Bayesian Analysis (2019) **14**, Number 3, pp. 937–1012

A Bayesian Conjugate Gradient Method (with Discussion)

Jon Cockayne[*], Chris J. Oates[†], Ilse C.F. Ipsen[‡], and Mark Girolami[§]

Abstract. A fundamental task in numerical computation is the solution of large linear systems. The conjugate gradient method is an iterative method which offers rapid convergence to the solution, particularly when an effective preconditioner is employed. However, for more challenging systems a substantial error can be present even after many iterations have been performed. The estimates obtained in this case are of little value unless further information can be provided about, for example, the magnitude of the error. In this paper we propose a novel statistical model for this error, set in a Bayesian framework. Our approach is a strict generalisation of the conjugate gradient method, which is recovered as the posterior mean for a particular choice of prior. The estimates obtained are analysed with Krylov subspace methods and a contraction result for the posterior is presented. The method is then analysed in a simulation study as well as being applied to a challenging problem in medical imaging.

Keywords: probabilistic numerics, linear systems, Krylov subspaces.

MSC 2010 subject classifications: 62C10, 62F15, 65F10.

1 Introduction

This paper presents an iterative method for solution of systems of linear equations of the form

$$A\boldsymbol{x}^* = \boldsymbol{b}, \tag{1}$$

where $A \in \mathbb{R}^{d\times d}$ is an invertible matrix and $\boldsymbol{b} \in \mathbb{R}^d$ is a vector, each given, while $\boldsymbol{x}^* \in \mathbb{R}^d$ is to be determined. The principal novelty of our method, in contrast to existing approaches, is that its output is a *probability distribution* over vectors $\boldsymbol{x} \in \mathbb{R}^d$ which reflects knowledge about $\boldsymbol{x}^*$ after expending a limited amount of computational effort. This allows the output of the method to be used, in a principled *anytime* manner, tailored to reflect a constrained computational budget. In a special case, the mode of this distribution coincides with the estimate provided by the standard conjugate gradient method, whilst the probability mass is proven to contract onto $\boldsymbol{x}^*$ as more iterations are performed.

[*]Department of Statistics, University of Warwick, Coventry, CV4 7AL, j.cockayne@warwick.ac.uk, url: http://www.joncockayne.com

[†]School of Mathematics and Statistics, Herschel Building, Newcastle University, NE1 7RU, chris.oates@ncl.ac.uk

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC, 27695-8205, m.girolami@imperial.ac.uk

[§]Department of Mathematics, Huxley Building, Imperial College London, London, SW7 2AZ, ipsen@ncsu.edu

 https://doi.org/10.1214/19-BA1145

Challenging linear systems arise in a wide variety of applications. Of these, partial differential equations (PDEs) should be emphasised, as these arise frequently throughout the applied sciences and in engineering (Evans, 2010). Finite element and finite difference discretisations of PDEs each yield large, sparse linear systems which can sometimes be highly ill-conditioned, such as in the classically ill-posed backwards heat equation (Evans, 2010). Even for linear PDEs, a detailed discretisation may be required. This can result in a linear system with billions of degrees of freedom and require specialised algorithms to be even approximately solved practically (e.g. Reinarz et al., 2018). Another example arises in computation with Gaussian measures (Bogachev, 1998; Rasmussen, 2004), in which analytic covariance functions, such as the exponentiated quadratic, give rise to challenging linear systems. This has an impact in a number of related fields, such as symmetric collocation solution of PDEs (Fasshauer, 1999; Cockayne et al., 2016), numerical integration (Larkin, 1972; Briol et al., 2018) and generation of spatial random fields (Besag and Green, 1993; Parker and Fox, 2012; Schäfer et al., 2017). In the latter case, large linear systems must often be solved to sample from these fields, such as in models of tropical ocean surface winds (Wikle et al., 2001) where systems may again be billion-dimensional. Thus, it is clear that there exist many important situations in which error in the solution of a linear system cannot practically be avoided.

1.1 Linear Solvers

The solution of linear systems is one of the most ubiquitous problems in numerical analysis and Krylov subspace methods (Hestenes and Stiefel, 1952; Liesen and Strakos, 2012) are among the most successful at obtaining an approximate solution at low cost. Krylov subspace methods belong to the class of *iterative methods* (Saad, 2003), which construct a sequence $(\boldsymbol{x}_m)$ that approaches $\boldsymbol{x}^*$ and can be computed in an efficient manner. Iterative methods provide an alternative to *direct methods* (Davis, 2006; Allaire and Kaber, 2008) such as the LU or Cholesky decomposition, which generally incur higher cost as termination of the algorithm after $m < d$ iterations is not meaningful. In certain cases an iterative method can produce an accurate approximation to $\boldsymbol{x}^*$ with reduced computational effort and memory usage compared to a direct method.

The conjugate gradient (CG) method (Hestenes and Stiefel, 1952) is a popular iterative method, and perhaps the first instance of a Krylov subspace method. The error arising from CG can be shown to decay exponentially in the number of iterations, but convergence is slowed when the system is poorly conditioned. As a result, there is interest in solving equivalent *preconditioned* systems (Allaire and Kaber, 2008), either by solving $P^{-1}A\boldsymbol{x}^* = P^{-1}\boldsymbol{b}$ (left-preconditioning) or $AP^{-1}P\boldsymbol{x}^* = \boldsymbol{b}$ (right-preconditioning), where P is chosen both so that $P^{-1}A$ (or AP^{-1}) has a lower condition number than A itself, and so that computing the solution of systems $P\boldsymbol{y} = \boldsymbol{c}$ is computationally inexpensive for arbitrary $\boldsymbol{y}$ and $\boldsymbol{c}$. Effective preconditioning can dramatically improve convergence of CG, and of Krylov subspace methods in general, and is recommended even for well-conditioned systems owing to how rapidly conjugacy is lost in CG when implemented numerically. One reasonably generic method for sparse systems involves approximate factorisation of the matrix, through an incomplete LU or incomplete Cholesky decomposition (e.g. Ajiz and Jennings, 1984; Saad, 1994). Other common approaches

exploit the structure of the problem. For example, in numerical solution of PDEs a coarse discretisation of the system can be used to construct a preconditioner for a finer discretisation (e.g. Bramble et al., 1990). A more detailed survey of preconditioning methods can be found in many standard texts, such as Benzi (2002) and Saad (2003). However, no approach is universal, and in general careful analysis of the structure of the problem is required to determine an effective preconditioner (Saad, 2003, p. 283). At worst, constructing a good preconditioner can be as difficult as solving the linear system itself.

In situations where numerical error cannot practically be made negligible, an estimate for the error $\boldsymbol{x}_m - \boldsymbol{x}^*$ must accompany the output $\boldsymbol{x}_m$ of any linear solver. The standard approach is to analytically bound $\|\boldsymbol{x}_m - \boldsymbol{x}^*\|$ by some function of the residual $\|A\boldsymbol{x}_m - \boldsymbol{b}\|$, for appropriate choices of norms, then to monitor the decay of the relative residual. In implementations, the algorithm is usually terminated when this reaches machine precision, which can require a very large number of iterations and substantial computational effort. This often constitutes the principal bottleneck in contemporary applications. The contribution of this paper is to demonstrate how Bayesian analysis can be used to develop a richer, probabilistic description for the error in estimating the solution $\boldsymbol{x}^*$ with an iterative method. From a user's perspective, this means that solutions from the presented method can still be used in a principled way, even when only a small number of iterations can be afforded.

1.2 Probabilistic Numerical Methods

The concept of a probabilistic numerical method dates back to Larkin (1972). The principal idea is that problems in numerical analysis can be cast as inference problems and are therefore amenable to statistical treatment. *Bayesian* probabilistic numerical methods (Cockayne et al., 2017) posit a prior distribution for the unknown, in our case $\boldsymbol{x}^*$, and condition on a finite amount of information about $\boldsymbol{x}^*$ to obtain a posterior that reflects the level of uncertainty in $\boldsymbol{x}^*$, given the finite information obtained. In contemporary applications, it is common for several numerical methods to be composed in a *pipeline* to perform a complex task. For example, climate models (such as Roeckner et al., 2003) involve large systems of coupled differential equations. To simulate from these models, many approximations must be combined. Bayesian probabilistic numerical methods are of particular interest in this setting, as a probabilistic description of error can be coherently propagated through the pipeline to describe the structure of the overall error and study the contribution of each component of the pipeline to that error (Hennig et al., 2015). As many numerical methods rely on linear solvers, understanding the error incurred by these numerical methods is critical. Other works to recently highlight the value of statistical thinking in this application area includes Calvetti et al. (2018).

In recent work, Hennig (2015) treated the problem of solving (1) as an inference problem for the matrix A^{-1}, and established correspondence with existing iterative methods by selection of different matrix-valued Gaussian priors within a Bayesian framework. This approach was explored further in Bartels and Hennig (2016). There, it was observed that the posterior distribution over the matrix in Hennig (2015) produces the

same factors as in the LU or Cholesky decompositions.[1] Our contribution takes a fundamentally different approach, in that a prior is placed on the solution $\boldsymbol{x}^*$ rather than on the matrix A^{-1}. There are advantages to the approach of Hennig (2015), in that solution of multiple systems involving the same matrix is trivial. However we argue that it is more intuitive to place a prior on $\boldsymbol{x}^*$ than on A^{-1}, as one might more easily reason about the solution to a system than the elements of the inverse matrix. Furthermore, the approach of placing a prior on $\boldsymbol{x}^*$ is unchanged by any left-preconditioning of the system, while the prior of Hennig (2015) is not preconditioner-invariant.

Contribution The main contributions of this paper are as follows:

- The *Bayesian conjugate gradient* (BayesCG) method is proposed for solution of linear systems. This is a novel probabilistic numerical method in which both prior and posterior are defined on the solution space for the linear system, $\mathbb{R}^d$. We argue that placing a prior on the solution space is more intuitive than existing probabilistic numerical methods and corresponds more directly with classical iterative methods. This makes substitution of BayesCG for existing iterative solvers simpler for practitioners.
- The specification of the prior distribution is discussed in detail. Several natural prior covariance structures are introduced, motivated by preconditioners or Krylov subspace methods. In addition, a hierarchical prior is proposed in which all parameters can be marginalised, allowing automatic adjustment of the posterior to the scale of the problem. This discussion provides some generic prior choices to make application of BayesCG more straightforward for users unfamiliar with probabilistic numerical methods.
- It is shown that, for a particular choice of prior, the posterior mode of BayesCG coincides with the output of the standard CG method. An explicit algorithm is provided whose complexity is shown to be a small constant factor larger than that of the standard CG method. Thus, BayesCG can be efficiently implemented and could be used in place of classical iterative methods with marginal increase in computational cost.
- A thorough convergence analysis for the new method is presented, with computational performance in mind. It is shown that the posterior mean lies in a particular Krylov subspace, and rates of convergence for the mean and contraction for the posterior are presented. The distributional quantification of uncertainty provided by this method is shown to be conservative in general.

The structure of the paper is as follows: In Section 2 BayesCG is presented and its inputs discussed. Its correspondence with CG is also established for a particular choice of prior. Section 3 demonstrates that the mean from BayesCG lies in a particular Krylov subspace and presents a convergence analysis of the method. In Section 4 the critical

[1] Recall that the Cholesky decomposition is a symmetric version of the LU decomposition for symmetric positive-definite matrices.

issue of prior choice is addressed. Several choices of prior covariance are discussed and a hierarchical prior is introduced to allow BayesCG to adapt to the scale of the problem. Section 5 contains implementation details, while in Section 6 the method is applied to a challenging problem in medical imaging which requires repeated solution of a linear system arising from the discretisation of a PDE. The paper concludes with a discussion in Section 7. Proofs of all theoretical results are provided in the electronic supplement (Cockayne et al., 2019).

2 Methods

We begin in Section 2.1 by defining a Bayesian probabilistic numerical method for the linear system in (1). In Section 2.2 a correspondence to the CG method is established. In Section 2.3 we discuss a particular choice of search directions that define BayesCG. Throughout this paper, note that A is not required to be symmetric positive-definite, except for in Section 2.2.

2.1 Probabilistic Linear Solver

In this section we present a general probabilistic numerical method for solving (1). The approach taken is Bayesian, so that the method is defined by the choice of prior and the information on which the prior is to be conditioned. For this work, the information about $\boldsymbol{x}^*$ is linear and is provided by *search directions* $\boldsymbol{s}_i$, $i = 1, \ldots, m \ll d$, through the matrix-vector products

$$y_i := (\boldsymbol{s}_i^\top A)\boldsymbol{x}^* = \boldsymbol{s}_i^\top \boldsymbol{b}. \tag{2}$$

The matrix-vector products on the right-hand-side are assumed to be computed without error,[2] which implies a likelihood model in the form of a Dirac distribution:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \delta(\boldsymbol{y} - S_m^\top A\boldsymbol{x}), \tag{3}$$

where S_m denotes the matrix whose columns are $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m$. This section assumes the search directions are given *a-priori*. The specific search directions which define BayesCG will be introduced in Section 2.3.

In general the recovery of $\boldsymbol{x}^*$ from $m < d$ pieces of information is ill-posed. The prior distribution serves to regularise the problem, in the spirit of Tikhonov (1963); Stuart (2010). Linear information is well-adapted to inference with stable distributions[3] such as the Gaussian or Cauchy distributions, in that the posterior distribution is available in closed-form. Optimal estimation with linear information is also well-understood (cf. Traub et al., 1988). To proceed, let $\boldsymbol{x}$ be a random variable, which will be used to model epistemic uncertainty regarding the true solution $\boldsymbol{x}^*$, and endow $\boldsymbol{x}$ with the prior distribution

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_0, \Sigma_0), \tag{4}$$

[2] i.e. in exact arithmetic.

[3] Let X_1 and X_2 be independent copies of a random variable X. Then X is said to be *stable* if, for any constants $\alpha, \beta > 0$, the random variable $\alpha X_1 + \beta X_2$ has the same distribution as $\gamma X + \delta$ for some constants $\gamma > 0$ and δ.

where $\boldsymbol{x}_0$ and Σ_0 are each assumed to be known *a-priori*, an assumption that will be relaxed in Section 4. It will be assumed throughout that Σ_0 is a symmetric and positive-definite matrix.

Having specified the prior and the information, there exists a unique Bayesian probabilistic numerical method which outputs the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y}_m)$ (Cockayne et al., 2017) where $\boldsymbol{y}_m = [y_1, \ldots, y_m]^\top$ satisfies $\boldsymbol{y}_m = S_m^\top A\boldsymbol{x}^* = S_m^\top \boldsymbol{b}$. This is made clear in the following result:

Proposition 1 (Probabilistic Linear Solver)**.** *Let* $\Lambda_m = S_m^\top A\Sigma_0 A^\top S_m$ *and* $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$*. Then the posterior distribution is given by*

$$p(\boldsymbol{x}|\boldsymbol{y}_m) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_m, \Sigma_m),$$

where

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0 \tag{5}$$

$$\Sigma_m = \Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A\Sigma_0. \tag{6}$$

This provides a distribution on $\mathbb{R}^d$ that reflects the state of knowledge given the information contained in $\boldsymbol{y}_m$. The mean, $\boldsymbol{x}_m$, could be viewed as an approximation to $\boldsymbol{x}^*$ that might be provided by a numerical method. From a computational perspective, the presence of the $m \times m$ matrix Λ_m^{-1} could be problematic as this implies a second linear system must be solved, albeit at a lower cost $\mathcal{O}(m^3)$. This could be addressed to some extent by updating Λ_m^{-1} iteratively using the Woodbury matrix inversion lemma, though this would not reduce the overall cost. However, as the search directions can be chosen arbitrarily, this motivates a choice which *diagonalises* Λ_m, to make the inverse trivial. This will be discussed further in Section 2.3.

Note that the posterior distribution is singular, in that $\det(\Sigma_m) = 0$. This is natural since what uncertainty remains in directions not yet explored is simply the restriction of the prior, in the measure-theoretic sense, to the subspace orthogonal to the columns of $S_m^\top A$. As a result, the posterior distribution is concentrated on a linear subspace of $\mathbb{R}^d$. Singularity of the posterior makes computing certain quantities difficult, such as posterior probabilities. Nevertheless, Σ_m can be decomposed using techniques such as the singular-value decomposition, so sampling from the posterior is straightforward.

For a positive-definite matrix M, define the matrix-induced inner-product of two vectors in $\mathbb{R}^d$ by $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_M = \boldsymbol{x}^\top M \boldsymbol{x}'$, with associated norm $\|\cdot\|_M$. The following basic result establishes that the posterior covariance provides a connection to the error of $\boldsymbol{x}_m$ when used as a point estimator:

Proposition 2.

$$\frac{\|\boldsymbol{x}_m - \boldsymbol{x}^*\|_{\Sigma_0^{-1}}}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_{\Sigma_0^{-1}}} \leq \sqrt{\mathrm{tr}(\Sigma_m \Sigma_0^{-1})}.$$

Thus the right hand side provides an upper bound on the relative error of the estimator $\boldsymbol{x}_m$ in the Σ_0^{-1}-norm. This is a weak result and tighter results for specific

search directions are provided later. In addition to bounding the error $\boldsymbol{x}_m - \boldsymbol{x}^*$ in terms of the posterior covariance Σ_m, we can also compute the rate of contraction of the posterior covariance itself:

Proposition 3.

$$\mathrm{tr}(\Sigma_m \Sigma_0^{-1}) = d - m.$$

The combination of Propositions 2 and 3 implies that the posterior mean $\boldsymbol{x}_m$ is consistent and, since the posterior covariance characterises the width of the posterior, Proposition 3 can be viewed as a posterior contraction result. This result is intuitive; after exploring m linearly independent search directions, $\boldsymbol{x}^*$ has been perfectly identified in an m-dimensional linear subspace of $\mathbb{R}^d$. Thus, after adjusting for the weighting of $\mathbb{R}^d$ provided by the prior covariance Σ_0, it is natural that an appropriate measure of the size of the posterior should also converge at a rate that is linear.

2.2 Correspondence with the Conjugate Gradient Method

In this section we examine the correspondence of the posterior mean $\boldsymbol{x}_m$ described in Proposition 1 with the CG method. It is frequently the case that Bayesian probabilistic numerical methods have some classical numerical method as their mean, due to the characterisation of the conditional mean of a probability distribution as the L_2-best element of the underlying space consistent with the information provided (Diaconis, 1988; Cockayne et al., 2017).

The Conjugate Gradient Method A large class of iterative methods for solving linear systems defined by positive-definite matrices A can be motivated by sequentially solving the following minimisation problem:

$$\boldsymbol{x}_m = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{K}_m} \|\boldsymbol{x} - \boldsymbol{x}^*\|_A,$$

where $\mathcal{K}_m$ is a sequence of m-dimensional linear subspaces of $\mathbb{R}^d$. It is straightforward to show that this is equivalent to:

$$\boldsymbol{x}_m = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{K}_m} f(\boldsymbol{x}),$$

where $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A \boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{b}$ is a convex quadratic functional. Let $S_m \in \mathbb{R}^{d \times m}$ denote a matrix whose columns are arbitrary linearly independent search directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m$, with $\mathrm{range}(S_m) = \mathcal{K}_m$. Let $\boldsymbol{x}_0$ denote an arbitrary starting point for the algorithm. Then $\boldsymbol{x}_m = \boldsymbol{x}_0 + S_m \boldsymbol{c}$ for some $\boldsymbol{c} \in \mathbb{R}^m$ which can be computed by solving $\nabla f(\boldsymbol{x}_0 + S_m \boldsymbol{c}) = 0$. This yields:

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + S_m (S_m^\top A S_m)^{-1} S_m^\top (\boldsymbol{b} - A\boldsymbol{x}_0). \tag{7}$$

In CG (Hestenes and Stiefel, 1952) the search directions are constructed to simplify the inversion in (7) by imposing that the search directions are A-conjugate, that is,

$\langle \boldsymbol{s}_i^{\mathrm{CG}}, \boldsymbol{s}_j^{\mathrm{CG}} \rangle_A = 0$ whenever $i \neq j$. A set $\{\boldsymbol{s}_i\}$ of A-conjugate vectors is also said to be A-*orthogonal*, while if the vectors additionally have $\|\boldsymbol{s}_i\|_A = 1$ for each i they are said to be A-*orthonormal*. For simplicity of notation, we will usually work with A-orthonormal search directions, but in most implementations of CG the normalisation step can introduce stability issues and is therefore avoided.

