \(\phi\). This solution does not have a finite rank separated representation and hence we expect monotonic convergence as we increase the rank of the approximation.

Figure 1 shows the convergence in the rank, \(J\), of the PGD approximation of \(\phi\) for the Galerkin, the two least-squares Div-Grad (LSQDG-1 & LSQDG-2), and the least-squares extended Div-Grad (LSQXDG) PGD algorithms. The discretisation used was a spectral element method with \(N = 8\) and \(K_x = K_y = 3\). This plot shows that even least-squares methods based on the non-homogeneous elliptic Div-Grad system converge at a rate that is competitive with the Galerkin PGD. However, the extended Div-Grad system does appear to be the best of the least-squares PGD algorithms with a rate of convergence very similar to the Galerkin PGD.

Figure 1(a) does not present a clear winner in terms of convergence in the primary dependent variable \(\phi\). However, we also note that \(\phi\) does not appear in the problematic \(H^{-1}\)-norm in the continuous least-squares estimate (4.5) for the Div-Grad system. For this reason, in Figure 1(b), we have plotted convergence of the approximation for \(u\). Note that the Galerkin PGD has been left out of this comparison since \(u\) does not appear in the second-order formulation (4.1).

Figure 1(b) provides confirmation that the least-squares PGD algorithms based on the non-homogeneous elliptic Div-Grad system converge slower than the least-squares
PGD algorithm based on the homogeneous elliptic extended Div-Grad system. It further indicates that the Div-Grad system which uses a mesh-parameter weighted $L^2$-norm (LSQDG-2) converges faster than the non weighted $L^2$-norm approach (LSQDG-1).

Finally, in Figure 2, we have compared CPU times for each of the four PGD algorithms as the rank of the approximation is increased. This clearly shows that LSQDG-1 is much more expensive than the other methods. The reason for this difference is that LSQDG-1 takes longer to converge in the alternating directions fixed point algorithm. Note that LSQDG-2 and LSQXDG are indistinguishable in terms of CPU time.

Example 4.2. (Rank-2 Solution): To further test that the algorithms are performing as expected, consider an example defined on the same domain as Example 4.1 with homogeneous boundary conditions and with source term

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y) + 2(2 - x^2 - y^2).$$
This problem has the exact solution \( \phi = \sin(\pi x) \sin(\pi y) + (1 - x^2)(1 - y^2) \) for the primary dependent variable \( \phi \). This solution has a rank-2 separated representation and hence we expect that the PGD algorithms are able to converge after two iterations.

\[ \phi = \sin(\pi x) \sin(\pi y) + (1 - x^2)(1 - y^2) \]

![Graph](image)

**Fig. 3:** Comparison of Least-Squares and Galerkin PGDs for Example 4.2

In Figure 3 we see that all the algorithms besides LSQDG-1 converge in two iterations as expected. This indicates that this algorithm does not sufficiently represent the continuous problem. In other words, since the Div-Grad system is non-homogeneous elliptic, the continuous least-squares estimate (4.5) has not been sufficiently preserved in the LSQDG-1 algorithm.

On the other hand, we find that LSQDG-2 is still able to converge in two iterations. This highlights the significance of using weighted \( L^2 \)-norms for non-homogeneous elliptic systems. However, as in the previous example, we find that the LSQXDG algorithm displays superior convergence behaviour. All the algorithms except LSQDG-1 run very quickly, hence we do not compare CPU time for this example.

**4.6. Conclusions.** In these examples the superior least-squares algorithm is consistently the one based on the homogeneous elliptic extended Div-Grad system. It always performs as expected in terms of capturing the natural rank of the exact solutions and displays superior levels of convergence, particularly for the additional dependent variable \( u \). Although it may not seem useful to have good convergence in the variable \( u \) it is often the case in applications that the additional dependent variables have important physical meaning. It is also the case that the primary variables can appear in the \( H^{-1} \)-norms in the continuous least-squares estimates e.g. the pressure in the Stokes problem which will be discussed in the next section. Therefore, it is important to have the best rate of convergence for all dependent variables. We noted in Example 4.1 that LSQXDG was also the fastest of the least-squares algorithms in terms of CPU time.