Supposing that such a set of A-orthonormal search directions can be found, (7) simplifies to

$$\boldsymbol{x}_m^{\mathrm{CG}} = \boldsymbol{x}_0^{\mathrm{CG}} + S_m^{\mathrm{CG}} (S_m^{\mathrm{CG}})^\top (\boldsymbol{b} - A\boldsymbol{x}_0^{\mathrm{CG}}) \tag{8}$$

which lends itself to an iterative numerical method:

$$\boldsymbol{x}_m^{\mathrm{CG}} = \boldsymbol{x}_{m-1}^{\mathrm{CG}} + \boldsymbol{s}_m^{\mathrm{CG}} (\boldsymbol{s}_m^{\mathrm{CG}})^\top (\boldsymbol{b} - A\boldsymbol{x}_{m-1}^{\mathrm{CG}}).$$

Search directions are also constructed iteratively, motivated by gradient descent on the function $f(\boldsymbol{x})$, whose negative gradient is given by $-\nabla f(\boldsymbol{x}) = \boldsymbol{b} - A\boldsymbol{x}$. The initial un-normalised search direction $\tilde{\boldsymbol{s}}_1^{\mathrm{CG}}$ is chosen to be $\tilde{\boldsymbol{s}}_1^{\mathrm{CG}} = \boldsymbol{r}_0^{\mathrm{CG}} = \boldsymbol{b} - A\boldsymbol{x}_0^{\mathrm{CG}}$, so that $\boldsymbol{s}_1^{\mathrm{CG}} = \tilde{\boldsymbol{s}}_1^{\mathrm{CG}} / \|\tilde{\boldsymbol{s}}_1^{\mathrm{CG}}\|_A$. Letting $\boldsymbol{r}_m^{\mathrm{CG}} = \boldsymbol{b} - A\boldsymbol{x}_m^{\mathrm{CG}}$, subsequent search directions are given by

$$\tilde{\boldsymbol{s}}_m^{\mathrm{CG}} := \boldsymbol{r}_{m-1}^{\mathrm{CG}} - \langle \boldsymbol{s}_{m-1}^{\mathrm{CG}}, \boldsymbol{r}_{m-1}^{\mathrm{CG}} \rangle_A \boldsymbol{s}_{m-1}^{\mathrm{CG}} \tag{9}$$

with $\boldsymbol{s}_m^{\mathrm{CG}} = \tilde{\boldsymbol{s}}_m^{\mathrm{CG}} / \|\tilde{\boldsymbol{s}}_m^{\mathrm{CG}}\|_A$. This construction leads to search directions $\boldsymbol{s}_1^{\mathrm{CG}}, \ldots, \boldsymbol{s}_m^{\mathrm{CG}}$ which form an A-orthonormal set.

Equation (8) makes clear the following proposition, which shows that for a particular choice of prior the CG method is recovered as the posterior mean from Proposition 1:

Proposition 4. *Assume A is symmetric and positive-definite. Let $\boldsymbol{x}_0 = \boldsymbol{0}$ and $\Sigma_0 = A^{-1}$. Then, taking $S_m = S_m^{CG}$, (5) reduces to $\boldsymbol{x}_m = \boldsymbol{x}_m^{CG}$.*

This result provides an intriguing perspective on the CG method, in that it represents the estimate produced by a rational Bayesian agent whose prior belief about $\boldsymbol{x}^*$ is modelled by $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, A^{-1})$. Dependence of the prior on the inaccessible matrix inverse is in accordance with the findings in Hennig (2015, Theorem 2.4 and Lemma 3.4), in which an analogous result was presented. As observed in that paper, the appearance of A^{-1} in the prior covariance is not practically useful, as while the matrix inverse cancels in the expression for $\boldsymbol{x}_m$, it remains in the expression for Σ_m.

2.3 Search Directions

In this section the choice of search directions for the method in Proposition 1 will be discussed, initially by following an information-based complexity argument (Traub et al., 1988). For efficiency purposes, a further consideration is that Λ_m should be easy to invert. This naturally suggests that search directions should be chosen to be conjugate with respect to the matrix $A\Sigma_0 A^\top$, rather than A. Note that this approach *does not* require A to be positive-definite, as $A\Sigma_0 A^\top$ is positive-definite for any non-singular A. Two choices of search direction will be discussed:

Optimal Information One choice is to formulate selection of S_m in a decision-theoretic framework, to obtain *optimal information* in the nomenclature of Cockayne et al. (2017). Abstractly, denote the probabilistic numerical method discussed above by $P[\cdot; \mu, S_m] : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$, where $\mathcal{P}(\mathbb{R}^d)$ is the set of all distributions on $\mathbb{R}^d$. The function $P[\boldsymbol{b}; \mu, S_m]$ takes a right-hand-side $\boldsymbol{b} \in \mathbb{R}^d$, together with a prior $\mu \in \mathcal{P}(\mathbb{R}^d)$ and a set of search directions S_m and outputs the posterior distribution from Proposition 1. Thus $P[\boldsymbol{b}; \mu, S_m]$ is a measure and $P[\boldsymbol{b}; \mu, S_m](\mathrm{d}\boldsymbol{x})$ denotes its infinitesimal element.

For general $\mu \in \mathcal{P}(\mathbb{R}^d)$, define the *average risk* associated with the search directions S_m to be

$$R(S_m, \mu) = \iint L(\boldsymbol{x}, \boldsymbol{x}^*) P[A\boldsymbol{x}^*; \mu, S_m](\mathrm{d}\boldsymbol{x}) \mu(\mathrm{d}\boldsymbol{x}^*), \tag{10}$$

where $L(\boldsymbol{x}, \boldsymbol{x}^*)$ represents a loss incurred when $\boldsymbol{x}$ is used to estimate $\boldsymbol{x}^*$. This can be thought of as a measure of the performance of the probabilistic numerical method, averaged both over the class of problems described by μ and over the output of the method. *Optimal information* in this paper concerns selection of S_m to minimise $R(S_m, \mu)$. The following proposition characterises optimal information for the posterior in Proposition 1 in the case of a squared-error loss function and when $\boldsymbol{x}_0 = \boldsymbol{0}$. Let $A^{-\top} = (A^{-1})^\top$, and let $M^{\frac{1}{2}}$ denote a square-root of a symmetric positive-definite matrix M with the property that $M^{\frac{\top}{2}} M^{\frac{1}{2}} = M$, where $M^{\frac{\top}{2}} = (M^{\frac{1}{2}})^\top$.

Proposition 5. *Suppose $\mu = \mathcal{N}(\boldsymbol{0}, \Sigma_0)$ and consider the squared-error loss $L(\boldsymbol{x}, \boldsymbol{x}^*) = \|\boldsymbol{x} - \boldsymbol{x}^*\|_M^2$ where M is an arbitary symmetric positive-definite matrix. Optimal information for this loss is given by*

$$S_m = A^{-\top} M^{\frac{\top}{2}} \Phi_m,$$

where Φ_m is the matrix whose columns are the m leading eigenvectors of $M^{\frac{1}{2}} \Sigma_0 M^{\frac{\top}{2}}$, normalised such that $\Phi_m^\top \Phi_m = I$.

The dependence of the optimal information on $A^{-\top}$ is problematic except for when $M = A^\top A$, which corresponds to measuring the performance of the algorithm through the residual $\|A\boldsymbol{x}_m - \boldsymbol{b}\|_2^2$. While this removes dependence on the inverse matrix, finding the search directions in this case requires computing the eigenvectors of $A\Sigma_0 A^\top$, the complexity of which would dominate the cost of computing the posterior in Proposition 1.

Conjugacy A second, more practical method for obtaining search directions that diagonalise Λ_m is similar to that taken in CG. Search directions are constructed which are conjugate to the matrix $A\Sigma_0 A^\top$ by following a similar procedure to that described in Section 2.2.

Proposition 6 (Conjugate Search Directions $\implies$ Iterative Method)**.** *Assume that the search directions are $A\Sigma_0 A^\top$-orthonormal. Denote $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$. Then, $\boldsymbol{x}_m$ in (5) simplifies to*

$$\boldsymbol{x}_m = \boldsymbol{x}_{m-1} + \Sigma_0 A^\top \boldsymbol{s}_m (\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1})$$

while to compute Σ_m in (6) it suffices to store only the vectors $\Sigma_0 A^\top \boldsymbol{s}_j$, for $j = 1, \ldots, m$.

On the surface, the form of this posterior differs slightly from that in Proposition 1, in that the data are given by $\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}$ rather than $\boldsymbol{s}_m^\top \boldsymbol{r}_0$. However, when search directions are conjugate, the two expressions are equivalent:

$$\begin{aligned}\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1} &= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top A \boldsymbol{x}_{m-1} \\ &= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top A \boldsymbol{x}_0 - \underbrace{\boldsymbol{s}_m^\top A \Sigma_0 A^\top S_{m-1}^\top}_{=0} \boldsymbol{r}_0 = \boldsymbol{s}_m^\top \boldsymbol{r}_0. \end{aligned} \tag{11}$$

Use of $\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}$ reduces the amount of storage required compared to direct application of (5). It also helps with stability as, while search directions can be shown to be conjugate mathematically, the accumulation of numerical error from floating point precision is such that numerical conjugacy may not hold, a point discussed further in Section S4.1 of the supplement.

An approach to constructing conjugate search directions for our probabilistic linear solver is now presented, again motivated by gradient descent.

Proposition 7 (Bayesian Conjugate Gradient Method)**.** *Recall the definition of the residual* $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$*. Denote* $\tilde{\boldsymbol{s}}_1 = \boldsymbol{r}_0$ *and* $\boldsymbol{s}_1 = \tilde{\boldsymbol{s}}_1 / \|\tilde{\boldsymbol{s}}_1\|_{A\Sigma_0 A^\top}$*. For* $m > 1$ *let*

$$\tilde{\boldsymbol{s}}_m = \boldsymbol{r}_{m-1} - \langle \boldsymbol{s}_{m-1}, \boldsymbol{r}_{m-1} \rangle_{A\Sigma_0 A^\top} \boldsymbol{s}_{m-1}.$$

Further, assume $\tilde{\boldsymbol{s}}_m \neq \boldsymbol{0}$ *and let* $\boldsymbol{s}_m = \tilde{\boldsymbol{s}}_m / \|\tilde{\boldsymbol{s}}_m\|_{A\Sigma_0 A^\top}$*. Then for each* m*, the set* $\{\boldsymbol{s}_i\}_{i=1}^m$ *is* $A\Sigma_0 A^\top$*-orthonormal, and as a result* $\Lambda_m = I$*.*

This is termed a Bayesian *conjugate gradient* method for the same reason as in CG, as search directions are chosen to be the direction of gradient descent subject to a conjugacy requirement, albeit a different one than in standard CG. In the context of Proposition 4, note that the search directions obtained coincide with those obtained from CG when A is symmetric positive-definite and $\Sigma_0 = A^{-1}$. Thus, BayesCG is a strict generalisation of CG. Note, however, that these search directions are constructed in a data-driven manner, in that they depend on the right-hand-side $\boldsymbol{b}$. This introduces a dependency on $\boldsymbol{x}^*$ through the relationship in (1) which is not taken into account in the conditioning procedure and leads to conservative uncertainty assessment, as will be demonstrated in Section 6.1.

3 BayesCG as a Krylov Subspace Method

In this section a thorough theoretical analysis of the posterior will be presented. Fundamental to the analysis in this section is the concept of a *Krylov subspace*.

Definition 8 (Krylov Subspace)**.** *The* Krylov subspace $K_m(M, \boldsymbol{v})$*,* $M \in \mathbb{R}^{d\times d}$*,* $\boldsymbol{v} \in \mathbb{R}^d$ *is defined as*

$$K_m(M, \boldsymbol{v}) := \operatorname{span}(\boldsymbol{v}, M\boldsymbol{v}, M^2\boldsymbol{v}, \ldots, M^m\boldsymbol{v}).$$

For a vector $\boldsymbol{w} \in \mathbb{R}^d$*, the* shifted Krylov subspace *is defined as*

$$\boldsymbol{w} + K_m(M, \boldsymbol{v}) := \operatorname{span}(\boldsymbol{w} + \boldsymbol{v}, \boldsymbol{w} + M\boldsymbol{v}, \boldsymbol{w} + M^2\boldsymbol{v}, \ldots, \boldsymbol{w} + M^m\boldsymbol{v}).$$

It is well-known that CG is a Krylov subspace method for symmetric positive-definite matrices A (Liesen and Strakos, 2012), meaning that

$$\boldsymbol{x}_m^{\mathrm{CG}} = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+K_{m-1}(A,\boldsymbol{r}_0)}{\arg\min} \|\boldsymbol{x}-\boldsymbol{x}^*\|_A.$$

It will now be shown that the posterior mean for BayesCG, presented in Proposition 6, is a Krylov subspace method. For convenience, let $K_m^* := \boldsymbol{x}_0 + K_m(\Sigma_0 A^\top A, \Sigma_0 A^\top \boldsymbol{r}_0)$.

Proposition 9. *The BayesCG mean $\boldsymbol{x}_m$ satisfies*

$$\boldsymbol{x}_m = \underset{\boldsymbol{x}\in K_{m-1}^*}{\arg\min} \|\boldsymbol{x}-\boldsymbol{x}^*\|_{\Sigma_0^{-1}}.$$

This proposition gives an alternate perspective on the observation that, when A is symmetric positive-definite and $\Sigma_0 = A^{-1}$, the posterior mean from BayesCG coincides with $\boldsymbol{x}_m^{\mathrm{CG}}$: Indeed, for this choice of Σ_0, K_m^* coincides with $\boldsymbol{x}_0 + K_m(A, \boldsymbol{r}_0)$ and furthermore, since under this choice of Σ_0 the norm minimised in Proposition 9 is $\|\cdot\|_A$, it is natural that the estimates $\boldsymbol{x}_m$ and $\boldsymbol{x}_m^{\mathrm{CG}}$ should be identical.

Proposition 9 allows us to establish a convergence rate for the BayesCG mean which is similar to that which can be demonstrated for CG. Let $\kappa(M) = \|M\|_2\|M^{-1}\|_2$ denote the condition number of a matrix M in the matrix 2-norm. Now, noting that $\kappa(\Sigma_0 A^\top A)$ his well-defined, as Σ_0 and A are each nonsingular, we have:

Proposition 10.

$$\frac{\|\boldsymbol{x}_m-\boldsymbol{x}^*\|_{\Sigma_0^{-1}}}{\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_{\Sigma_0^{-1}}} \leq 2\left(\frac{\sqrt{\kappa(\Sigma_0 A^\top A)}-1}{\sqrt{\kappa(\Sigma_0 A^\top A)}+1}\right)^m.$$

This rate is similar to the well-known convergence rate which for CG, in which $\kappa(\Sigma_0 A^\top A)$ is replaced by $\kappa(A)$. However, since it holds that $\kappa(A^\top A) \geq \kappa(A)$, the convergence rate for BayesCG will often be worse than that for CG, unless Σ_0 is chosen judiciously to reduce the condition number of $\kappa(\Sigma_0 A^\top A)$. Thus it appears that there is a price to be paid when uncertainty quantification is needed. This is unsurprising, as it is generally the case that uncertainty quantification is associated with additional cost over methods for which uncertainty quantification is not provided.

Nevertheless, the rate of convergence in Proposition 10 is significantly faster than the rate obtained in Proposition 2. The reason for this is that knowledge about how the search directions S_m were chosen has been exploited. The directions used in BayesCG are motivated by gradient descent on $f(\boldsymbol{s})$. Thus, if gradient descent is an effective heuristic for the problem at hand, then the magnitude of the error $\boldsymbol{x}_m - \boldsymbol{x}^*$ will decrease at a rate which is sub-linear. The same cannot be said for $\mathrm{tr}(\Sigma_m \Sigma_0^{-1})$ which continues to converge linearly as proven in Proposition 3. Thus, the posterior covariance will in general be conservative when the BayesCG search directions are used. This is verified empirically in Section 6.1.

4 Prior Choice

The critical issue of prior choice is now examined. In Section 4.1 selection of the prior covariance structure will be discussed. Then in Section 4.2 a hierarchical prior will be introduced to address the scale of the prior.

4.1 Covariance Structure

When A is symmetric positive-definite, one choice which has already been discussed is to set $\Sigma_0 = A^{-1}$, which results in a posterior mean equal to the output of CG. However correspondance of the posterior mean with CG does not in itself justify this modelling choice from a probabilistic perspective and moreover this choice is not practical, as access to A^{-1} would give immediate access to the solution of (1). We therefore discuss some alternatives for the choice of Σ_0.

Natural Prior Taking inspiration from probabilistic numerical methods for PDEs (Cockayne et al., 2016; Owhadi, 2015), a natural choice presents itself: The object through which information about $\boldsymbol{x}^*$ is extracted is $\boldsymbol{b}$, so it is natural, and mathematically equivalent, to place a relatively uninformative prior on the elements of $\boldsymbol{b}$ rather than on $\boldsymbol{x}^*$ itself. If $\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, I)$ then the implied prior model for $\boldsymbol{x}^*$ is $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, (A^\top A)^{-1})$. This prior is as impractical as that which aligns the posterior mean with CG, but has the attractive property that convergence is instantaneous when the search directions from Proposition 7 are used, as shown in Section S3.1 of the supplement.

Preconditioner Prior For systems in which a preconditioner is available, the preconditioner can be thought of as providing an approximation to the linear operator A. Inspired by the impractical natural covariance $(A^\top A)^{-1}$, one approach proposed in this paper is to set $\Sigma_0 = (P^\top P)^{-1}$, when a preconditioner P can be found. Since by design the action of P^{-1} can be computed efficiently, so too can the action of Σ_0. As mentioned in Section 1.1, the availability of a good preconditioner is problem-dependent.

Krylov Subspace Prior The analysis presented in Section 3 suggests another potential prior, in which probability mass is distributed according to an appropriate Krylov subspace $K_n(M, \boldsymbol{b})$. Consider a distribution constructed as the linear combination

$$\boldsymbol{x}_K = \sum_{i=0}^{n} w_i M^i \boldsymbol{b}, \tag{12}$$

where $\boldsymbol{w} := (w_0, \ldots, w_n) \sim \mathcal{N}(\boldsymbol{0}, \Phi)$ for some positive-definite matrix Φ. The distribution on $\boldsymbol{x}_K$ induced by (12) is clearly Gaussian with mean $\boldsymbol{0}$. To determine its covariance, note that the above expression can be rewritten as $\boldsymbol{x}_K = K_n \boldsymbol{w}$, where $K_n \in \mathbb{R}^{d \times (n+1)}$ is the matrix whose columns form a basis of the Krylov subspace $K_n(M, \boldsymbol{b})$, as would be given by the Lanczos or Arnoldi algorithms (Golub and Van Loan, 2013, Chapter 9).