On the other hand, the algorithms based on the non-homogeneous elliptic Div-Grad system generally performed quite poorly. In Example 4.2 they failed to capture the natural rank of the exact solutions as well as exhibiting poor rates of convergence for \( u \) in the infinite rank case in Example 4.1. As far as CPU time is concerned, in Example 4.1, we noticed that LSQDG-1 was significantly slower than the other
algorithms. We believe the inferiority of these algorithms can be explained by the non-homogeneous ellipticity of the Div-Grad system. Indeed, we believe that the continuous least-squares estimate (4.5) is not sufficiently preserved in the context of the PGD by these two algorithms. We also noticed that using the weighted $L^2$-norm in LSQDG-2, in general, resulted in a significant improvement in performance over the unweighted case. However, it was not significant enough to improve performance beyond that of LSQXDG.

In conclusion, to construct efficient and reliable least-squares PGD algorithms, homogeneous ellipticity of the underlying system appears to be a key factor. This is not the case in standard implementations of least-squares methods where non-homogeneous elliptic systems are often preferred for their simplicity. However, we have found significant evidence that least-squares algorithms based on non-homogeneous elliptic systems perform poorly within the PGD framework.

The algorithm that performs most consistently in these examples is the Galerkin PGD algorithm. It yields the best rates of convergence which are comparable with those for the LSQXDG algorithm contrary to the observations of Nouy [24]. More significantly the Galerkin PGD algorithm was the most efficient algorithm since it involved the solution of much smaller linear systems. In the following section we consider a problem which does not possess a natural energy minimisation principle, namely the Stokes problem. In this case the proof of convergence for Galerkin PGDs no longer holds and hence convergence is no longer guaranteed. Furthermore, there are LBB stability issues when using a Galerkin PGD algorithm. For these reasons efficient least-squares PGD algorithms based on experience from the Poisson problem are considered.

5. Least-Squares Formulation of the Stokes Problem. We now turn our attention to the Stokes problem. There are problems when applying a PGD algorithm to the Galerkin formulation of this problem since we can no longer guarantee the required LBB stability when seeking solutions in the non-linear manifold, $\Sigma$, of rank-one tensors. By using a least-squares formulation instead of the Galerkin formulation we no longer solve a saddle-point problem and hence there is no longer any need to satisfy the LBB condition.

The Stokes problem in its classical form is given by:

\begin{align}
\nabla^2 \mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega, \\
\nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega, \\
\mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma.
\end{align}

There are several possible equivalent first-order systems for the Stokes problem and a wide selection have been documented in the thesis of Proot [25]. In this paper we consider two of these formulations. The first is the velocity-vorticity-pressure (VVP) system which is the most commonly used reformulation of the Stokes problem in the literature. This is because it requires comparatively fewer dependent variables and can give a direct and accurate approximation to the vorticity. Unfortunately, this formulation does not supply us with a homogeneous elliptic system when Dirichlet boundary conditions on the velocities are used. Therefore, we also consider the extended velocity gradient-velocity-pressure (Extended VGVP) system reformulation which has a larger number of dependent variables but does supply us with a homogeneous elliptic system when Dirichlet boundary conditions on the velocities are imposed.
5.1. VVP System. To derive the velocity-vorticity-pressure formulation of the Stokes problem we first define the vorticity in 2D by \( \omega = \nabla \times u \). Then using the identity \( \nabla^\perp (\nabla \times u) = -\nabla^2 u + \nabla (\nabla \cdot u) \) together with incompressibility \( \nabla \cdot u = 0 \) we can write \(-\nabla^2 u = \nabla^\perp (\nabla \times u) = \nabla^\perp \omega \). Hence the VVP system is given by:

\begin{align}
(5.4) & \quad \nabla^\perp \omega + \nabla p = f \quad \text{in } \Omega, \\
(5.5) & \quad \omega - \nabla \times u = 0 \quad \text{in } \Omega, \\
(5.6) & \quad \nabla \cdot u = 0 \quad \text{in } \Omega, \\
(5.7) & \quad u = g \quad \text{on } \Gamma.
\end{align}

The following estimate for this formulation can be derived from the ADN theory (e.g. see Bochev and Gunzburger [10]):

\begin{align}
(5.8) & \quad \|u\|_q + \|\omega\|_{q+1} + \|p\|_{q+1} \leq C_q \left( \|\nabla^\perp \omega + \nabla p\|_q + \|\omega - \nabla \times u\|_{q+1} + \|\nabla \cdot u\|_{q+1} \right),
\end{align}

for some constant \( C_q > 0 \). In the same way as for the Div-Grad system, this can be extended to all \( q \in \mathbb{R} \) (see Bochev and Gunzburger [9]). Hence we can choose \( q = -1 \) to overcome practical implementation issues related to the required differentiability of the involved function spaces. This yields the following coercivity estimate:

\begin{align}
(5.9) & \quad \|u\|_1 + \|\omega\|_0 + \|p\|_0 \leq C_{-1} \left( \|\nabla^\perp \omega + \nabla p\|_{-1} + \|\omega - \nabla \times u\|_0 + \|\nabla \cdot u\|_0 \right).
\end{align}

Note that these coercivity estimates (5.8)-(5.9) rely on the assumption that there exists a unique solution. Since the pressure can only be evaluated up to a constant, we need to include an additional constraint in the quadratic least-squares functionals to ensure uniqueness. For the Stokes problem we use the zero mean pressure constraint \( \ell(p) = \int_{\Omega} p \, d\Omega = 0 \). The \( H^{-1} \)-norm in (5.9) is treated in the same way as for the Div-Grad system yielding the following two quadratic least-squares functionals:

\begin{align}
(5.10) & \quad J_1(u, \omega, p) = \|\nabla^\perp \omega + \nabla p - f\|_0^2 + \|\omega - \nabla \times u\|_0^2 + \|\nabla \cdot u\|_0^2 + \mu \|\ell(p)\|^2, \\
(5.11) & \quad J_2(u, \omega, p) = h^2 \|\nabla^\perp \omega + \nabla p - f\|_0^2 + \|\omega - \nabla \times u\|_0^2 + \|\nabla \cdot u\|_0^2 + \mu \|\ell(p)\|^2,
\end{align}

where \( \mu > 0 \) is an adjustable constant. We are then able to derive the Euler-Lagrange equations associated with the minimisation of the functionals (5.10)-(5.11): Find \( \mathbf{v} = (u, \omega, p) \in H^1_0(\Omega) \times L^2(\Omega) \times L^2(\Omega) \) such that:

\[
A_k(\mathbf{v}, \mathbf{v}^*) = L_k(\mathbf{v}^*), \quad \forall \mathbf{v}^* = (u^*, \omega^*, p^*) \in (H^1_0(\Omega))^2 \times L^2(\Omega) \times L^2(\Omega),
\]

for \( k = 1, 2 \), where

\[
A_1(\mathbf{v}, \mathbf{v}^*) = \langle \nabla^\perp \omega + \nabla p, \nabla^\perp \omega^* + \nabla p^* \rangle_0 + \langle \omega - \nabla \times u, \omega^* - \nabla \times u^* \rangle_0 + \langle \nabla \cdot u, \nabla \cdot u^* \rangle_0 + \mu \|\ell(p)\| \|\ell(p^*)\|,
\]

\[
A_2(\mathbf{v}, \mathbf{v}^*) = h^2 \langle \nabla^\perp \omega + \nabla p, \nabla^\perp \omega^* + \nabla p^* \rangle_0 + \langle \omega - \nabla \times u, \omega^* - \nabla \times u^* \rangle_0 + \langle \nabla \cdot u, \nabla \cdot u^* \rangle_0 + \mu \|\ell(p)\| \|\ell(p^*)\|,
\]