Irrespective of choice of K_n, the covariance of $\boldsymbol{x}_K$ is given by $\mathbb{E}(\boldsymbol{x}_K \boldsymbol{x}_K^\top) = K_n \Phi K_n^\top$ so that $\boldsymbol{x}_K \sim \mathcal{N}(\boldsymbol{0}, K_n \Phi K_n^\top)$. One issue with this approach is that the computation of

the matrix K_n is of the same computational complexity as n iterations of BayesCG, requiring n matrix-vector products. To ensure that this cost does not dominate the procedure, it is necessary to take $n < m \ll d$. However, in this situation $\boldsymbol{x}^* \notin K_n(\boldsymbol{b}, M)$, so it is necessary to add additional probability mass on the space orthogonal to $K_n(M, \boldsymbol{b})$, to ensure that $\boldsymbol{x}^*$ lies in the prior support. To this end, let $K_n^{\perp}(\boldsymbol{b}, M) = \mathbb{R}^d \setminus K_n(\boldsymbol{b}, M)$, and let $K_n^{\perp}$ denote a matrix whose columns span $K_n^{\perp}(\boldsymbol{b}, M)$. Let $\boldsymbol{x}_K^{\perp} = K_n^{\perp}\boldsymbol{w}^{\perp}$, where $\boldsymbol{w}^{\perp} \sim \mathcal{N}(\boldsymbol{0}, \varphi I)$ for a scaling parameter $\varphi \in \mathbb{R}$. Then, the proposed Krylov subspace prior is given by

$$\boldsymbol{x} \; (= \boldsymbol{x}_0 + \boldsymbol{x}_K + \boldsymbol{x}_K^{\perp}) \; \sim \mathcal{N}\left(\boldsymbol{x}_0, K_n \Phi K_n^{\top} + \varphi K_n^{\perp}(K_n^{\perp})^{\top}\right).$$

The selection of the parameters of this prior, and issues related to its implementation, are discussed in Section S3.2 of the supplement.

4.2 Covariance Scale

For the distributional output of BayesCG to be useful it must be well-calibrated. Loosely speaking, this means that the true solution $\boldsymbol{x}^*$ should typically lie in a region where most of the posterior probability mass is situated. As such, the scale of the posterior variance should have the ability to adapt and reflect the difficulty of the linear system at hand. This can be challenging, partially because the magnitude of the solution vector is *a-priori* unknown and partially because of the aforementioned fact that the dependence of S_m on $\boldsymbol{x}^*$ is not accounted for in BayesCG.

In this section we propose to treat the prior scale as an additional parameter to be learned; that is we consider the prior model $p(\boldsymbol{x}|\nu) = \mathcal{N}(\boldsymbol{x}_0, \nu\Sigma_0)$, where $\boldsymbol{x}_0, \Sigma_0$ are as before, while $\nu \in \mathbb{R}^+$. This can be viewed as a generalised version of the prior in (4), which is recovered when $\nu = 1$. In this section we consider learning ν in a hierarchical Bayesian framework, but we note that ν could also be heuristically calibrated. An example of such a heuristic procedure is outlined in Section S4.3 of the supplement.

The approach pursued below follows a standard approach in Bayesian linear regression (Gelman et al., 2014). More generally, one could treat the entire covariance as unknown and perform similar conjugate analysis with an inverse-Wishart prior, though this extension was not explored. Consider then endowing ν with Jeffreys' (improper) reference prior $p(\nu) \propto \nu^{-1}$. The conjugacy of this prior with the Gaussian distribution is such that the posterior marginal distributions $p(\nu|\boldsymbol{y}_m)$ and $p(\boldsymbol{x}|\boldsymbol{y}_m)$ can be found analytically. For the following proposition, IG denotes an inverse-gamma distribution, while MVT_m denotes a multivariate t distribution with m degrees of freedom.

Proposition 11 (Hierarchical BayesCG)**.** *When $p(\boldsymbol{x}|\nu)$ and $p(\nu)$ are as above, the posterior marginal for ν is given by*

$$p(\nu|\boldsymbol{y}_m) = \mathsf{IG}\left(\frac{m}{2}, \frac{1}{2}\boldsymbol{r}_0^{\top} S_m \Lambda_m^{-1} S_m^{\top} \boldsymbol{r}_0\right)$$

while the posterior marginal for $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\boldsymbol{y}_m) = \mathsf{MVT}_m\left(\boldsymbol{x}_m, \frac{\boldsymbol{r}_0^{\top} S_m \Lambda_m^{-1} S_m^{\top} \boldsymbol{r}_0}{m} \Sigma_m\right).$$

When the search directions are $A\Sigma_0 A^\top$-orthonormal, this simplifies to

$$p(\nu|\boldsymbol{y}_m) = \mathsf{IG}\left(\frac{m}{2}, \frac{m}{2}\nu_m\right)$$
$$p(\boldsymbol{x}|\boldsymbol{y}_m) = \mathsf{MVT}_m\left(\boldsymbol{x}_m, \nu_m\Sigma_m\right),$$

where $\nu_m := \|S_m^\top \boldsymbol{r}_0\|_2^2/m$.

Since $\boldsymbol{r}_0$ reflects the initial error $\boldsymbol{x}_0 - \boldsymbol{x}^*$, the quantity ν_m can be thought of as describing the difficulty of the problem. Thus in this approach the scale of the posterior is data-dependent.

5 Implementation

In this section some important details of the implementation of BayesCG are discussed.

Computational Cost The cost of BayesCG is a constant factor higher than the cost of CG as three, rather than one, matrix-vector multiplications are required. Thus, the overall cost is $\mathcal{O}(md^2)$ when the search directions from Proposition 7 are used. Note that this cost assumes that A and Σ_0 are dense matrices; in the case of sparse matrices the cost of the matrix-vector multiplications is driven by the number of nonzero entries of each matrix rather than the dimension d.

Termination Criteria An appealing use of the posterior distribution might be to derive a probabilistic termination criterion for BayesCG. Recall from Proposition 2 that $\boldsymbol{x}_m$ approaches $\boldsymbol{x}^*$ at a rate bounded by $\sigma_m := \sqrt{\mathrm{tr}(\Sigma_m\Sigma_0^{-1})}$, and from Proposition 3 that $\mathrm{tr}(\Sigma_m\Sigma_0^{-1}) = d - m$. To decide in practice how many iterations of BayesCG should be performed we propose a termination criterion based upon the posterior distribution from Proposition 11:

$$\sigma_m^2 := \mathrm{tr}(\Sigma_m\Sigma_0^{-1}) \times \nu_m = (d-m)\nu_m.$$

Thus, termination when $\sigma_m < \epsilon$, for some tolerance $\epsilon > 0$ that is user-specified, might be a useful criterion. However, Proposition 2 is extremely conservative, and since Proposition 10 establishes a much faster rate of convergence for $\|\boldsymbol{x}_m - \boldsymbol{x}^*\|_{\Sigma_0^{-1}}$ in the case of BayesCG search directions, this is likely to be an overcautious stopping criterion in the case of BayesCG. Furthermore, since this involves a data-driven estimate of scale, the term ν_m is not uniformly decreasing with m. As a result, in practise we advocate using a more traditional termination criterion based upon monitoring the residual; see Golub and Van Loan (2013, Section 11.3.8) for more detail. Further research is needed to establish whether the posterior distribution can provide a useful termination criterion.

Full pseudocode for the BayesCG method, including the termination criterion, is presented in Algorithm 1. Two algebraic simplifications have been exploited here relative to the presentation in the main text; these are described in detail in Section S2 of the supplement. A Python implementation can be found at github.com/jcockayne/bcg.

Algorithm 1 Computation of the posterior distribution described in Proposition 6. The implementation is optimised compared to that given in Proposition 6; see Supplement S2 for detail. Further note that, for clarity, all required matrix-vector multiplications have been left explicit, but for efficiency these should be calculated once-per-loop and stored. Σ_m can be computed from this output as $\Sigma_m = \Sigma_0 - \Sigma_F \Sigma_F^\top$.

procedure BAYESCG($A, \boldsymbol{b}, \boldsymbol{x}_0, \Sigma_0, \epsilon, m_{\max}$) ▷ ($\epsilon$ the tolerance)
  Σ_F initialised to a matrix of size $(d \times 0)$ ▷ ($m_{\min}$ the minimum # iterations)
  $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - A\boldsymbol{x}_0$ ▷ ($m_{\max}$ the maximum # iterations)
  $\tilde{\boldsymbol{s}}_1 \leftarrow \boldsymbol{r}_0$
  $\tilde{\nu}_0 \leftarrow 0$
  for $m = 1, \ldots, m_{\max}$ **do**
    $E^2 \leftarrow \tilde{\boldsymbol{s}}_m^\top A \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m$
    $\alpha_m \leftarrow \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}{E^2}$
    $\boldsymbol{x}_m \leftarrow \boldsymbol{x}_{m-1} + \alpha_m \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m$
    $\boldsymbol{r}_m \leftarrow \boldsymbol{r}_{m-1} - A\boldsymbol{x}_m$
    $\Sigma_F \leftarrow [\Sigma_F, \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m / E]$
    $\tilde{\nu}_m \leftarrow \tilde{\nu}_{m-1} + \frac{(\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1})}{E^2}$
    if $\|r_m\|_2 < \epsilon$ **then**
      break
    end if
    $\beta_m \leftarrow \frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}$
    $\tilde{\boldsymbol{s}}_{m+1} \leftarrow \boldsymbol{r}_m + \beta_m \tilde{\boldsymbol{s}}_m$
  end for
  $\nu_m \leftarrow \tilde{\nu}_m / m$
  return $\boldsymbol{x}_m, \Sigma_F, \nu_m$
end procedure

6 Numerical Results

In this section two numerical studies are presented. First we present a simulation study in which theoretical results are verified. Second we present an application to electrical impedance tomography, a challenging medical imaging technique in which linear systems must be repeatedly solved.

6.1 Simulation Study

The first experiment in this section is a simulation study, the goals of which are to empirically examine the convergence properties of BayesCG. Additional results which compare the output of the algorithm to the probabilistic approach of Hennig (2015) are presented in Section S4.2 of the supplementary material.

For our simulation study, a matrix A was generated by randomly drawing its eigenvalues $\lambda_1, \ldots, \lambda_d$ from an exponential distribution with parameter γ. A sparse, symmetric-positive definite matrix with these eigenvalues was then drawn using the

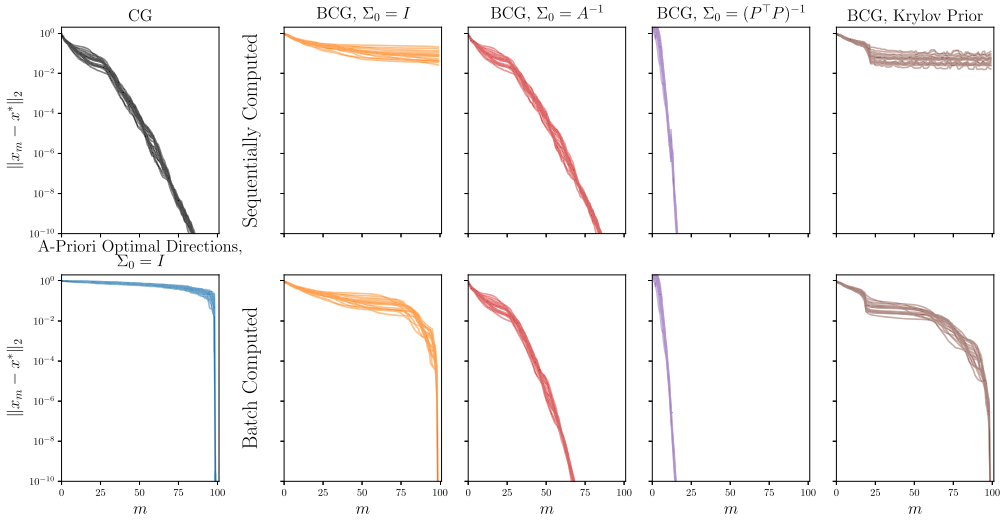


Figure 1: Convergence in mean of BayesCG (BCG). For several independent test problems, $\boldsymbol{x}^* \sim \mu_{\text{ref}}$, the error $\|\boldsymbol{x}_m - \boldsymbol{x}^*\|_2$ was computed. The standard CG method (top left) was compared to variants of BayesCG (right), corresponding to different prior covariances Σ_0. The search directions used for BayesCG were either computed sequentially (top right) or in batch (bottom right). For comparison, the *a priori* optimal search directions for BayesCG are shown in the bottom left panel.

MATLAB function `sprandsym`. The proportion of non-zero entries was taken to be 20%. Subsequently, a vector $\boldsymbol{x}^*$ was drawn from a reference distribution μ_{ref} on $\mathbb{R}^d$, and $\boldsymbol{b}$ was computed as $\boldsymbol{b} = A\boldsymbol{x}^*$. Throughout, the reference distribution for $\boldsymbol{x}^*$ was taken to be $\mu_{\text{ref}} = \mathcal{N}(\mathbf{0}, I)$. For this experiment $d = 100$ and $\gamma = 10$. In all cases the prior mean was taken to be $\boldsymbol{x}_0 = \mathbf{0}$. The prior covariance was alternately taken to be $\Sigma_0 = I$, $\Sigma_0 = A^{-1}$ and $\Sigma_0 = (P^\top P)^{-1}$ where P was a preconditioner found by computing an incomplete Cholesky decomposition with zero fill-in. This decomposition is simply a Cholesky decomposition in which the (approximate) factor $\hat{L}$ has the same sparsity structure as A. The preconditioner is then given by $P = \hat{L}\hat{L}^\top$. The matrix $\hat{L}$ can be computed at a computational cost of $\mathcal{O}(\text{nnz}(A)^3)$ where $\text{nnz}(A)$ is the number of nonzero entries of A. Furthermore, P^{-1} is cheap to apply because its Cholesky factor is explicit. In addition, the Krylov subspace prior introduced in Section 4.1 has been examined. While it has been noted that the choice $\Sigma_0 = A^{-1}$ is generally impractical, for this illustrative example A^{-1} has been computed directly. Additional experimental results which apply the methodology discussed in this section to higher-dimensional problems is presented in Section S5.

Point Estimation In Figure 1 the convergence of the posterior mean $\boldsymbol{x}_m$ from BayesCG is contrasted with that of the output of CG, for many test problems $\boldsymbol{x}^*$ with a fixed sparse matrix A. To study the impact of the numerical breakdown of conjugacy in the search directions, two choices of search directions were used; the *sequentially-computed*

search directions are those described in Proposition 7, while the *batch-computed* search directions enforce conjugacy by employing a full Gram-Schmidt orthogonalisation. The batch-computed search directions are thus given by:

$$\tilde{\boldsymbol{s}}_m^C := \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \left\langle \boldsymbol{s}_i^C, \boldsymbol{r}_{m-1} \right\rangle_{A\Sigma_0 A^\top} \boldsymbol{s}_{m-1}^C$$
$$\boldsymbol{s}_m^C := \tilde{\boldsymbol{s}}_m^C / \|\tilde{\boldsymbol{s}}_m^C\|_{A\Sigma_0 A^\top}.$$

These search directions are mathematically identical to the BayesCG search directions $\{\boldsymbol{s}_i\}_{i=1}^m$, but explicitly orthogonalising with respect to all $m-1$ previous directions ensures that numerical conjugacy is maintained. However, note that when the batch-computed search directions are used an additional loop of complexity $\mathcal{O}(m)$ must be performed. Thus, the cost of the BayesCG algorithm with batch-computed search directions is $\mathcal{O}(m^2 d^2)$.

As expected from the result of Proposition 9, the convergence of the BayesCG mean vector when $\Sigma_0 = I$ is slower than in CG. In this case, the speed of convergence for BayesCG is controlled by $\kappa(A^\top A)$ which is larger than the corresponding $\kappa(A)$ for CG. The *a priori* optimal search directions also appear to yield a slower rate than the BayesCG search directions, owing to the fact that they do not exploit knowledge of $\boldsymbol{b}$. Similarly as expected, the posterior mean when $\Sigma_0 = A^{-1}$ is identical to the estimate for $\boldsymbol{x}_m$ obtained from CG. The fastest rate of convergence was achieved when $\Sigma_0 = (P^\top P)^{-1}$, which provides a strong motivation for using a preconditioner prior if such a preconditioner can be computed, though note that a preconditioned CG method would converge at a yet faster rate gated by $\kappa(P^{-1}A)$.

In the lower row of Figure 1 the convergence is shown when using batch-computed directions. Here convergence appears to be faster than when using the sequentially-computed directions, at correspondingly higher computational cost. The batch-computed directions provide an exact solution after $m = d$ iterations, in contrast to the sequentially-computed directions, for which numerical conjugacy may not hold.

Convergence for the Krylov subspace prior introduced in Section 4.1 is plotted in the right-hand column. The size of the computed subspace was set to $n = 20$, with $M = A$. The matrix Φ was chosen to be diagonal, with $\Phi_{ii} = [2\sigma\xi^i]^2$, as discussed in Section S3.2 of the supplement. Here $\sigma = \|\boldsymbol{x}^*\|_A$ and $\xi = \frac{\kappa(A)-1}{\kappa(A)+1}$, as these quantities are easily computable in this simplified setting. The remaining parameter was set to $\gamma = 0.01$, so that low prior weight was given to the remaining subspaces. With the sequentially computed directions significant numerical instability is observed starting at $m = 20$. This does not occur with the batch computed directions, where a jump in the convergence rate is seen at this iteration.

Posterior Covariance The full posterior output from BayesCG will now be evaluated. In Figure 2, the convergence rate of $\mathrm{tr}(\Sigma_m)$ is plotted for the same set of problems just described to numerically verify the result presented in Proposition 3. It is clear that when the more informative CG or BayesCG search directions are used, the rate of

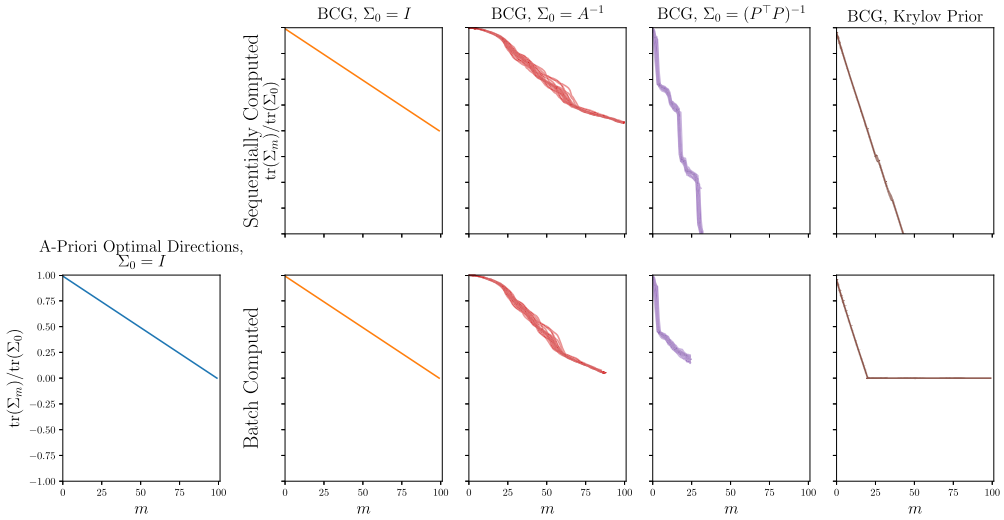


Figure 2: Convergence in posterior covariance of BayesCG (BCG), as measured by $\mathrm{tr}(\Sigma_m)$. The experimental setup was as in Figure 1, here with $\mathrm{tr}(\Sigma_m)/\mathrm{tr}(\Sigma_0)$ plotted.

contraction in the posterior mean does not transfer to the posterior covariance. In the remaining columns of the figure, $\mathrm{tr}(\Sigma_m)$ appears to contract at a roughly linear rate, in contrast to the exponential rate observed for $\boldsymbol{x}_m$. This indicates that tightening the bound provided in Proposition 3 is unlikely to be possible. Furthermore, in the last two columns of Figure 2, the impact of numerical non-conjugacy is apparent as the posterior covariance takes on negative values at around $m = 20$.