and

\[
L_1(\mathbf{v}^*) = \langle f, \nabla^\perp \omega^* + \nabla p^* \rangle_0, \quad L_2(\mathbf{v}^*) = h^2 \langle f, \nabla^\perp \omega^* + \nabla p^* \rangle_0.
\]
5.2. Extended VGVP System. We now consider the velocity gradient-velocity pressure formulation of the Stokes problem. Define the velocity gradient by:

\[ \nabla = (\nabla u)^T = \left( \begin{array}{ccc} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{array} \right) = \left( \begin{array}{cc} V_1 & V_2 \\ V_3 & V_4 \end{array} \right). \]

If we then define the divergence of a tensor to be the divergence of its rows then we obtain the identity \( \nabla \cdot \nabla = \nabla^2 u \). Hence we can rewrite the Stokes problem (5.1)-(5.3) as the following first-order VGVP system:

\begin{align*}
(5.12) & \quad - \nabla \cdot \nabla + \nabla p = f \quad \text{in } \Omega, \\
(5.13) & \quad \nabla \cdot u = 0 \quad \text{in } \Omega, \\
(5.14) & \quad \nabla - (\nabla u)^T = 0 \quad \text{in } \Omega, \\
(5.15) & \quad u = g \quad \text{on } \Gamma.
\end{align*}

Unfortunately, it has been shown by Cai et al. [12] that this does not lead to a homogeneous elliptic system. However, in the same manner as we did for the extended Div-Grad formulation of the Poisson equation we can include additional redundant equations to provide us with a problem which is homogeneous elliptic. Indeed, this leads to the following extended VGVP system:

\begin{align*}
(5.16) & \quad - \nabla \cdot \nabla + \nabla p = f \quad \text{in } \Omega, \\
(5.17) & \quad \nabla \cdot u = 0 \quad \text{in } \Omega, \\
(5.18) & \quad \nabla - (\nabla u)^T = 0 \quad \text{in } \Omega, \\
(5.19) & \quad \nabla (\text{Tr} \nabla) = 0 \quad \text{in } \Omega, \\
(5.20) & \quad \nabla \times \nabla = 0 \quad \text{in } \Omega, \\
(5.21) & \quad u = 0 \quad \text{on } \Gamma, \\
(5.22) & \quad n \times \nabla = 0 \quad \text{on } \Gamma.
\end{align*}

The additional boundary condition holds since from (5.18) we have that \( n \times \nabla = n \times (\nabla u)^T = 0 \) since the boundary condition on \( u \) implies that its tangential derivatives vanish on the boundary. Note that for simplicity we have considered homogeneous velocity Dirichlet boundary conditions (5.21). For the non-homogeneous case, \( u = g \) on \( \Gamma \), the additional boundary condition (5.22) should be replaced by \( n \times \nabla = n \times (\nabla g)^T \).

The first redundant equation, (5.19), is satisfied since \( \text{Tr} \nabla = V_1 + V_4 = \nabla \cdot u = 0 \). The second redundant equation, (5.20), is satisfied since if we define the curl of a tensor to be the curl of its rows then we have

\[ \nabla \times \nabla = \left( \begin{array}{cc} \frac{\partial^2 u}{\partial x \partial y} - \frac{\partial^2 u}{\partial y \partial x} \\ \frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 v}{\partial y \partial x} \end{array} \right) = 0. \]

This system has been proven to be homogeneous elliptic in an ad hoc manner by Cai et al. [12] but can also be verified using ADN theory (see Croft [17]). This yields the following estimate:

\begin{align*}
\|u\|_1 + \|p\|_1 + \|\nabla u\|_1 & \leq C(\| - \nabla \cdot \nabla + \nabla p\|_0 + \|\nabla \cdot u\|_0 + \|\nabla - (\nabla u)^T\|_0 \\
& \quad + \|\nabla (\text{Tr} \nabla)\|_0 + \|\nabla \times \nabla\|_0),
\end{align*}

(5.23)
for some constant $C > 0$. This leads to the following quadratic least-squares functional:

$$J_3(u, p, V) = \| -\nabla \cdot V + \nabla p - f \|_0^2 + \| \nabla \cdot u \|_0^2 + \| V - (\nabla u)^T \|_0^2$$

$$+ \| \nabla (\text{Tr} V) \|_0^2 + \| \nabla \times V \|_0^2 + \mu \ell(p)^2. \tag{5.24}$$

We are then able to derive the Euler-Lagrange equations associated with the minimisation of (5.24): Find $v = (u, p, V) \in H_0^1(\Omega) \times H^1(\Omega) \times H_\text{\text{div}}^1(\Omega)$ such that:

$$A_3(v, v^*) = L_3(v^*), \quad \forall v^* = (u^*, p^*, V^*) \in H_0^1(\Omega) \times H^1(\Omega) \times H_\text{\text{div}}^1(\Omega),$$

where:

$$H_\text{\text{div}}^1(\Omega) = \{ V \in H^1(\Omega) \mid n \times V = 0 \text{ on } \Gamma \},$$

and where:

$$A_3(v, v^*) = \langle -\nabla \cdot V + \nabla p, -\nabla \cdot V^* + \nabla p^* \rangle_0 + \langle V - (\nabla u)^T, V^* - (\nabla u^*)^T \rangle_0$$

$$+ \langle \nabla \cdot u, \nabla \cdot u^* \rangle_0 + \langle \nabla (\text{Tr} V), \nabla (\text{Tr} V^*) \rangle_0 + \langle \nabla \times V, \nabla \times V^* \rangle_0$$

$$+ \mu \ell(p) \ell(p^*),$$

and

$$L_3(v^*) = \langle f, -\nabla \cdot V^* + \nabla p^* \rangle_0,$$

where we have included the zero mean pressure constraint $\ell(p) = \int_{\Omega} p \, d\Omega = 0$ to ensure uniqueness of the solution and where, as before, $\mu > 0$ is an adjustable constant.

5.3. Numerical Results. Example 5.1. (Infinite Rank Pressure Solution): Consider the Stokes problem (5.1)-(5.3) on the domain $\Omega = [-1, 1]^2$ with homogeneous Dirichlet boundary conditions on the velocity components and with source term

$$f(x, y) = \left( \frac{\pi y \cos(\pi xy) + 4\pi^2 \sin(2\pi y)(2 \cos(2\pi x) - 1)}{\pi x \cos(\pi xy) - 4\pi^2 \sin(2\pi x)(2 \cos(2\pi y) - 1)} \right),$$

This Stokes problem has the following exact solution:

$$u = \left( -\sin(2\pi y)(\cos(2\pi x) - 1) \right), \quad p = \sin(\pi xy).$$

The velocity, $u$, possesses a natural rank-1 separated representation and so we might expect the PGD algorithms to converge in a single iteration for the velocity. The pressure, on the other hand, does not have a finite rank separated representation and so we expect this to simply converge monotonically as we increase the rank of the approximation.

Figure 4 shows the convergence in the rank, $J$, for both the non-homogeneous elliptic VVP least-squares PGD algorithms. We used a spectral element discretisation with $N = 8$ and $K_x = K_y = 3$. Here VVP-1 and VVP-2 denote the methods based on the least squares functionals $J_1(u, \omega, p)$ and $J_2(u, \omega, p)$ defined by (5.10) and (5.11) respectively. We observe disappointing rates of convergence in both cases, in particular for the vorticity and pressure since these are the dependent variables which appear
in the $H^{-1}$-norm in the continuous least-squares estimate (5.9). We also note that there is no significant difference in the rates of convergence for these two methods.