Uncertainty Quantification We now turn to an assessment of the quality of the uncertainty quantification (UQ) being provided. The same experimental setup was used as in the previous sections, however rather than running each variant of BayesCG to $m = d$, instead $m = 10$ was used to ensure that UQ is needed. To avoid the issue of negative covariances seen in Figure 2, the batch-computed search directions were used throughout.

First, the Gaussian version of BayesCG from Proposition 6 was evaluated. To proceed we used the following argument: When the UQ is well-calibrated, we could consider $\boldsymbol{x}^*$ as plausibly being drawn from the posterior distribution $\mathcal{N}(\boldsymbol{x}_m, \Sigma_m)$. Note that Σ_m is of rank $d - m$, but assessing uncertainty in its null space is not of interest as in this space $\boldsymbol{x}^*$ has been determined exactly. Since Σ_m is positive semidefinite, it has the singular-value decomposition

$$\Sigma_m = U \begin{bmatrix} D & 0_{d-m,m} \\ 0_{m,d-m} & 0_{m,m} \end{bmatrix} U^\top,$$

where $0_{m,n}$ denotes an $m \times n$ matrix of zeroes, $D \in \mathbb{R}^{(d-m)\times(d-m)}$ is diagonal and $U \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. The first $d - m$ columns of U, denoted U_{d-m}, form

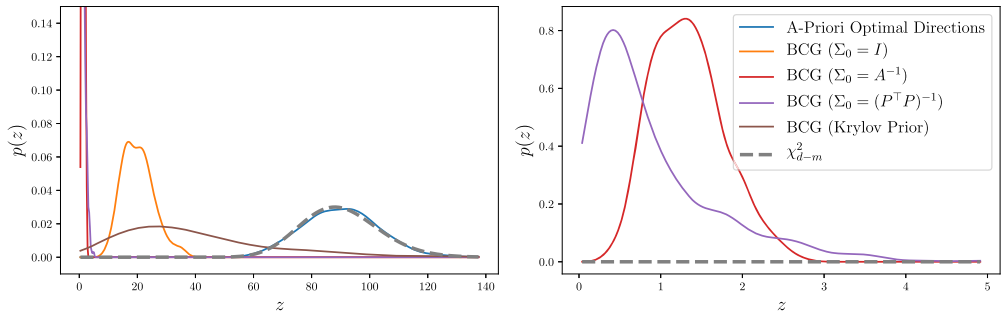


Figure 3: Assessment of the uncertainty quantification provided by the Gaussian BayesCG method, with different choices for search directions and Σ_0. Plotted are kernel density estimates for the statistic Z based on 500 randomly sampled test problems. These are compared with the theoretical distribution of Z when the posterior distribution is well-calibrated. The right panel zooms in on the estimate for $\Sigma_0 = A^{-1}$ and $\Sigma_0 = (P^\top P)^{-1}$.

a basis of range(Σ_m), the subspace of $\mathbb{R}^d$ in which $\boldsymbol{x}^*$ is still uncertain. Under this hypothesis we can therefore derive a test statistic

$$U_{d-m} D^{-\frac{1}{2}} U_{d-m}^\top (\boldsymbol{x}^* - \boldsymbol{x}_m) \sim \mathcal{N}(\boldsymbol{0}, I_{d-m})$$
$$\implies \quad Z(\boldsymbol{x}^*) := \| D^{-\frac{1}{2}} U_{d-m}^\top (\boldsymbol{x}^* - \boldsymbol{x}_m) \|_2^2 \sim \chi^2_{d-m},$$

where here $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Note that the pre-factor U_{d-m} is not necessary in the final expression as $\| \cdot \|_2$ is unitarily invariant.

Thus to evaluate the UQ we can draw many test problems $\boldsymbol{x}^* \sim \mu_{\text{ref}}$, evaluate the test statistic $Z(\boldsymbol{x}^*)$ and compare the empirical distribution of this statistic to χ^2_{d-m}. If the posterior distribution is well-calibrated we expect that the empirical distribution of the test statistic will resemble χ^2_{d-m}. An overly-conservative posterior will exhibit a "left-shift" in its density, as $\boldsymbol{x}_m$ is closer to $\boldsymbol{x}^*$ than was expected. Likewise, an overly confident posterior will exhibit a "right-shift".

In Figure 3 the empirical distribution of the statistic Z was compared to its theoretical distribution for different prior covariances. The empirical distributions were plotted as kernel density estimates based upon the computed statistic for 500 sampled test problems. Clearly the *a priori* optimal directions provide well-calibrated UQ, while for BayesCG the UQ provided by the posterior was overly-conservative for the prior covariances $\Sigma_0 = I$, A^{-1} and $(P^\top P)^{-1}$. This reflects the fact that the search directions encode knowledge of $\boldsymbol{b}$, but this knowledge is not reflected in the likelihood model used for conditioning, as discussed following Proposition 7. Furthermore, note that the quality of the UQ seems to worsen as the convergence rate for $\boldsymbol{x}_m$ improves, with $\Sigma_0 = (P^\top P)^{-1}$ providing the most conservative UQ.

For the Krylov subspace prior, which encodes intuition for how search directions are selected, better UQ was provided. Though the empirical distribution of Z is not

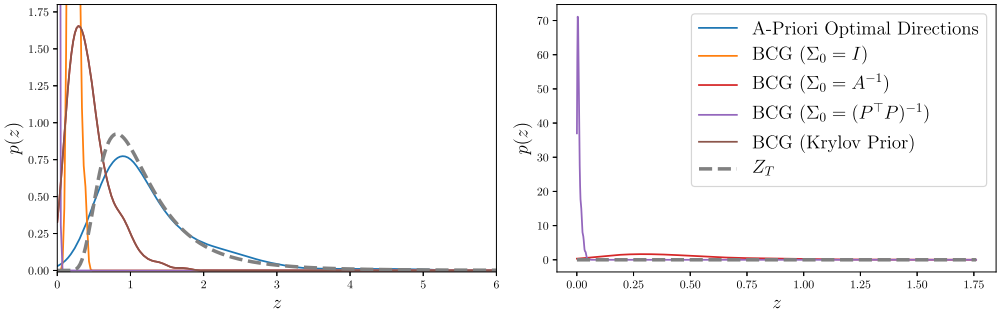


Figure 4: Assessment of the uncertainty quantification provided by the multivariate t BayesCG method, for the same prior covariances and search directions as in Figure 3.

identical to the theoretical distribution, the supports of the two distributions overlap. Thus, while the Krylov subspace prior does not fully remedy the issue caused by the use of $\boldsymbol{b}$ in the search directions, some improvement is seen through the incorporation of knowledge of $\boldsymbol{b}$ into the prior.

Next we assessed the UQ provided by the multivariate t posterior presented in Proposition 11. A similar procedure was followed to the Gaussian case, with a different test statistic. Let $S \sim \mathcal{N}(\mathbf{0}, I)$, $T \sim \mathsf{MVT}_m(\boldsymbol{\mu}, \Sigma)$ and $U \sim \chi^2_m$. Then, it can be shown that

$$\frac{1}{\sqrt{m}} U_{d-m} D^{-\frac{1}{2}} U_{d-m}^\top (T - \boldsymbol{\mu}) \overset{d}{=} \frac{S}{\sqrt{U}} \implies \frac{1}{m} \| D^{-\frac{1}{2}} U_{d-m}^\top (T - \boldsymbol{\mu}) \|_2^2 \overset{d}{=} \frac{\|S\|_2^2}{U}.$$

In the present setting, $\boldsymbol{\mu} = \boldsymbol{x}_m$ and $\Sigma = \Sigma_m$. Furthermore $\|S\|_2^2 \sim \chi^2_{d-m}$. Lastly, multiplying both sides by $m/(d-m)$ we have

$$Z(\boldsymbol{x}^*) := \frac{1}{d-m} \| D^{-\frac{1}{2}} U_{d-m}^\top (\boldsymbol{x}_m - \boldsymbol{x}^*) \|_2^2 \overset{d}{=} \frac{\frac{\|S\|_2^2}{(d-m)}}{\frac{U}{m}}.$$

The ratio on the right-hand-side is known to follow an $F(d-m, m)$ distribution. In Figure 4 the empirical distribution of the test statistic $Z(\boldsymbol{x}^*)$ was compared to the $F(d-m, m)$ distribution for each of the posterior distributions considered. Again, the posterior distribution based on the *a priori* optimal search directions was well-calibrated, while the posteriors from BayesCG trade fast convergence in mean with well-calibrated UQ. As before, BayesCG with the Krylov subspace prior appears to provide the best-calibrated UQ of the (practically useful) priors considered.

Note that in both Figure 3 and Figure 4, for the choice $\Sigma_0 = (P^\top P)^{-1}$, which has the most rapidly converging mean in Figure 1, poor UQ properties are observed, making this otherwise appealing choice impractical. To address this we have explored a heuristic procedure for setting ν_m, which aims to match the posterior spread to an appropriate estimate of the error $\|\boldsymbol{x}_m - \boldsymbol{x}^*\|_2$. This procedure is reported in Section S4 of the supplement, along with experimental results based upon it.

6.2 Electrical Impedance Tomography

Electrical impedance tomography (EIT) is an imaging technique used to estimate the internal conductivity of an object of interest (Somersalo et al., 1992). This conductivity is inferred from measurements of voltage induced by applying stimulating currents through electrodes attached to its boundary. EIT was originally proposed for medical applications as a non-invasive diagnostic technique (Holder, 2004), but it has also been applied in other fields, such as engineering (Oates et al., 2019).

The physical relationship between the inducing currents and resulting voltages can be described by a PDE, most commonly the complete electrode model (CEM) (Cheng et al., 1989). Consider a domain $D \subset \mathbb{R}^n$ representing the object of interest, where typically $n = 2$ or $n = 3$. Denote by ∂D the boundary of D, and let $\sigma(\boldsymbol{z})$ denote the conductivity field of interest, where $\boldsymbol{z} \in D$. Denote by $\{e_l\}_{l=1}^L$ the L electrodes, where each $e_l \subset \partial D$ and $e_l \cap e_m = \emptyset$ whenever $l \neq m$. Let $v(\boldsymbol{z})$ denote the voltage field, and let $\{I_{i,l}\}_{l=1}^L$ denote the set of stimulating currents applied to the electrodes. Let $\{V_{i,l}^\sigma\}_{l=1}^L$ denote the corresponding voltages, and let $\boldsymbol{n}$ denote the outward-pointing normal vector on ∂D. The subscript i here is to distinguish between multiple *stimulation patterns* which are generally applied in sequence and are of relevance to the inversion problem for determining $\sigma(\boldsymbol{z})$ later. Denote by $\{\zeta_l\}_{l=1}^L$ the contact impedance of each electrode. The contact impedances are used to model the fact that the contact between the electrode and the boundary of the domain is imperfect. Then the CEM is given by

$$\begin{aligned}
-\nabla \cdot (\sigma(\boldsymbol{z}) \nabla v(\boldsymbol{z})) &= 0 && \boldsymbol{z} \in D \\
\int_{e_l} \sigma(\boldsymbol{z}) \frac{\partial v}{\partial \boldsymbol{n}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} &= I_{i,l} && l = 1, \ldots, L \\
\sigma(\boldsymbol{z}) \frac{\partial v}{\partial \boldsymbol{n}}(\boldsymbol{z}) &= 0 && \boldsymbol{z} \in \partial D \setminus \bigcup_{l=1}^L e_l \\
v(\boldsymbol{z}) + \zeta_l \sigma(\boldsymbol{z}) \frac{\partial v}{\partial \boldsymbol{n}}(\boldsymbol{z}) &= V_{i,l}^\sigma && \boldsymbol{z} \in e_l,\ l = 1, \ldots, L.
\end{aligned} \tag{13}$$

A solution of this PDE is the tuple $(v(\boldsymbol{z}), V_{i,1}^\sigma, \ldots, V_{i,L}^\sigma)$, consisting of the interior voltage field and the voltage measurements on the electrodes. The numerical solution of this PDE can be reduced to the solution of a linear system of the form in (1), as will shortly be explained.

Having specified the PDE linking stimulating currents to resulting voltages, it remains to describe the approach for determining $\sigma(\boldsymbol{z})$ from noisy voltage measurements. These physical voltage measurements are denoted by the matrix $V \in \mathbb{R}^{L \times (L-1)}$, where $V_{i,l}$ is the voltage obtained from stimulation pattern i at electrode l. The recovery problem can be cast in a Bayesian framework, as formalised in Dunlop and Stuart (2016). To this end, a prior distribution for the conductivity field is first posited and denoted μ_σ. Then, the posterior distribution μ_σ^V is defined through its Radon–Nikodym derivative with respect to the prior as

$$\frac{\mathrm{d}\mu_\sigma^V}{\mathrm{d}\mu_\sigma}(\sigma) \propto \exp(-\Phi(\sigma; V)),$$

where $\Phi(\sigma; V)$ is known as a *potential* function and $\exp(-\Phi(\sigma; V))$ is the likelihood. This posterior distribution is for an infinite-dimensional quantity-of-interest and is generically nonparametric, thus sampling techniques such as the preconditioned Crank–Nicolson (pCN) algorithm Cotter et al. (2013) are often employed to access it. Such algorithms require repeated evaluation of $\Phi(\sigma; V)$ and thus the repeated solution of a PDE. Thus, there is interest in ensuring that $\Phi(\sigma; V)$ can be computed at low cost.

Experimental Setup The experimental set-up is shown in Figure 5a and is due to Isaacson et al. (2004). This is described in detail in Section S.6 of the supplement. In the absence of specific data on the accuracy of the electrodes, and for convenience, the observational noise was assumed to be Gaussian with standard deviation $\delta = 1$. This implies a potential of the form:

$$\Phi(\sigma; V) = \sum_{i=1}^{L-1} \sum_{l=1}^{L} \frac{(V_{i,l} - V_{i,l}^{\sigma})^2}{2\delta^2} = \frac{1}{2\delta^2}(\vec{V} - \vec{V}^{\sigma})^{\top}(\vec{V} - \vec{V}^{\sigma}),$$

where V^{σ} is the matrix with (i,l)-entry $V_{i,l}^{\sigma}$. The notation $\vec{V} \in \mathbb{R}^{L(L-1)}$ denotes the vectorisation of V, formed by concatenating columns of V into a vector as described in Section S4.2.

Apart from in pathological cases, there is no analytical solution to the CEM and thus evaluating $\Phi(\sigma; V)$ requires an approximate solution of (13). Here a finite-element discretisation was used to solve the weak form of (13), as presented in Dunlop and Stuart (2016) and described in more detail in the supplement. This discretisation results in a sparse system of equations $A\boldsymbol{x}^* = \boldsymbol{b}$, where A is in this context referred to as a *stiffness matrix*. To compute A and $\boldsymbol{b}$, standard piecewise linear basis functions were used, and the computations were performed using the `FEniCS` finite-element package. A fine discretisation of the PDE will necessarily yield a high-dimensional linear system to be solved. We propose to use BayesCG to approximately solve the linear system, and propagate the solver uncertainty into the inverse problem associated with recovery of the conductivity field. In essence, this provides justification for small values of m to be used in the linear solver and yet ensure that the inferences for σ remain valid.

The Gaussian version of BayesCG was used throughout, as described in Proposition 6. Thus, assume that the output from BayesCG is $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_m, \Sigma_m)$. The finite element approximation to the voltages $V_{i,l}^{\sigma}$ is linearly related to the solution $\boldsymbol{x}^*$ of the linear system, so that BayesCG implies a probability model for the voltages of the form $\vec{V}^{\sigma} \sim \mathcal{N}(\vec{V}_m^{\sigma}, \Sigma_m^{\sigma})$ for some $\vec{V}_m^{\sigma}$ and Σ_m^{σ}; for brevity we leave these expressions implicit. The approach proposed is to derive a new potential $\hat{\Phi}$, obtained by marginalising the posterior distribution output from BayesCG in the likelihood. It is straightforward to show that, for the Gaussian likelihood, this marginalisation results in the new potential

$$\hat{\Phi}(\sigma; V) = \frac{1}{2}(\vec{V} - \vec{V}_m^{\sigma})^{\top}(\Sigma_m^{\sigma} + \delta^2 I)^{-1}(\vec{V} - \vec{V}_m^{\sigma}).$$

Thus, the new likelihood $\exp(-\hat{\Phi}(\sigma; V))$ is still Gaussian, but with a covariance inflated by Σ_m^{σ}, which describes the level of accuracy in the BayesCG solver. It will be shown

that replacing Φ with $\hat{\Phi}$ leads to a posterior distribution $\hat{\mu}_\sigma^V$ for the conductivity field which is appropriately to account for the accuracy of BayesCG.

Throughout this section the prior distribution over the conductivity field was taken to be a centered log-Gaussian distribution, $\log(\sigma) \sim \mathcal{GP}(0, k)$, with a Matérn 5/2 covariance as given by:

$$k(\boldsymbol{z}, \boldsymbol{z}') = a\left(1 + \frac{\sqrt{5}\|\boldsymbol{z} - \boldsymbol{z}'\|_2}{\ell} + \frac{5\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}\|\boldsymbol{z} - \boldsymbol{z}'\|_2}{\ell}\right).$$

The length-scale parameter ℓ was set to $\ell = 1.0$, while the amplitude a was set to $a = 9.0$ to ensure that where the posterior distribution is concentrated has significant probability mass under the prior. Results for application of BayesCG to the solution of this PDE, also known as the *forward problem*, are similar to those described in the previous section and are presented in Section S6 of the supplement.

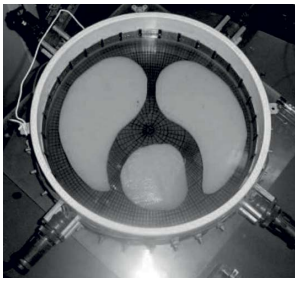
Inverse Problem In this section, the solution to the inverse problem when using the BayesCG potential $\hat{\Phi}$ is compared to the posterior obtained from the exact potential Φ. In the latter case CG was used to solve the system to convergence to provide a brute-force benchmark. For BayesCG, the prior was centered, $\boldsymbol{x}_0 = \mathbf{0}$, and the preconditioner prior covariance, $\Sigma_0 = (P^\top P)^{-1}$, was used. BayesCG was run to $m = 80$ iterations, for the mesh with $N_d = 64$. This mesh results in a linear system with $d = 311$, so 80 iterations represents a relatively small amount of computational effort.

In Figure 5 the posterior distribution over the conductivity field is displayed. In Figures 5b and 5c, respectively, the exact posterior mean and the posterior mean from BayesCG are plotted. Note that, as indicated in the previous section, many of the features of the conductivity field have been recovered even though a relatively small number of iterations have been performed. In Figure 5d the ratio of the pointwise posterior standard deviation from BayesCG to that in CG is plotted. Clearly, throughout the entire spatial domain, the posterior distribution has a larger standard deviation, showing that the posterior uncertainty from BayesCG has successfully been transferred to the posterior over the conductivity field. This results in a posterior distribution which is wider to account for the fact that an imperfect solver was used to solve the forward problem. Overall, the integrated standard deviation over the domain is 0.0365 for BayesCG, while for the exact posterior it is 0.0046.

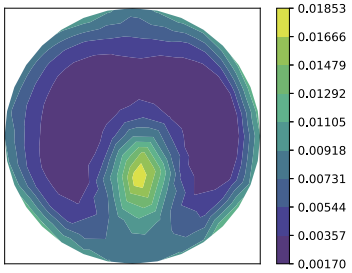
This example illustrates how BayesCG could be used to relax the computational effort required in EIT in such a way that the posterior is widened to account for the imperfect solution to the forward problem. This setting, as well as other applications of this method, should be explored in more detail in future work.

7 Conclusion and Discussion

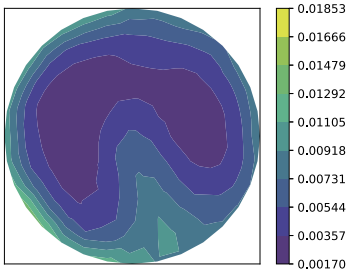
In this paper we have introduced and theoretically analysed the Bayesian conjugate gradient method, a Bayesian probabilistic numerical method for the solution of linear systems of equations. Given the ubiquity of linear systems in numerical computation,



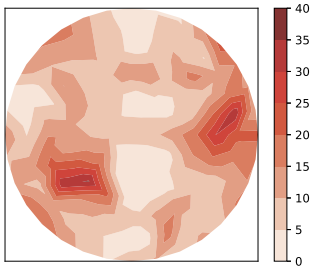
(a) Set-up for the experiment described in Section 6.2.



(b) Exact posterior mean for $\log \sigma$



(c) BCG-based posterior mean for $\log \sigma$



(d) Ratio of point-wise posterior standard deviation, for BayesCG-based compared to exact.

Figure 5: Comparison of the posterior distribution over the conductivity field, when using BayesCG to solve the linear system arising from the forward problem compared to using standard CG.

the question of how to approximate their solution is fundamental. Contrary to CG and other classical iterative methods, BayesCG outputs a probability distribution, providing a principled quantification of uncertainty about the solution after exploring an m-dimensional subspace of $\mathbb{R}^d$. Through the numerical example in Section 6.2 we have shown how this output could be used to make meaningful inferences in applied problems, with reduced computational cost in terms of iterations performed. This could be applied to a broad range of problems in which solution of large linear systems is a bottleneck, examples of which have been given Section 1.1.

Prior Choice Prior choice was discussed in detail. An important question that arises here is to what extent the form of the prior can be relaxed. Indeed, in many applied settings information is known about $\boldsymbol{x}^*$ which cannot be encoded into a Gaussian prior. For example, the solution of PDEs is often known to be sign-constrained. When encoding this information into the prior it is likely that the conjugacy properties exploited to construct a closed-form posterior will be lost. Then, interrogating such posteriors would require sampling techniques such as the numerical disintegration procedure of Cockayne

et al. (2017), which would incur a dramatically higher cost. Research to determine what prior knowledge can be encoded (either exactly or approximately) without sacrificing numerical performance will be an important future research direction.

It was shown how a numerical analyst's intuition that the conjugate gradient method "tends to work well" can be encoded into a Krylov-based prior. This went some way towards compensating for the fact that the search directions in BayesCG are constructed in a data-driven manner which is not explicitly acknowledged in the likelihood. Alternative heuristic procedures for calibrating the UQ were explored in the supplement, Section S4.3. An important problem for future research will be to provide practical and theoretically justified methods for ensuring the posterior UQ is well-calibrated.

Computational Cost and Convergence The computational cost of BayesCG is only a constant factor higher than that of CG. However, the convergence rates reported in Section 3 can be slower than those of CG. To achieve comparable convergence rates, the prior covariance Σ_0 must be chosen to counteract the fact that the rate is based on $\kappa(\Sigma_0 A^\top A)$ rather than $\kappa(A)$, and this can itself incur a substantial computational cost. Future work will focus on reducing the cost associated with BayesCG.

Supplementary Material

Supplementary Material for "Bayesian Conjugate-Gradient Method" (DOI: 10.1214/19-BA1145SUPP; .pdf).

References

Ajiz, M. A. and Jennings, A. (1984). "A robust incomplete Choleski-conjugate gradient algorithm." *International Journal for Numerical Methods in Engineering*, 20(5): 949–966. MR0749826. doi: https://doi.org/10.1002/nme.1620200511. 938

Allaire, G. and Kaber, S. M. (2008). *Numerical Linear Algebra*, volume 55 of *Texts in Applied Mathematics*. Springer New York. MR2365296. doi: https://doi.org/10.1007/978-0-387-68918-0. 938

Bartels, S. and Hennig, P. (2016). "Probabilistic Approximate Least-Squares." In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*. 939

Benzi, M. (2002). "Preconditioning Techniques for Large Linear Systems: A Survey." *Journal of Computational Physics*, 182(2): 418–477. MR1941848. doi: https://doi.org/10.1006/jcph.2002.7176. 939

Besag, J. and Green, P. J. (1993). "Spatial statistics and Bayesian computation." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 25–37. MR1210422. 938

Bogachev, V. I. (1998). *Gaussian Measures*, volume 62. American Mathematical Society Providence. MR1642391. doi: https://doi.org/10.1090/surv/062. 938

Bramble, J. H., Pasciak, J. E., and Xu, J. (1990). "Parallel Multilevel Preconditioners." *Mathematics of Computation*, 55(191): 1–22. MR1023042. doi: https://doi.org/10.2307/2008789. 938

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. (2018). "Probabilistic Integration: A Role in Statistical Computation?" arXiv:1512.00933. 938

Calvetti, D., Pitolli, F., Somersalo, E., and Vantaggi, B. (2018). "Bayes Meets Krylov: Statistically Inspired Preconditioners for CGLS." *S*IAM Review, 60(2): 429–461. MR3797727. doi: https://doi.org/10.1137/15M1055061. 939

Cheng, K.-S., Isaacson, D., Newell, J. C., and Gisser, D. G. (1989). "Electrode models for electric current computed tomography." *IEEE Transactions on Biomedical Engineering*, 36(9): 918–924. MR1080512. doi: https://doi.org/10.1137/0150096. 957

Cockayne, J., Oates, C. J., Ipsen, I. C. F., and Girolami, M. (2019). "Supplementary Material for "Bayesian Conjugate-Gradient Method"." *Bayesian Analysis*. doi: https://doi.org/10.1214/19-BA1145SUPP. 941

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2017). "Bayesian Probabilistic Numerical Methods." arXiv:1702.03673. 939, 942, 943, 945, 961

Cockayne, J., Oates, C., Sullivan, T. J., and Girolami, M. (2016). "Probabilistic Meshless Methods for Partial Differential Equations and Bayesian Inverse Problems." arXiv:1605.07811v1. 938, 948

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). "MCMC methods for functions: Modifying old algorithms to make them faster." *Statistical Science*, 28(3): 424–446. MR3135540. doi: https://doi.org/10.1214/13-STS421. 958

Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA. MR2270673. doi: https://doi.org/10.1137/1.9780898718881. 938

Diaconis, P. (1988). "Bayesian numerical analysis." *Statistical Decision Theory and Related Topics IV*, 1: 163–175. MR0927099. 943

Dunlop, M. M. and Stuart, A. M. (2016). "The Bayesian formulation of EIT: Analysis and algorithms." *Inverse Problems and Imaging*, 10: 1007–1036. MR3610749. doi: https://doi.org/10.3934/ipi.2016030. 957, 958

Evans, L. (2010). *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. Providence, Rhode Island: American Mathematical Society, second edition. MR2597943. doi: https://doi.org/10.1090/gsm/019. 937, 938

Fasshauer, G. E. (1999). "Solving differential equations with radial basis functions: Multilevel methods and smoothing." *Advances in Computational Mathematics*, 11(2–3): 139–159. MR1731694. doi: https://doi.org/10.1023/A:1018919824891. 938

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian Data Analysis*, volume 2. CRC press Boca Raton, FL. MR3235677. 949

Golub, G. H. and Van Loan, C. F. (2013). *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition. MR3024913. 948, 950

Hennig, P. (2015). "Probabilistic Interpretation of Linear Solvers." *SIAM Journal on Optimization*, 25(1): 234–260. MR3301314. doi: https://doi.org/10.1137/140955501. 939, 940, 944, 951

Hennig, P., Osborne, M. A., and Girolami, M. (2015). "Probabilistic numerics and uncertainty in computations." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 471(2179): 20150142. MR3378744. doi: https://doi.org/10.1098/rspa.2015.0142. 939

Hestenes, M. R. and Stiefel, E. (1952). "Methods of conjugate gradients for solving linear systems." *Journal of Research of the National Bureau of Standards*, 49(6): 409. MR0060307. 938, 943

Holder, D. S. (2004). *Electrical Impedance Tomography: Methods, History and Applications*. CRC Press. 957

Isaacson, D., Mueller, J. L., Newell, J. C., and Siltanen, S. (2004). "Reconstructions of chest phantoms by the D-bar method for electrical impedance tomography." *IEEE Transactions on Medical Imaging*, 23(7): 821–828. 958

Larkin, F. M. (1972). "Gaussian measure in Hilbert space and applications in numerical analysis." *The Rocky Mountain Journal of Mathematics*, 2(3): 379–421. MR0303193. doi: https://doi.org/10.1216/RMJ-1972-2-3-379. 938, 939

Liesen, J. and Strakos, Z. (2012). *Krylov Subspace Methods*. Principles and Analysis. Oxford University Press. MR3024841. 938, 947

Oates, C. J., Cockayne, J., Aykroyd, R. G., and Girolami, M. (2019). "Bayesian Probabilistic Numerical Methods in Time-Dependent State Estimation for Industrial Hydrocyclone Equipment." *Journal of the American Statistical Association*. To appear. 957

Owhadi, H. (2015). "Bayesian numerical homogenization." *Multiscale Modeling & Simulation*, 13(3): 812–828. MR3369060. doi: https://doi.org/10.1137/140974596. 948

Parker, A. and Fox, C. (2012). "Sampling Gaussian distributions in Krylov spaces with conjugate gradients." *SIAM Journal on Scientific Computing*, 34(3): B312–B334. MR2970281. doi: https://doi.org/10.1137/110831404. 938

Rasmussen, C. E. (2004). "Gaussian Processes in Machine Learning." In *Advances in Intelligent Data Analysis VIII*, 63–71. Berlin, Heidelberg: Springer Berlin Heidelberg. 938

Reinarz, A., Dodwell, T., Fletcher, T., Seelinger, L., Butler, R., and Scheichl, R. (2018). "Dune-composites – A new framework for high-performance finite element modelling of laminates." *Composite Structures*, 184: 269–278. 938

Roeckner, E., Bäuml, G., Bonaventura, L., Brokopf, R., Esch, M., and Giorgetta, M. (2003). "The atmospheric general circulation model ECHAM 5. PART I: Model description." Technical report, MPI für Meteorologie. 939

Saad, Y. (1994). "ILUT: A dual threshold incomplete LU factorization." *Numerical Linear Algebra with Applications*, 1(4): 387–402. MR1306700. doi: https://doi.org/10.1002/nla.1680010405. 938

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition. MR1990645. doi: https://doi.org/10.1137/1.9780898718003. 938, 939

Schäfer, F., Sullivan, T. J., and Owhadi, H. (2017). "Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity." arXiv:1706.02205. 938

Shewchuk, J. R. (1994). "An introduction to the conjugate gradient method without the agonizing pain." Technical report. 964

Somersalo, E., Cheney, M., and Isaacson, D. (1992). "Existence and uniqueness for electrode models for electric current computed tomography." *SIAM Journal on Applied Mathematics*, 52(4): 1023–1040. MR1174044. doi: https://doi.org/10.1137/0152060. 957

Stuart, A. M. (2010). "Inverse problems: A Bayesian perspective." *Acta Numerica*, 19: 451–559. MR2652785. doi: https://doi.org/10.1017/S0962492910000061. 941

Tikhonov, A. N. (1963). "On the solution of ill-posed problems and the method of regularization." In *Doklady Akademii Nauk*, volume 151, 501–504. Russian Academy of Sciences. MR0162377. 941

Traub, J. F., Wasilkowski, G. W., and Woźniakowski, H. (1988). *Information-Based Complexity*. Computer Science and Scientific Computing. Academic Press, Inc., Boston, MA. With contributions by A. G. Werschulz and T. Boult. MR0958691. 941, 944

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001). "Spatiotemporal Hierarchical Bayesian Modeling Tropical Ocean Surface Winds." *Journal of the American Statistical Association*, 96(454): 382–397. MR1939342. doi: https://doi.org/10.1198/016214501753168109. 938

Acknowledgments

Chris J. Oates and Mark Girolami were supported by the Lloyd's Register Foundation programme on Data-Centric Engineering. Ilse C.F. Ipsen was supported in part by NSF grant DMS-1760374. Mark Girolami was supported by EPSRC grants [EP/R034710/1, EP/R018413/1, EP/R004889/1, EP/P020720/1], an EPSRC Established Career Fellowship [EP/J016934/3] and a Royal Academy of Engineering Research Chair.

This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The authors are grateful to Shewchuk (1994). Techniques described therein formed the basis of some of the proofs in this paper. They would also like to thank Philipp Hennig and Simon Bartels for their help in the implementation of their algorithm and Tim Sullivan for his feedback on the manuscript.

Invited Discussion

Philipp Hennig*†

I congratulate Jon Cockayne and his colleagues for reinvigorating the thread of research on probabilistic linear algebra methods. Linear solvers take a special role in the ongoing effort to develop a consistent interpretation of numerical algorithms as autonomous Bayesian inference agents that has come to be known under the label *probabilistic numerics*. Linear solvers form the lowest layer of the numerics stack, used as sub-routines of more elaborate operations like solving integrals, and differential equations; constrained and non-linear optimization. One may thus perhaps expect them to be particularly easy to understand and interpret as inference agents. Alas, describing linear algebra routines in the language of probabilistic inference has proven challenging, and even the present, deeply thoughtful paper, does not yet provide a final answer. How hard a task this is depends on what we actually mean when we speak of a *probabilistic linear algebra method*:

- Should it use novel kinds of prior information — in particular, information that can not already be encoded via a pre-conditioner in present-day methods — to outperform the **point estimates** of classic linear solvers on more specific classes of problems, either in the worst or the average case? Such methods likely require exploration strategies different from that of the classic iterative solvers, motivated and constructed from internal, iteratively refined model of the problem. They will thus be called **active** in this text.
- Or is our goal less ambitious, and we "only" want to endow existing linear algebra routines with structured and calibrated **uncertainty**? Structured error estimates could be used to more aggressively control the use of computational resources, or they could be propagated up the stack, along with the solution $\boldsymbol{x}^*$, to be used by whoever called upon the linear algebra agent, similar to how a LAPACK solver returns a matrix factorization alongside $\boldsymbol{x}^*$ for later use. A method that achieves this post-hoc kind of uncertainty while "watching" another linear solver do its thing will be called **passive** below.

Cockayne and his colleagues have focused on passive functionality in their empirical evaluation and left the construction of search directions s_m to the tried and trusted Lanczos process. My own 2015 paper contained an active formulation (more below), but I only succeeded in re-constructing conjugate gradients, not in improving upon it. Active probabilistic solvers remain elusive. The first part of this comments reviews why this advanced functionality is so hard to achieve. In the second part, I will highlight

*University of Tübingen and Max Planck Institute for Intelligent Systems, Tübingen, Germany, ph@tue.mpg.de

†The author gratefully acknowledges financial support by the European Research Council through ERC StG Action 757275 PANAMA.

that for passive solvers, while Cockayne and colleagues have proposed a self-contained framework, significantly better uncertainty quantification is possible if one is willing to use empirical Bayesian calibration with strong priors. I believe this highlights that there is still a gap between what we have now and how good even passive solvers could be.

1 The Classic Point Estimates Remain Undefeated

A key reason why probabilistic linear solvers, especially active ones, are harder to define than one may think is that the "linear problem"

$$A\boldsymbol{x}^* = \boldsymbol{b} \tag{1}$$

studied in the Discussion Paper is not actually *linear* where it matters: In the matrix A. The standard recipe of probabilistic numerics is to capture a numerical problem by assigning a probability distribution to the intractable or computationally demanding part of the problem, then use tractable computations linked to the intractable variable through a likelihood function to derive a posterior. In Equation (1), the matrix is the source of computational complexity, so it would seem natural to assign a distribution $p(A)$ to it. But of course, if $C\boldsymbol{y}^* = \boldsymbol{b}$, then $(A+C)\boldsymbol{z}^* = \boldsymbol{b}$ does **not** imply $\boldsymbol{z}^* = \boldsymbol{x}^* + \boldsymbol{y}^*$, so simple models like a Gaussian distribution on A do not yield elegant frameworks from which one could derive a posterior on $\boldsymbol{x}^*$. But we *must* find simple models, precisely because linear algebra is such a low-level operation that a complicated solution is not acceptable. In my earlier paper (Hennig, 2015), I studied two different ways to address this challenge, one of which[1] is to assign a Gaussian prior to the *inverse* A^{-1} (presupposing its existence)

$$p(A) = \mathcal{N}(\overrightarrow{A^{-1}}; \overrightarrow{A_0^{-1}}, \Xi_0),$$

where $\overrightarrow{X}$ is a vectorization operation on matrices, $A_0^{-1} \in \mathbb{R}^{N\times N}$ is a prior mean and $\Xi \in \mathbb{R}^{d^2\times d^2}$ is a symmetric positive covariance matrix.[2] Such a prior is convenient, first because it allows tractable inference on A^{-1} from linear observations in the form of matrix-vector products $AS_m = Y_m$ (which, assuming exact computations, is equivalent to the linear observation $S_m = A^{-1}Y_m$), and secondly because any Gaussian posterior $p(A^{-1} \mid S_m, Y_m) = \mathcal{N}(\overrightarrow{A^{-1}}; \overrightarrow{A_m^{-1}}, \Xi_m)$ on A^{-1} directly implies a posterior, also Gaussian, on the linear projection given by the solution $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$, given by

$$p(\boldsymbol{x}^* \mid S_m, Y_m) = \mathcal{N}(\boldsymbol{x}^*; A_m^{-1}\boldsymbol{b}, (I\otimes \boldsymbol{b})\Xi_m(I\otimes \boldsymbol{b})^\intercal). \tag{2}$$

Cockayne et al. advocate for instead assigning a prior directly on the solution vector x^*. This is closely related to the matrix inverse prior in the sense of Equation (2)

[1] The other option discussed in the op.cit. is to use a Gaussian prior on A, infer a posterior mean on A, then use the convenient form of this mean estimate with the matrix inversion lemma to construct a *point estimate* (with only approximately Gaussian uncertainty) for $A^{-1}\boldsymbol{b} = \boldsymbol{x}^*$. This form will be used in Section 2.1.

[2] Unless stated otherwise, this comment strives to use the same notation as the discussion paper.

but, as the Discussion Paper shows, the vector prior has some practical advantages.[3] In particular, it is much easier to handle, avoiding tedious derivations with Kronecker-structured covariances on matrix-valued objects. A recent working paper (Bartels et al., 2018) studies the connections between vector- and matrix-valued priors in detail.