Figure 5 shows convergence in the rank for the velocity and pressure in all three least-squares PGD algorithms for the Stokes problem. We have only compared these dependent variables since they are the only ones shared by both the VVP and VGVP systems. It is clear from this that the algorithm based on the homogeneous elliptic extended VGVP formulation (XVGVP) displays the best rate of convergence.

Unfortunately, none of the algorithms captured the natural rank-1 separated form of the true solution to the velocity. We believe the reason for this is due to the coupling in the Stokes problem. That is to say since the pressure is coupled with the velocity and the pressure does not possess a finite rank separated representation then we cannot expect these algorithms to converge in a single iteration while there is still room to improve the pressure solution by increasing the rank. This is not an issue we experienced with the Poisson equation since the dependent variables, $\phi$ and $u$, were
both either finite rank or both infinite rank.

In Figure 6, we have plotted the CPU times for each of the three algorithms. The XVGVP algorithm is only slightly slower than the VVP algorithms and this is despite the fact that the XVGVP algorithm involves linear systems almost twice the size of those in the VVP algorithms. Certainly this slight increase in computational cost is insignificant when we consider how much faster the XVGVP algorithm converges in the rank.

Finally, in Figure 7, we have plotted the CPU times for the same problem without the zero mean pressure constraint imposed implicitly. Due to the iterative nature of the PGD we are still able to obtain a solution for the pressure and we can then simply modify the solution to have zero mean afterwards by adding a suitable constant. The reason for showing this plot is that the imposition of the zero mean pressure yields a linear system for the pressure which involves a full matrix. It is then reasonable to assume that this may result in a computationally slower algorithm. However, as we
can see from Figure 7, this is not the case. The VVP algorithms in particular are significantly slower whereas the speed of the XVGVP algorithm is relatively unaltered. The large runtime increase of the VVP algorithms is due to the algorithm getting stuck in the alternating directions linearisation. This can be seen by the large jumps in Figure 7. In fact the VVP algorithms require a very coarse convergence criterion in the linearisation in order to make them run at all. This not only highlights the importance of imposing the zero mean pressure condition implicitly but also highlights issues related to the non-homogeneous elliptic VVP formulations.

**Example 5.2. (Rank-1 Pressure Solution):**

To test our reasoning for the algorithms not capturing the rank-1 nature of the velocity solution we now consider following example on the same domain as Example 5.1 with homogeneous boundary conditions and source term

\[
f(x, y) = \begin{pmatrix}
\pi \cos(\pi x) \sin(\pi y) + 4\pi^2 \sin(2\pi y)(2\cos(2\pi x) - 1)

\pi \sin(\pi x) \sin(\pi y) - 4\pi^2 \sin(2\pi x)(2\cos(2\pi y) - 1)
\end{pmatrix}.
\]

This has the following exact solution:

\[
\mathbf{u} = \begin{pmatrix}
-\sin(2\pi y)(\cos(2\pi x) - 1) \\
\sin(2\pi x)(\cos(2\pi y) - 1)
\end{pmatrix},
\]

\[p = \sin(\pi x) \sin(\pi y).
\]

In Figure 8 we have plotted the convergence in the rank of the velocity and pressure for all three least-squares Stokes algorithms. We used the same discretisation as in the previous example. From this plot we can see that the only algorithm which was able to capture the rank-1 nature of the velocity and pressure is the homogeneous elliptic XVGVP algorithm. Indeed, we find once again that the non-homogeneous elliptic VVP algorithms display very poor rates of convergence particularly for the pressure. We also note that after the second iteration of the PGD the XVGVP algorithm obtains a solution to the pressure which is significantly worse than the previous iteration. We are not sure what the reason for this might be but in a
practical situation it would not be an issue since the global convergence criterion would be satisfied after the first iteration of the PGD and this increase in error of the pressure would not be experienced.

Figure 9 displays the CPU times for the three least-squares algorithms for this example. From this we can see that the XVGVP algorithm now displays a runtime which is much more competitive with the VVP algorithms than in the previous example. Combined with the extremely superior rate of convergence of the XVGVP algorithm it is clear that in this case it is by far the superior method.