Nevertheless, there are also arguments for the matrix prior, and the strongest one may be the desire for an active probabilistic solver. The method of conjugate gradients, like virtually all numerical methods, is not an estimation rule, but an autonomous agent following an adaptive policy to construct its search directions. If what we are aiming for is a probabilistic method to improve upon Conjugate Gradients (CG), we can not just sit and watch CG (i.e. the Lanczos process) at work. We want an algorithm, roughly, of the following form:

Algorithm 1 Template for an active probabilistic linear solver.

1 **procedure** SOLVE($A(.), b, p_0$) ▷ probabilistic linear solver with prior p_0
2 $\quad \boldsymbol{x}_0 \leftarrow \mathbb{E}_{p_0}(A^{-1})\boldsymbol{b}$ ▷ initial guess
3 $\quad \boldsymbol{r}_0 \leftarrow A\boldsymbol{x}_0 - \boldsymbol{b}$
4 $\quad$ **while** $\|r_i\| >$ tol, i++ **do** ▷ run to convergence in residual
5 $\quad\quad \boldsymbol{d}_i \leftarrow -\mathbb{E}_{p_{i-1}}(A_0^{-1})\boldsymbol{r}_{i-1}$ ▷ compute optimization direction
6 $\quad\quad \boldsymbol{z}_i \leftarrow A\boldsymbol{d}_i$ ▷ **observe**
7 $\quad\quad \alpha_i \leftarrow -\frac{\boldsymbol{d}_i^\intercal \boldsymbol{r}_{i-1}}{\boldsymbol{d}_i^\intercal \boldsymbol{z}_i}$ ▷ optimal step-size
8 $\quad\quad \boldsymbol{x}_i \leftarrow \boldsymbol{x}_{i-1} + \alpha_i \boldsymbol{d}_i$ ▷ update estimate for $\boldsymbol{x}$
9 $\quad\quad \boldsymbol{r}_i \leftarrow \boldsymbol{r}_{i-1} + \alpha_i \boldsymbol{z}_i$ ▷ new gradient at $\boldsymbol{x}_i$
10 $\quad\quad p_i \leftarrow \text{INFER}(D, Z)$ ▷ estimate A or A^{-1}
11 $\quad$ **end while**
12 $\quad$ **return** $\boldsymbol{x}_i$
13 **end procedure**

This is almost exactly the textbook skeleton of conjugate gradients, except for line 10 of this algorithm, where the Lanczos correction β_m is replaced by a probabilistic estimate. As its input, the algorithm takes a full prior probability distribution on A^{-1} in place of single pre-conditioner (which can be thought of as a point-estimate of A^{-1}). My 2015 paper contains a construction of certain choices[4] of Gaussian priors that turn the above algorithm exactly into CG. They can be extended to allow for the use of a preconditioner. But, sadly, to my knowledge no tractable prior choice has been found yet that would improve upon the behavior of preconditioned CG either in the worst or average case, not even on an interesting subset of the symmetric positive cone that could not also be addressed with minor variations on CG itself. The Discussion Paper corroborates that achieving active performance is indeed hard.

[3] Cockayne et al. note that the matrix prior is not invariant under left-preconditioning while the vector prior is. One may debate whether this is a bug or a feature: As the authors note in their paper, preconditioners amount to prior information; so it might make sense that a preconditioner should change the prior, not vanish within it.

[4] The short summary is that there are two separate families of priors $p_{a,b}(A)$ (using the aforementioned point estimate construction for A^{-1}) and $p_{c,d}(A^{-1})$ on the matrix and its inverse, respectively, both indexed by two scalars $a, b, c, d \in \mathbb{R}_+$, which recover CG. In Section 2.1 I will use a particularly simple choice among them.

2 Uncertainty Quantification Can Still be Improved

Even if we abandon the hope of beating CG at its own game — point-estimating $\boldsymbol{x}_*$ — we may still want to surround CG's estimate with a meaningful error bar. Cockayne et al. propose a way to do so in Section 4.2 of their paper, employing the well-known Student-t mechanism. Their construction is elegant in its analytic simplicity, but the problem with this approach is that conjugate Gauss-inverse-Gamma hierarchical inference assumes that the observations are iid., i.e. that the search directions are chosen at random. A well-known property of CG (implied by Propositions 9 & 10 in the Discussion Paper) is that since it is a Krylov method, its search directions, loosely speaking,[5] explore the eigenvectors $\boldsymbol{v}_i$ and -values λ_i of A in roughly descending order of the values $\lambda_i \boldsymbol{v}_i^\intercal \boldsymbol{r}_0$. This behavior must be taken into account when calibrating the posterior.

A simple alternative, suggested, but not properly fleshed out in the experimental section of my 2015 paper, is to impose some structural — potentially dangerous, but also potentially valuable — prior assumptions on A's eigenvalue spectrum and adapt this model based on the numbers collected by CG. A prime candidate for useful information for this purpose is given by the projections

$$a(m) = \frac{\boldsymbol{s}_m^\intercal A \boldsymbol{s}_m}{\|\boldsymbol{s}_m\|^2}, \tag{3}$$

which are collected by CG during its run anyway (cf. line 7 in Algorithm 1 above, the scaling difference between $\boldsymbol{s}_i$ and $\boldsymbol{d}_i$ cancels) and thus are available for uncertainty quantification at no computational overhead. I will use the remainder of this comment to construct an indicative example of how such a process could work. For space, I will focus on calibrating a matrix-valued prior on A. Analogous formulations can be used to calibrate Gaussian priors on A^{-1} and thus $\boldsymbol{x}_*$.

2.1 Active Uncertainty Quantification for CG

Consider a run of CG on the problem from Equation (1), yielding a collection S_m, Y_m of vectors as defined in the Discussion Paper, satisfying $AS_m = Y_m$ and $S^\intercal Y_m = \mathrm{diag}_i(\boldsymbol{s}_i^\intercal \boldsymbol{y}_i)$ (since CG's search directions are A-conjugate. And since we assume A is symmetric positive definite (spd), so is this diagonal matrix). For notational simplicity, I will use the standardized form $\tilde{Y}_m := Y_m (S_m^\intercal Y)^{-1/2}$.

The *other* way (compared to the one in Section 1) to define CG outlined in Hennig (2015) is as arising from Algorithm 1 with a zero-mean Gaussian prior of covariance

$$\Xi_{ij,k\ell} = \mathrm{cov}(A_{ij}, A_{k\ell}) = {}^1\!/_2 (W_{0,ik} W_{0,j\ell} + W_{0,i\ell} W_{jk}), \tag{4}$$

where W is chosen such that $W_0 S = Y$. An obvious but intractable choice is $W_0 = A$, as Cockayne et al. note, too. We could set $W_0 = A_M$ in what might be called an empirical Bayesian approach. This would ensure $WS = Y$ But then the posterior variance vanishes, because it is given by

$$W_M = W_0 - W_0 S (S^\intercal W_0 S)^{-1} S^\intercal W_0 = \tilde{Y}\tilde{Y}^\intercal - \tilde{Y}\tilde{Y}^\intercal = 0. \tag{5}$$

[5] For a more precise statement, see, e.g. Nocedal and Wright (2006, Equation 5.29).

I believe there is a lot left to do in the middle ground between these two extremes, the intractable and the trivial one, to estimate a prior covariance parameter W_0 that is consistent with the choice of A_M and achieves non-zero posterior variance according to different desiderata for calibration. In this text, let us see how far we can go if we allow ourselves to make regularity assumptions. What I will propose below is not unrelated to the "heuristic calibration" proposed in Appendix S4.3 of the discussion paper; but it uses an explicit extrapolation of error to estimate average case-error rather than an upper bound. And it remains consistent with the *actions* of CG, even if it, again, can not hope to improve upon them.

We are looking for a tractable way to choose W_0 so it acts like A on the span of S (and is thus consistent with CG's actions), and also estimate its effect on the complement of this space using regularity assumptions about A to achieve a calibrated error estimate. The general form for W_0 (also given in Hennig (2015), and related to the Krylov-subspace prior of Cockayne et al.) is thus

$$W_0 = \tilde{Y}\tilde{Y}^\intercal + (I - S(S^\intercal S)^{-1}S^\intercal)\Omega(I - S(S^\intercal S)^{-1}S^\intercal), \tag{6}$$

with a general spd matrix Ω. The projection matrices surrounding Ω ensure that it only acts on the space *not* covered by $A_M = \tilde{Y}\tilde{Y}^\intercal$ In absence of further prior knowledge about A, we might choose the simple scalar form $\Omega = \omega I$, which simplifies Equation (6) to

$$\begin{aligned} W_0 &= \tilde{Y}\tilde{Y}^\intercal + \omega(I - S(S^\intercal S)^{-1}S^\intercal) \quad \text{and} \\ W_M &= W_0 - W_0 S(S^\intercal W_0 S)S^\intercal W_0 = W_0 - \tilde{Y}\tilde{Y}^\intercal = \omega(I - S(S^\intercal S)^{-1}S^\intercal). \end{aligned} \tag{7}$$

The scale ω can then be interpreted as scaling the remaining uncertainty over the entire null-space of S, the space not yet explored by CG. How should ω be set (this task is related to setting the covariance scale ν — Section 4.2 in the Discussion Paper)?

Figure 1 shows results constructed from a run of CG on a specific matrix: The Sarcos dataset (Vijayakumar and Schaal, 2000)[6] is a popular, simple test setup for kernel regression. It was used to construct a kernel ridge regression problem $A\boldsymbol{x} = \boldsymbol{b}$ with the symmetric positive definite matrix

$$A := k_{XX} + \sigma^2 I \in \mathbb{R}^{14\,828 \times 14\,828}, \tag{8}$$

with noise level $\sigma = 0.1$, and where k is the isotropic radial Gaussian kernel

$$k(a,b) = \exp\left(-\frac{1}{2\eta^2}\sum_i (a_i - b_i)^2\right) \qquad \text{for } a, b \in \mathbb{R}^{21} \tag{9}$$

with length-scale $\eta = 2$. On this problem, standard CG was run for $M = 300$ steps. The plot shows the sequence of projections of A arising as $a(m)$ (cf. Equation 3) which can

[6] See also §2.5 in Rasmussen and Williams (2006). The data can be found at http://www.gaussianprocess.org/gpml/data/. It contains a time series of trajectories mapping 21-dimensional inputs in $\mathbb{R}^{21}$ (positions, velocities and accelerations, respectively, of 7 joints of a robot arm) to 7 output torques. The first of these torques is typically used as the target in $\mathbb{R}$ for regression, as was done here, too. The entire training set contains 44 484 input-output pairs. For the purposes of this experiment, to allow some comparisons to analytical values, this was thinned by a factor of 1/3, to $N = 14\,828$ locations. The data was standardized to have vanishing mean and unit covariance.

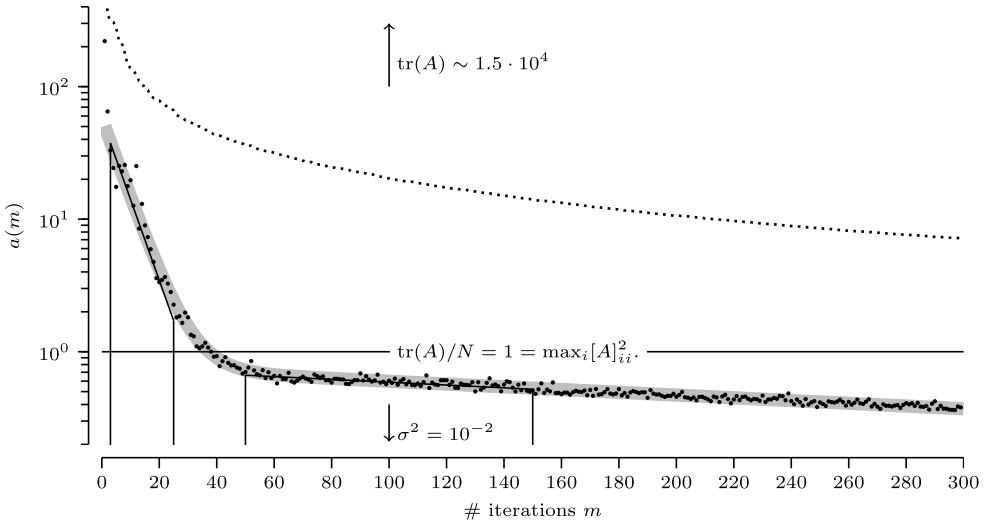


Figure 1: Scaled matrix projections collected by CG for the Sarcos problem. The plot shows, as a function of the iteration index m, the value of the scalar projection $a(m)$ (black circles). These observations for $m = [3, 4, \ldots, 24, 50, 51, \ldots, 149]$ (vertical lines) are used to estimate a structured regression model for $a(m)$ (details in text, local models as thin black lines). The regression line is shown as a broad gray curve. The plot also indicates the two strict upper ($\mathrm{tr}(A)$) and lower (σ^2) bounds on the eigenvalues of A, which are clearly loose (outside the plot range). The constant value of diagonal elements, which happens to be known for this problem, is indicated by a horizontal line. For comparison, the plots also shows the 300 largest eigenvalues of A (dotted).

be constructed from the m-th iteration of CG (cf. Algorithm 1). From Equation (8), there are straightforward upper and lower bounds both for elements of A and for $a(m)$:

$$\sigma^2 \leq \frac{v^\intercal A v}{v^\intercal v} \leq \mathrm{tr}\, A \qquad \forall\, v \in \mathbb{R}^N. \tag{10}$$

But both these bounds are relatively loose, as the Figure shows. We also know from the functional form of k_{XX} (Equation 9) that $[A]_{ij} \leq 1 + \sigma^2 \delta_{ij}$.

Although this experimental setup is not particularly challenging for CG, it shows the typical and much-studied empirical behaviour of this iterative solver: The collected projections rapidly decay as the solver explores an expanding sub-space of relevant directions. A small number of initial steps (in this example, from $m = 1$ to about $m = 50$) reveal large projections a_m. Then comes a 'kink' in the plot, followed by a relatively continuous decay over a longer time scale. It is tempting to think of the first phase as revealing dominant 'structure' in A while the remainder is 'noise'. But since there are $N - 50 \gg 50$ such suppressed directions, their overall influence is significant. Also note that, while the a_m exhibit a decaying trend, they do not in fact decrease monotonically.

Predicting General Matrix Projections

One possible use for the posterior mean A_M that emerges as a "side-effect" of CG is to construct an estimator $A_M \boldsymbol{v}$ for matrix-vector multiplications $A\boldsymbol{v}$ with arbitrary $\boldsymbol{v} \in \mathbb{R}^N$. If this is the target application, then ω should be set to provide the right scale for such projections. A way to achieve this is to use the auxiliary data from the CG run shown in Figure 1 and try to predict the value $\boldsymbol{v}^\intercal A\boldsymbol{v}$ for $\boldsymbol{v} \in \mathbb{R}^N$ outside of the span S.

Since this statement involves aspects of the matrix not yet observed, it must hinge on either prior knowledge or prior assumptions about A. Assumptions may be wrong, of course. In practice, we may face a trade-off between the desires for tight error estimates and formal guarantees. We also have to balance the cost and quality of a model: If calibrated uncertainty matters, we may be willing to invest time into building a good model. If error estimation is an afterthought, a loose upper bound may suffice.

Figure 1 shows such a model in grey. It was fitted as a function of the form

$$\hat{a}(m) = \sigma^2 + 10^{\xi_1 + \xi_2 m} + 10^{\xi_3 + \xi_4 m} \tag{11}$$

with real constants $\xi_1, \ldots, \xi_4$. The constants were fitted by an ad-hoc least squares scheme over the transformed observations $\log_{10} a(m)$ on the region $m = [3, 4, \ldots, 24]$ (for ξ_1, ξ_2) and $m = [50, 51, \ldots, 149]$ (for ξ_3, ξ_4, the contribution of the previous term can be essentially ignored in this domain, simplifying the fit). It is then possible to estimate the *average* value of a_m from any particular stopping point M to N, to get one first candidate for the scale ω:

$$\omega_{\text{projections}} := \frac{1}{N - M} \sum_{m=M}^{N} \hat{a}(m). \tag{12}$$

Under the posterior $p(A) = \mathcal{N}(A; A_M, W_M \otimes W_M)$, the marginal over a matrix projection $A\boldsymbol{v} = (I \otimes \boldsymbol{v})^\intercal \vec{A}$ is

$$p(A\boldsymbol{v}) = \mathcal{N}\left(A\boldsymbol{v}; A_M\boldsymbol{v}, \underbrace{\frac{1}{2}(W_M \boldsymbol{v}^\intercal W_M \boldsymbol{v} + (W_M \boldsymbol{v})(\boldsymbol{v}^\intercal W_M))}_{=:\Sigma_{\boldsymbol{v}}}\right). \tag{13}$$

Figure 2 shows results from experiments with random directions $\boldsymbol{v}$ whose elements were drawn from shifted Gaussian, uniform and binary distributions on the Sarcos set-up described above. The predicted scale $\omega = 0.02$ (fitted using Equation 12) is not perfect — indeed it would be surprising if it were, given the ad-hoc nature of the fitted model. But it captures the scale of the vector elements quite well. The more conservative estimate $\omega = 1$, let alone the hard upper bound $\omega = \operatorname{tr} A$, would give radically larger scales so wide that the corresponding probability density function (pdf) would not even be visible in the plot.

The plot also shows that the sampled matrix projections are modeled well by a Gaussian distribution. This is not surprising: since the elements of v are drawn iid. from

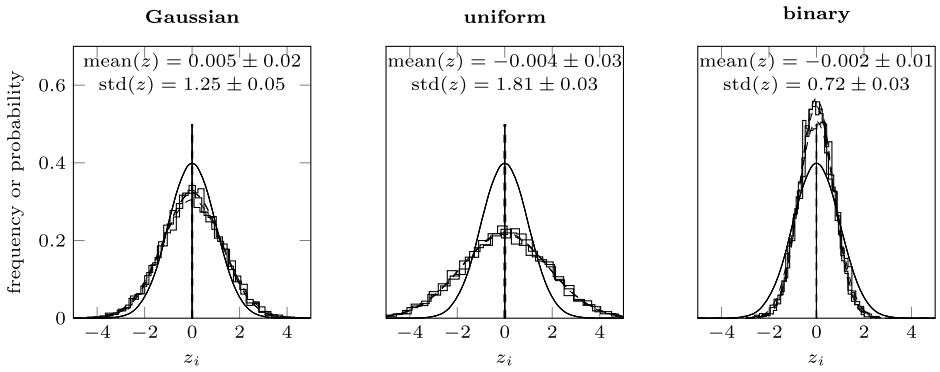


Figure 2: Predicting the projection $A\boldsymbol{v}$ after $M = 300$ steps of CG on the Sarcos problem (Figure 1). For this plot, the elements of a vector $\boldsymbol{v} \in \mathbb{R}^{14\,828}$ were drawn iid. from a unit-mean Gaussian distribution, a uniform distribution, and as binary values $[v]_i = \{-1, 1\}$, respectively for each panel. To simplify computations, all steps were only performed on a subset of 4000 randomly chosen indices of $A\boldsymbol{v}$. The plot investigates the standardized variable $z := \Sigma_{\boldsymbol{v}}^{-1/2}(A\boldsymbol{v} - \mathbb{E}_{p(A|S,Y)}(A\boldsymbol{v}))$, using the matrix square root of $\Sigma_{\boldsymbol{v}} := \mathrm{cov}_{p(A|,S,Y)}(A\boldsymbol{v})$, the predictive covariance of $A\boldsymbol{v}$ under the posterior. If the probabilistic model were perfectly calibrated, the elements of this vector should be distributed like independent standard Gaussian random variables (solid black pdf for reference). The plot shows three independent realizations of v, the empirical distribution of the actual elements z_i (histogram), and an empirical fit (dashed) of a Gaussian pdf to the elements of z. These means and standard deviations of such empirical distributions (estimated from 10 realisations of $\boldsymbol{v}$, not shown) are printed in each plot. This figure uses the value for the scale ω fitted as described in Section 2.1, which gives $\omega_{\text{projections}}(M = 300) = 0.02$ (in other words, for the naïve setting $\omega = 1$, the shown solid pdf would be about 50 times wider).

a distribution p_v, hence the Central Limit Theorem (CLT) applies, and the elements

$$[Av]_i = \sum_j [A]_{ij} v_j \tag{14}$$

are approximately Gaussian distributed with mean and variance

$$\mathbb{E}_{p_v}([Av]_i) = \mathbb{E}_{p_v}([v]_j) \sum_j [A]_{ij}, \quad \text{and} \quad \mathrm{var}_{p_v}([Av]_i) = \mathrm{var}_{p_v}([v]_j) \sum_j [A]_{ij}^2. \tag{15}$$

While this Gaussian *shape* of the plot is not surprising, it *is* reassuring that the solver's Gaussian posterior on $A\boldsymbol{v}$ manages to capture the two moments of this distribution rather well, even though it has no access to p_v. Constructed in this way from a hand-crafted regression model, the variance $\Sigma_{\boldsymbol{v}}$ offers a simple and computationally cheap, way to infer the right scale for the aspects of A not captured by the CG-mean estimate.

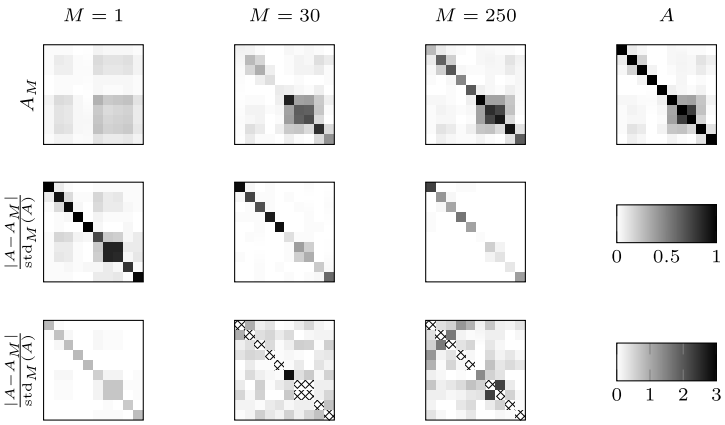


Figure 3: Contraction of the posterior measure on A during a CG run on the Sarcos problem (cf. Figure 1). Top row: Posterior mean A_M on a randomly sampled subset of 10 index pairs, after $M = 1, 30, 250$ steps of CG (the full $14\,828 \times 14\,828$ matrix is too big to print). The target sub-matrix of A is shown for comparison on the right. Middle and bottom row: Absolute estimation error $|A - A_M|$, scaled element-wise by the posterior standard deviation. The middle row shows the choice for the scaling $\omega = 1$ (gray-scale from 0 to 1), the bottom row for $\omega_{\text{projection}} \approx 0.02$, the value used for Figure 2 (gray-scale from 0 to 3). 'Miss-scaled' entries, were the scaled error is outside of the gray-scale range, are marked with a cross-hatch pattern.

Predicting Individual Matrix Elements

If the CLT helped in the above section, the estimation task becomes harder when we consider more explicit, deterministic aspects of the latent matrix A, such as individual matrix elements $[A]_{ij}$. The posterior marginal distribution on these scalars under the model (4), conditioned on CG's observations, is

$$p([A]_{ij} \mid Y, S) = \mathcal{N}\left([A]_{ij}; [A_M]_{ij}, \frac{1}{2}([W_M]_{ii}[W_M]_{jj} + [W_M]_{ij}^2)\right). \tag{16}$$

An argument in Hennig (2015, Equations (3.2)–(3.4)) shows that there is no scalar ω (in fact, not even a full spd matrix W_0) such that the posterior variance is a tight prediction for the approximation error on *all* matrix elements. So we are forced to choose between a worst-case, hard error bound on all matrix elements, and a reasonably scaled error estimate that can be too small for some elements.

Figure 3 shows the progression of the posterior distribution for an increasing number of CG steps on the Sarcos task (the Figure only shows a small sub-set of the matrix elements for visibility). The top row shows the posterior mean converging towards the true matrix A. The bottom row shows the element-wise posterior marginal variance,[7]

[7]Under the joint posterior, these matrix elements are correlated. So one should not try to build a mental histogram of the numbers in the plotted matrix and ask about their relative frequencies.

for the two different choices of the scale parameter ω: The top row shows the result of a choice for which the posterior standard-deviation becomes a hard upper bound on the square estimation error,

$$\frac{1}{2}([W_0]_{ii}[W_0]_{jj} + [W_0]_{ij}^2) > [A - A_M]_{ij}^2. \tag{17}$$

For symmetric positive definite matrices (which obey $[A]_{ij}^2 \leq [A]_{ii}[A]_{jj}$), a simple way to ensure this is to set $\omega^2 = \max_i [A]_{ii}^2 \geq \max_{ij}[A]_{ij}$ (which can be found in $\mathcal{O}(N)$). For the example of the Sarcos matrix this bound is 1.01. The Figure confirms that the variance provides an upper bound, but also shows this bound to be rather loose for off-diagonal elements (i.e. by far the majority of matrix elements!). The plots' bottom row shows the scaling achieved by using the estimated value $\omega_{\text{projections}} = 0.02$ already used in Figure 2. This is not a hard bound, but the plot shows this scaled estimate to provide a better average error estimate for off-diagonal elements (values of value ~ 1, note differing color scale in each row). On the diagonal, the error estimates can be far off, though. Some of the outliers (marked by a cross-hatch pattern) can have ratios of true to estimated error beyond 10.

3 Summary

The work of Jon Cockayne and his colleagues brings us closer to a theory of probabilistic iterative liner solvers. But there remains a big gap between what one would like to have from such methods and what they are currently able to do. In this comment, I have offered an optimistic case-study: For a single, rather simple least-squares problem, it is possible to construct a heuristic algorithm of linear cost ($\mathcal{O}(N)$, for estimations involving a symmetric positive definite matrix in $\mathbb{R}^{N\times N}$) that yields relatively well-calibrated, average-case error estimates (I have focused on elements of the matrix A, but the same methodology also allows estimating A^{-1} and $\boldsymbol{x}_*$). These experiments are anecdotal at best; they come without theory and generalization. But I hope that they spark an interest in others to construct exactly these formal foundations. After all, linear solvers are the bedrock of computation. It is hard to find an area with more application leverage.

Ultimately, though, the ambition should be to build *active* probabilistic linear solvers, that use strong prior information to improve upon classic (pre-conditioned) general solvers. Such algorithms remain elusive, even though generative prior information is often available in linear problems. The key challenge here is the formulation of expressive but tractable families of priors.

References

Bartels, S., Cockayne, J., Ipsen, I. C., Girolami, M., and Hennig, P. (2018). "Probabilistic Linear Solvers: A Unifying View." *arXiv preprint* arXiv:1810.03398. 968

Hennig, P. (2015). "Probabilistic Interpretation of Linear Solvers." *SIAM Journal on Optimization*, 25(1): 210–233. MR3301314. doi: https://doi.org/10.1137/140955501. 966, 967, 968, 969, 970, 974

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media. MR2244940. 969

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT. MR2514435. 970

Vijayakumar, S. and Schaal, S. (2000). "Locally Weighted Projection Regression: Incremental Real Time Learning in High Dimensional Space." In Langley, P. (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 1079–1086. 970

Acknowledgments

I am grateful to Simon Bartels for in-depth discussions on the topic of this discussion over several years, and Jonathan Wenger for a close reading of a draft of this text.

Invited Discussion

Xudong Li* and Ethan X. Fang†,

We congratulate Cockayne and his colleagues for this nice paper. We find that it is very interesting to interpret a classical numerical method – the conjugate gradient (CG), as a probabilistic method from the Bayesian perspective. This approach provides natural uncertainty quantification for the error of the solution, which is important in practice.

Purely from the numerical perspective, we provide a re-interpretation of the proposed Bayesian CG method. In particular, we point out that the proposed method is equivalent to applying the classical CG to the following equivalent but reformulated linear system

$$\underbrace{A\Sigma_0 A^T}_{M}\underbrace{\left\{(\Sigma_0 A^T)^{-1}x\right\}}_{\widetilde{x}} = b. \tag{1}$$

This is a well-known trick in numerical analysis/optimization that when A is ill-conditioned, we may use this approach to accelerate the computation by carefully choosing a Σ_0. Indeed, the classical conjugate direction method (including CG as a special case), when applied for solving (1), takes the following updating rule (e.g., equation (5.2) in Nocedal and Wright (2006)) with respect to $\widetilde{x}$:

$$\widetilde{x}_m = \widetilde{x}_{m-1} + s_m s_m^T(b - M\widetilde{x}_{m-1}),$$

where search directions s_i, $i = 1, \ldots, m$, are assumed to be M-orthonormal (i.e., $A\Sigma_0 A^T$-orthonormal). Let $x_m = (\Sigma_0 A^T)\widetilde{x}_m$ for all $m \geq 1$. We have that

$$x_m = x_{m-1} + (\Sigma_0 A^T)s_m s_m^T(b - A\Sigma_0 A^T\widetilde{x}_{m-1}) = x_{m-1} + (\Sigma_0 A^T)s_m s_m^T(b - Ax_{m-1}),$$

which is exactly the updating formula in Proposition 6. Moreover, it is not difficult to see that ignoring the Σ_F and $\tilde{\nu}$ parts, Algorithm 1 in the paper can be obtained via applying the standard form of classical CG (e.g., Algorithm 5.2 in Nocedal and Wright (2006)) for solving problem (1). Meanwhile, it seems that line 10 in Algorithm 1 should be $r_m = r_{m-1} - \alpha_m A\tilde{s}_m$.

Now from the above equivalence, we can simplify the proof of convergence results (Propositions 9 and 10) for the Bayesian CG without much difficulty. As a simple illustration, in the subsequent analysis, we show how Proposition 9 can be obtained by using the above equivalence property. Firstly, we see from the classical analysis of the CG for (1) that

$$\widetilde{x}_m = \underset{\widetilde{x}\in\widetilde{x}_0+K_{m-1}(M,r_0)}{\arg\min} \|\widetilde{x} - \widetilde{x}^*\|_M,$$

*School of Data Science, Shanghai Center for Mathematical Sciences, Fudan University, Shanghai 200438, China, lixudong@fudan.edu.cn

†Department of Statistics, Pennsylvania State University, University Park, PA 16802, USA, xxf13@psu.edu

where $r_0 = b - M\widetilde{x}_0 = b - Ax^0$ and $\widetilde{x}^* = M^{-1}b = (\Sigma_0 A^T)^{-1}A^{-1}b = (\Sigma_0 A^T)^{-1}x^*$. Then, it holds that

$$(\Sigma_0 A^T)^{-1}x_m = \underset{(\Sigma_0 A^T)^{-1}x \in (\Sigma_0 A^T)^{-1}x_0 + K_{m-1}(M,r_0)}{\arg\min} \|(\Sigma_0 A^T)^{-1}(x - x^*)\|_{A\Sigma_0 A^T}. \quad (2)$$

Since

$$K_{m-1}(M, r_0) = \mathrm{span}(r_0, Mr_0, M^2 r_0, \ldots, M^{m-1}r_0),$$

and

$$\Sigma_0 A^T M^k = (\Sigma_0 A^T A)^k \Sigma_0 A^T, \quad \forall\, k = 1, \ldots, m-1,$$

we have that

$$\begin{aligned}
\Sigma_0 A^T K_{m-1}(M, r_0) &= \mathrm{span}(\Sigma_0 A^T r_0, \Sigma_0 A^T M r_0, \Sigma_0 A^T M^2 r_0, \ldots, \Sigma_0 A^T M^{m-1} r_0) \\
&= \mathrm{span}\big(\Sigma_0 A^T r_0, (\Sigma_0 A^T A)(\Sigma_0 A^T r_0), (\Sigma_0 A^T A)^2(\Sigma_0 A^T r_0), \\
&\quad \ldots, (\Sigma_0 A^T A)^{m-1}(\Sigma_0 A^T r_0)\big) \\
&= K_{m-1}(\Sigma_0 A^T A, \Sigma_0 A^T r_0).
\end{aligned}$$

Thus, we have that the constraint in (2) is equivalent to

$$x \in x_0 + \Sigma_0 A^T K_{m-1}(M, r_0) = x_0 + K_{m-1}(\Sigma_0 A^T A, \Sigma_0 A^T r_0).$$

By simple calculations, we further note that

$$\begin{aligned}
\|(\Sigma_0 A^T)^{-1}(x - x^*)\|^2_{A\Sigma_0 A^T} &= \langle (\Sigma_0 A^T)^{-1}(x - x^*), A\Sigma_0 A^T (\Sigma_0 A^T)^{-1}(x - x^*)\rangle \\
&= \langle (\Sigma_0 A^T)^{-1}(x - x^*), A(x - x^*)\rangle \\
&= \langle \Sigma_0^{-1}(x - x^*), x - x^*\rangle \\
&= \|x - x^*\|^2_{\Sigma_0^{-1}}.
\end{aligned}$$

Therefore, (2) implies that

$$x_m = \underset{x \in x_0 + K_{m-1}(\Sigma_0 A^T A, \Sigma_0 A^T r_0)}{\arg\min} \|x - x^*\|_{\Sigma_0^{-1}},$$

which is exactly Proposition 9 in the paper. Proposition 10 can also be obtained in a similar way by using the equivalent property and the classical results of the rate of convergence for CG.

In addition to providing the above re-interpretation of the proposed Bayesian CG, another important aspect to consider is the computational feasibility related to the choice of Σ_0. In certain case, the computational costs of the matrix/vector multiplications associated with Σ_0 can be much more expensive than those associated with A or A^T. For example, the matrix A could be sparse, but the selected prior variance-covariance Σ_0 could be dense. Therefore, the choice of Σ_0 should also be factored in the evaluation of the computational complexity and the algorithms and the claim that "the cost of BayesCG is a constant factor ...are required" may need to be properly reconsidered.

References

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media. MR2244940. 977

Contributed Discussion

F.-X. Briol[*,†], F. A. DiazDelaO[‡], and P. O. Hristov[§]

We would like to congratulate the authors of Cockayne et al. (2019) on their insightful paper, and welcome this publication which we firmly believe will become a fundamental contribution to the growing field of probabilistic numerical methods and in particular the sub-field of Bayesian numerical methods. In this short piece, we first initiate a discussion on the choice of priors for solving linear systems, then propose an extension of the Bayesian conjugate gradient (BayesCG) algorithm for solving several related linear systems simultaneously.

1 Prior specification for Bayesian inference of linear systems

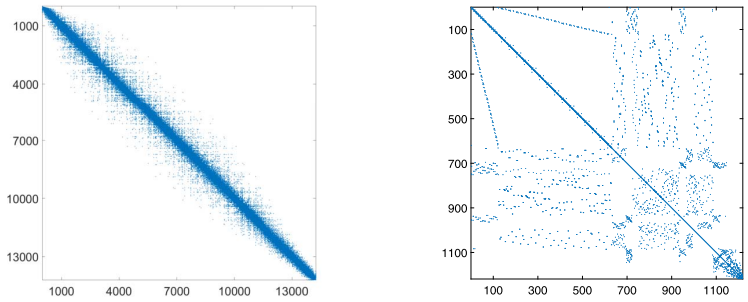
In the Bayesian paradigm, once a particular observation model is agreed upon, most of the work goes into selection of the prior. In the case of a linear system $Au = b$ and in particularly for conjugate gradient methods, our observation model consists of projections of b observed without noise. The authors of Cockayne et al. (2019) place a Gaussian prior on u, which provides advantages to placing a prior on the inverse of the matrix A (Hennig, 2015), including invariance to preconditioners. We agree that this is a significant advantage, but also think one could go much further in eliciting priors for solving linear systems, as is done for other Bayesian numerical methods.

In Bayesian quadrature, the task is to estimate $\Pi[f] = \int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x$, given evaluations of the integrand f at some locations on the domain $\mathcal{X}$. Clearly, the quantity of interest is $\Pi[f]$; yet, it is common to put a prior on f instead, which then induces a prior on $\Pi[f]$. For differential equations, the problem is to find the solution u of a system of equations $\mathcal{A}u(x) = g(x)$ (where $\mathcal{A}$ is some known integro-differential operator), given evaluations of g; and existing Bayesian methods also propose to specify a prior on g instead of the quantity of interest u. In both cases, the main motivation for placing priors on latent quantities is that this is more natural, or convenient, from a modelling point of view. At the same time, it is often possible to inspect the mathematical expression for the latent quantity, or we may at least have some additional information about it, such as smoothness or periodicity information. In such cases, encoding this information in the prior leads to algorithms with fast convergence rates and tighter credible intervals, as demonstrated for these Bayesian integration and differential equation methods (Cockayne et al., 2016; Briol et al., 2019). We believe that the same is likely to be true for the case of linear systems.

[*]Department of Statistical Science, University College London, UK, f.briol@ucl.ac.uk
[†]The Alan Turing Institute, London, UK
[‡]Department of Mathematics, University College London, UK, alex.diaz@ucl.ac.uk
[§]Institute for Risk and Uncertainty, The University of Liverpool, UK, P.Hristov2@liverpool.ac.uk

Indeed, in many applications, it is possible to know properties of A beforehand, such as information on its spectrum, conditioning or sparsity. We argue that it is more natural to encode this knowledge in a prior, and it may in fact lead to a better calibration of uncertainty. To illustrate this, consider some of the systems of differential equations used in engineering to describe fluid flow and structural response to loading, which are usually discretised into a linear system. In computational structural mechanics the operator $\mathcal{A}$ can be used to describe the *stiffness* of an assembled finite element model (FEM). Similarly, in computational fluid dynamics (CFD), $\mathcal{A}$ can represent mesh coefficient matrices. Since both of these matrices describe physical properties of the object under study, their sparsity patterns will be governed largely by the object's geometry. It is therefore common that analysts have some prior knowledge about $\mathcal{A}$, based on engineering insight and experience in solving similar systems.

Figure 1 provides examples of the form of A (i.e. discretisations of $\mathcal{A}$) for systems taking part in a typical coupled analysis of a jet engine compressor loading. The sparsity pattern shown in Figure 1(a) encodes the coefficients of an unstructured mesh for a two dimensional airfoil in a CFD simulation (Davis and Hu, 2011). The matrix in Figure 1(b) depicts the FEM stiffness matrix of the compressor disc and blades. Both geometries were meshed with two-dimensional triangular elements. In this context, the load on the compressor stage depends on the rotational speed and the force produced by its blades, which in turn depends on the rotational speed of the compressor. Employing similar chains of coupled models is not uncommon in design and analysis of complex engineering systems, and can further complicate the choice of a prior model. We believe that eliciting such priors for coupled systems is a crucial question, very much aligned with one of the ambitions of probabilistic numerics: the propagation of uncertainty through pipelines of computation (Hennig et al., 2015).



(a) (b)

Figure 1: Stiffness matrices with different degrees of sparsity and non-zero patterns. The systems described by these matrices are: (a) a laminar airfoil; (b) jet engine compressor fan.