Unlike the previous example, we will not show the CPU times when the zero mean pressure condition is not included implicitly. The reason for this being that without the implicit zero mean pressure we find that the algorithms fail to converge in the linearisation after a certain small number of PGD iterations. This in itself again indicates the importance of the inclusion of the zero mean pressure in this way.

5.4. Conclusions. The difference in performance between PGD algorithms based on homogeneous elliptic and non-homogeneous elliptic systems was found to be even more significant in the Stokes problem than for the Poisson problem. Indeed, the homogeneous elliptic XVGVP system yielded superior rates of convergence in both velocity and pressure as well as being able to capture the rank-1 nature of the solution in Example 5.2. We also found that, despite the much larger linear systems, the computational efficiency of the XVGVP algorithm measured in terms of CPU time was comparable with the VVP algorithms. This is a significant piece of evidence that supports the view that homogeneous elliptic systems are crucial to constructing efficient least-squares PGD algorithms. This strengthens the conclusions we made previously.

We also noted the significance of including the zero mean pressure constraint implicitly. Without it the VVP algorithms converged much slower and, in the case of Example 5.2, the algorithms even failed to converge in the linearisation step. This highlights the importance of having an underlying coercivity estimate since a requirement of Theorem 2.5 for such a coercivity estimate to exist was the existence of a unique solution. In the case of the Stokes problem this meant we needed the pressure to have a unique solution which we could enforce by including the zero mean
pressure constraint in the least-squares functional. Furthermore, the CPU times for the VVP algorithms were considerably larger than the XVGVP algorithm which provides further evidence that non-homogeneous elliptic algorithms are inferior to the homogeneous elliptic XVGVP algorithm.

**6. Conclusions and Future Work.** In this paper we have demonstrated that a crucial component to constructing efficient least-squares PGD algorithms is homogeneous ellipticity of the underlying system. In other words to construct an efficient least-squares PGD algorithm the discrete, low-rank, least-squares estimate is required to sufficiently represent the underlying continuous least squares estimate. When the system is homogeneous elliptic this is certainly the case since the continuous least-squares estimate involves only $L^2$-norms of the differential operators. This is extremely simple to implement discretely while still retaining the continuous estimate. However, homogeneous ellipticity is a term only associated with the estimates derived from the ADN theory. The ADN theory is a very powerful tool since it reduces verification of continuous estimates to the verification of some algebraic conditions but the drawback is that it is only applicable to a particular class of problems, namely linear elliptic PDEs with standard boundary conditions. However, it is possible to construct least-squares methods for a wider class of problems including non-standard boundary conditions, non-linear problems and even non-elliptic problems. We refer the interested reader to Bochev and Gunzburger [10] for a thorough review on this topic. It would certainly be of future interest to investigate how well least-squares PGDs perform for these sorts of problem.

It would also be of considerable interest to consider least-squares methods for non-symmetric problems, such as the convection-diffusion equation, which was considered in the context of the minimal residual PGD by Cancès et al. [14]. If one were able to construct efficient and stable convection dominated least-squares PGDs this could provide a potential starting point for applying least-squares PGDs to kinetic theory models in polymer rheology. In particular, to the Fokker-Planck equation which is essentially a high-dimensional, time dependent convection-diffusion equation which has already been the focus for a number of applications of the Galerkin PGD [3, 4, 19, 22, 23]. An advantage of least-squares PGDs is the ability to prove convergence without having to use a semi-implicit scheme as in Figueroa and Süli [19].

Additional areas of future work from a theoretical perspective include derivation of error estimates specific to the convergence rate of least-squares PGDs and exploration of how the continuous estimates in non-homogeneous elliptic systems are affected by the rank of the separated representation in the PGD.

**Acknowledgement.** The first author (TLDC) received funding from the Engineering and Physical Sciences Research Council of the United Kingdom to support his doctoral studies and this is gratefully acknowledged.

**REFERENCES**