2 A generalisation to multiple linear systems

BayesCG also provides an excellent opportunity to develop novel methodology for solving linear solvers. Suppose we have several linear systems which need to be solved either simultaneously or sequentially, such that for $j \in \{1, \ldots, J\}$, we want to solve[1]:

$$A_j x_j^* = b_j,$$

where $A_j \in \mathbb{R}^{d\times d}$, $x_j^* \in \mathbb{R}^d$ and $b_j \in \mathbb{R}^d$ for some $d \in \mathbb{N}_{>0}$. As discussed in de Roos and Hennig (2017), this is a common problem in statistics and machine learning. Take for example the issue of model selection for Gaussian processes: this includes calculating the log-marginal likelihood for several choices of covariance functions or covariance function hyperparameters, each requiring the solution of a linear system whose solutions will be closely related (atleast for similar choices of parameters). Similarly, for Bayesian inverse problems, the forward problem needs to be solved for several values of the parameters (perhaps over the path of some Markov chain Monte Carlo realisation), which will boil down to solving several closely related linear systems.

As principled Bayesians, it would be natural to construct a joint estimator on the solutions of these J linear systems, rather than estimating the solutions independently. This is particularly the case if we know anything about how the solutions of these linear systems relate to one another, in which case information available through search directions in the j^{th} system may be informative about the solution $x_{j'}^*$ for $j \neq j'$. This idea is closely related to transfer learning, which was recently advocated for problems in numerical analysis by Xi et al. (2018) (who focused on numerical integration). Although several methods exist to transfer information from one task to the other, such as recycled Krylov spaces (de Roos and Hennig, 2017), there are no existing Bayesian approach.

Interestingly, we show below that the BayesCG algorithm of Cockayne et al. (2019) may be generalised straightforwardly to this setting. All expressions below are given so as to mirror the notation of the original algorithm. The main point to make is that all of these systems can be seen as a single, larger, linear system of the form $\underline{A}\underline{x}^* = \underline{b}$ where $\underline{x} = ((x_1^*)^\top, \ldots, (x_J^*)^\top)^\top \in \mathbb{R}^{dJ}$, $\underline{b} = (b_1^\top, \ldots, b_J^\top)^\top \in \mathbb{R}^{dJ}$ and $\underline{A} \in \mathbb{R}^{dJ\times dJ}$ is of the form

$$\underline{A} = \text{BlockDiag}\,[A_1, \ldots, A_J] = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_J \end{pmatrix}.$$

We define the data obtained by $y_i = s_i^\top A x^* = s_i^\top b$ for $i \in \{1, \ldots, m\}$. We will define $\underline{S}_m \in \mathbb{R}^{dJ\times m}$ to be the matrix consisting of columns given by m search directions. The data can therefore be expressed in vector form as $\underline{y}_m = \underline{S}_m^\top \underline{b}$. Taking a Bayesian approach, we select a prior of the form $\mathcal{N}(\underline{x}, \underline{x}_0, \underline{\Sigma}_0)$, for some $\underline{x}_0 \in \mathbb{R}^{dJ}$ and $\underline{\Sigma}_0 \in \mathbb{R}^{dJ\times dJ}$. Conditioning on the data $\underline{y}_m$, we obtain a posterior of the form $\mathcal{N}(\underline{x}; \underline{x}_m, \underline{\Sigma}_m)$

[1] For simplicity of notation, we assume all systems are of the same size, but this could be generalised straightforwardly.

with $\underline{x}_m = \underline{x}_0 + \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m \underline{\Lambda}_m^{-1} \underline{S}_m^\top \underline{r}_0$, $\underline{\Sigma}_m = \underline{\Sigma}_0 - \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m \underline{\Lambda}_m^{-1} \underline{S}_m^\top \underline{A} \underline{\Sigma}_0$ where $\underline{r}_0 = \underline{b} - \underline{A}\underline{x}_0$ and $\underline{\Lambda}_m = \underline{S}_m^\top \underline{A} \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m$. The search directions which allow us to avoid the matrix inverse are $\underline{A}\underline{\Sigma}_0\underline{A}^\top$-orthogonal, and provide what we call the *multi-system BayesCG algorithm*. Let $r_m = \underline{b} - \underline{A}\underline{x}_m$, $\tilde{\underline{s}}_1 = \underline{r}_0$ and $\underline{s}_m = \tilde{\underline{s}}_m / \|\tilde{\underline{s}}_m\|_{\underline{A}\underline{\Sigma}_0\underline{A}^\top}$ for all m, then for $m > 1$, assuming that $\tilde{\underline{s}}_m \neq \underline{0} = (0, \ldots, 0)$, these directions are:

$$\tilde{\underline{s}}_m = \underline{r}_{m-1} - \langle \underline{s}_{m-1}, \underline{r}_{m-1} \rangle_{\underline{A}\underline{\Sigma}_0\underline{A}^\top} \underline{s}_{m-1}.$$

At this point, most of the equations in the two paragraph above look identical to those in the paper, but include larger vectors and matrices. We now make several remarks:

1. The search directions obtained through the multi-system BayesCG algorithm lead to some dependence across linear systems. That is, the estimator for x_j^* for some fixed j will be impacted by $A_{j'}, b_{j'}$ for some $j' \neq j$. This dependence will come from the matrix $\underline{\Sigma}_0$, the covariance matrix of our prior. This leads to a larger computational cost, due to the fact that we are now having to perform matrix-vector products of matrices of size $dJ \times dJ$, but this may be acceptable if it provides improved accuracy and uncertainty quantification.

2. Several special cases of prior matrix $\underline{\Sigma}_0$, inspired by vector-valued reproducing kernel Hilbert spaces or multi-output Gaussian processes, can be more convenient to use in practice due to their intepretability. One example are separable covariance functions, which were previously explored by Xi et al. (2018) for transfer learning in numerical integration. They take the form $\underline{\Sigma}_0 = B \otimes \Sigma_0$ where $\otimes$ denotes the Kronecker product, $B \in \mathbb{R}^{J \times J}$ and $\Sigma_0 \in \mathbb{R}^{d \times d}$. In this case, the matrix B can be seen as a covariance matrix across tasks (i.e. across linear systems), whilst Σ_0 is the covariance matrix which would otherwise be used for a single linear system. In particular, this approach would allow us to combine the algorithm with alternative transfer learning approaches, such as the Krylov subspace recycling discussed in de Roos and Hennig (2017) which can be used to select Σ_0.

3. In the case where $\underline{\Sigma}_0$ has block-diagonal form $\text{BlockDiag}\,[\Sigma_{0,1}, \ldots, \Sigma_{0,J}]$ for $\Sigma_{0,1}, \ldots, \Sigma_{0,J} \in \mathbb{R}^{d \times d}$, the multi-system Bayesian conjugate gradient method reduces to J separate instances of the BayesCG; it is therefore a strict generalisation.

4. The requirement that search directions are $\underline{A}\underline{\Sigma}_0\underline{A}^\top$-orthogonal forces us to solve the J linear systems simultaneously, obtaining one observation from each system at a given iteration of the multi-system BayesCG algorithm. This prevents us from considering the sequential case where we first solve A_1, then solve A_2 and so on. However, we envisage that alternative algorithms could be developed for this case, and could help provide informative priors in a sequential manner.

References

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. (2019). "Probabilistic integration: A role in statistical computation? (with discussion)." *Statistical Science*, 34(1): 1–22. MR3938958. doi: https://doi.org/10.1214/18-STS660. 980

Cockayne, J., Oates, C. J., Ipsen, I. C. F., and Girolami, M. (2019). "A Bayesian conjugate gradient method (with discussion)." *Bayesian Analysis*. MR3577382. doi: https://doi.org/10.1214/16-BA1017A. 980, 982

Cockayne, J., Oates, C. J., Sullivan, T., and Girolami, M. (2016). "Probabilistic meshless methods for partial differential equations and Bayesian inverse problems." *arXiv:1605.07811*. MR3577382. doi: https://doi.org/10.1214/16-BA1017A. 980

Davis, T. A. and Hu, Y. (2011). "The University of Florida Sparse Matrix Collection." *ACM Transactions on Mathematical Software*, 38(1). MR2865011. doi: https://doi.org/10.1145/2049662.2049663. 981

de Roos, F. and Hennig, P. (2017). "Krylov subspace recycling for fast iterative least-squares in machine learning." *arXiv:1706.00241*. 982, 983

Hennig, P. (2015). "Probabilistic interpretation of linear solvers." *SIAM Journal on Optimization*, 25. MR3301314. doi: https://doi.org/10.1137/140955501. 980

Hennig, P., Osborne, M. A., and Girolami, M. (2015). "Probabilistic numerics and uncertainty in computations." *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179). MR3378744. doi: https://doi.org/10.1098/rspa.2015.0142. 981

Xi, X., Briol, F.-X., and Girolami, M. (2018). "Bayesian quadrature for multiple related integrals." In *International Conference on Machine Learning, PMLR 80*, 5369–5378. 982, 983

Acknowledgments

F.-X. Briol was supported through the EPSRC grant [EP/R018413/1] and by The Alan Turing Institute's Data-Centric Engineering programme under the EPSRC grant [EP/N510129/1]. F. A. DiazDelaO acknowledges the support of The Alan Turing Institute's Data-Centric Engineering programme, where he was a visiting fellow under the EPSRC grant [EP/S001476/1].

Contributed Discussion

T. J. Sullivan[*,†]

1 Overview

In "A Bayesian conjugate gradient method", Cockayne, Oates, Ipsen, and Girolami add to the recent body of work that provides probabilistic/inferential perspectives on deterministic numerical tasks and algorithms. In the present work, the authors consider a conjugate gradient (CG) method for the solution of a finite-dimensional linear system $A\boldsymbol{x}^* = \boldsymbol{b}$ for $\boldsymbol{x}^* \in \mathbb{R}^d$, given an assuredly invertible symmetric matrix $A \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b} \in \mathbb{R}^d$, with $d \in \mathbb{N}$.

The authors derive and test an algorithm, BayesCG, that returns a sequence of normal distributions $\mathcal{N}(\boldsymbol{x}_m, \Sigma_m)$ for $m = 0, \ldots, d$, starting from a prior distribution $\mathcal{N}(\boldsymbol{x}_0, \Sigma_0)$. This sequence of normal distributions is defined using a recursive relationship similar to that defining the classical CG method, and indeed the BayesCG mean $\boldsymbol{x}_m$ coincides with the output of CG upon choosing $\Sigma_0 := A^{-1}$ — this choice is closely related to what the authors call the "natural prior covariance", $\Sigma_0 := (A^\top A)^{-1}$. The distribution $\mathcal{N}(\boldsymbol{x}_m, \Sigma_m)$ is intended to be an expression of posterior belief about the true solution $\boldsymbol{x}^*$ to the linear system under the prior belief $\mathcal{N}(\boldsymbol{x}_0, \Sigma_0)$ given the first m BayesCG search directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m$. Like CG, BayesCG terminates in at most d steps, at which point its mean $\boldsymbol{x}_d$ is the exact solution $\boldsymbol{x}^*$ and it expresses complete confidence in this belief by having $\Sigma_d = 0$. The convergence and frequentist coverage properties of the algorithm are investigated in a series of numerical experiments.

The field of probabilistic perspectives on numerical tasks has been enjoying a resurgence of interest in the last few years; see e.g. Oates and Sullivan (2019) for a recent survey of both historical and newer work. The authors' contribution is a welcome addition to the canon, showing as it does how classical methods (in this case CG; cf. the treatment of Runge–Kutta methods for ordinary differential equations by Schober et al. (2014)) can be seen as point estimators of particular instances of inferential procedures. It is particularly encouraging to see contributions coming from authors with both statistical and numerical-analytical expertise, and the possibilities for generalisation and further work are most interesting.

2 Questions and directions for generalisation

The article raises a number of natural questions and directions for generalisation and further investigation, which Cockayne et al. might use their rejoinder to address.

[*]Institute of Mathematics, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany, t.j.sullivan@fu-berlin.de

[†]Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany, sullivan@zib.de

Symmetry and generalisations It is interesting that, for the most part, BayesCG does not require that A be symmetric, and works instead with $A\Sigma_0 A^\top$. This prompts a question for the authors to consider in their rejoinder: How does BayesCG behave in the case that $A \in \mathbb{R}^{d\times d}$ is square but not invertible? Could one even show that the BayesCG posterior concentrates, within at most d iterations, to a distribution centred at the minimum-norm solution (in some norm on $\mathbb{R}^d$) of the linear system? One motivation for this question is that, if such a result could be established, then BayesCG could likely be applied to rectangular $A \in \mathbb{R}^{c\times d}$ and produce a sequence of normally-distributed approximate solutions to the minimum-norm least-squares problem, i.e. a probabilistically-motivated theory of Moore–Penrose pseudo-inverses. (Recall that the Moore–Penrose pseudo-inverse $A^\dagger \in \mathbb{R}^{d\times c}$ can be characterised as the solution operator for the minimum-norm least squares problem, i.e.,

$$A^\dagger \boldsymbol{b} = \arg\min \left\{ \|\boldsymbol{x}\|_{\mathbb{R}^d} \middle| \boldsymbol{x} \in \arg\min \left\{ \|A\boldsymbol{x}' - \boldsymbol{b}\|_{\mathbb{R}^c} \middle| \boldsymbol{x}' \in \mathbb{R}^d \right\} \right\}$$

for the usual norms on $\mathbb{R}^c$ and $\mathbb{R}^d$.)

The authors could also comment on whether they expect BayesCG to generalise easily to infinite-dimensional Hilbert spaces, analogously to CG (Fortuna, 1977, 1979; Málek and Strakoš, 2015), since experience has shown that analysis of infinite-dimensional statistical algorithms can yield powerful dimension-independent algorithms for the finite-dimensional setting (Cotter et al., 2013; Chen et al., 2018). A more esoteric direction for generalisation would be to consider fields other than $\mathbb{R}$. The generalisation of BayesCG to $\mathbb{C}$ would appear to be straightforward via a complex normal prior, but how about fields of finite characteristic?

Conditioning relations Could the authors use their rejoinder to provide some additional clarity about the relationship of BayesCG to exact conditioning of the prior normal distribution $\mathcal{N}(\boldsymbol{x}_0, \Sigma_0)$, and in particular whether BayesCG is exactly replicating the ideal conditioning step, approximating it, or doing something else entirely?

To make this question more precise, fix a weighted inner product on $\mathbb{R}^d$, e.g. the Euclidean, A-weighted, Σ_0-weighted, or $A\Sigma_0 A^\top$-weighted inner product. With respect to this inner product, let P_m be the (self-adjoint) orthogonal projection onto the Krylov space $\mathcal{K}_m$ and $P_m^\perp := I - P_m$ the (self-adjoint) orthogonal projection onto its orthogonal complement. The product measure

$$\mu_m := \mathcal{N}(P_m \boldsymbol{x}^*, 0) \otimes \mathcal{N}(P_m^\perp \boldsymbol{x}_0, P_m^\perp \Sigma_0 P_m^\perp)$$

on $\mathcal{K}_m \oplus \mathcal{K}_m^\perp$ is also a normal distribution on $\mathbb{R}^d$ that expresses complete confidence about the true solution to the linear system in the directions of the Krylov subspace and reverts to the prior in the complementary directions; μ_0 is the prior, and μ_d is a Dirac on the truth. The obvious question is, for some choice of weighted inner product, does BayesCG basically output μ_m, but in a clever way? Or is the output of BayesCG something else?

Uncertainty quantification: cost, quality, and terms of reference Cockayne et al. point out that the computational cost of BayesCG is a small multiple (a factor of three) of the cost of CG, and this would indeed be a moderate price to pay for high-quality

uncertainty quantification (UQ). However, based on the results presented in the paper, the UQ appears to be quite poorly calibrated. One could certainly try to overcome this shortcoming by improving the UQ. However, an alternative approach would be to choose a prior covariance structure Σ_0 that aggressively sparsifies and hence accelerates the linear-algebraic operations that BayesCG performs (particularly lines 7, 9, and 11 of Algorithm 1), while preserving the relatively poor UQ. Does this appear practical, in their view?

In fact, the whole question of "the UQ being well calibrated" is somewhat ill defined. Some might argue that, since $\boldsymbol{x}^*$ is deterministic, there is no scope for uncertainty in this setting. However, it certainly makes sense to ask – as the authors do in Section 6.1 – whether, when the problem setup is randomised, the *frequentist* coverage of BayesCG lines up with that implied by the randomisation of the problem. The statistic Z that Cockayne et al. introduce is intersting, and already captures several scenarios in which the UQ is well calibrated and several in which it is not; I strongly encourage Cockayne et al. to address in their rejoinder the *Bayesian* accuracy of BayesCG, e.g., to exhaustively sample the true posterior on $\boldsymbol{x}^*$ given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ and see whether this empirical distribution is well approximated by the BayesCG distribution $\mathcal{N}(\boldsymbol{x}_m, \Sigma_m)$.

As a related minor question, can BayesCG be seen as an (approximate Gaussian) *filtering* scheme for the given prior $\mathcal{N}(\boldsymbol{x}_0, \Sigma_0)$ and the filtration associated to the data stream $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$?

Precision formulation As formulated, BayesCG expresses a Gaussian belief about the solution of the linear system in terms of mean and covariance; Gaussian measures can also be expressed in terms of their mean and precision. Do the authors have a sense of whether the BayesCG algorithm be formulated as a sequence of precision updates, or does the singularity of the covariance in the Krylov directions essentially forbid this?

One motivation for seeking a precision formulation would be to render the "natural prior" $\Sigma_0 := (A^\top A)^{-1}$ of Section 4.1 more tractable, since this prior has an easily-accessible precision while accessing its covariance involves solving the original linear problem.

As the authors note, their "natural prior" is closely related to one introduced by Owhadi (2015) and applied in Cockayne et al. (2017). It seems that working with images of Gaussian white noise, such as this "natural prior", is presently producing considerable analytical and computational strides forward (Owhadi, 2017; Chen et al., 2018), and so this seems to be a topic worth further attention in the statistical community as a whole.

3 Minor comments

Rank and trace estimates Proposition 3, which states that $\operatorname{tr}(\Sigma_m \Sigma_0^{-1}) = d - m$, seems to miss the point slightly. It would be good to have a companion results to the effect that $\operatorname{rank} \Sigma_m = d - m$, and a more quantitative result for the trace such as $\operatorname{tr} \Sigma_m \leq C_{\Sigma_0}(d - m)$ for some constant $C_{\Sigma_0} \geq 0$ depending only on Σ_0.

Posterior and Krylov spaces It seems natural to ask whether the posterior nullspace $\ker \Sigma_m$ coincides with the Krylov space $\mathcal{K}_m$. Put another way, is the posterior column space the same as the orthogonal complement of the Krylov space, in the Euclidean or A-weighted inner product?

Square roots Is the square root $M^{1/2}$ introduced just before Proposition 5 required to be unique? Does it even matter whether a (unique) symmetric positive semidefinite square root or a Cholesky factor is chosen? This is related to the above discussion of symmetry and generalisations.

Interpretation of termination criteria In Section 5, the authors refer to "probabilistic termination criteria". As a matter of semantics, the termination criterion that they propose is in fact deterministic, albeit based on a probabilistic interpretation of the algorithmic quantities.

References

Chen, V., Dunlop, M. M., Papaspiliopoulos, O., and Stuart, A. M. (2018). "Dimension-Robust MCMC in Bayesian Inverse Problems." https://arxiv.org/abs/1803.03344. 986, 987

Cockayne, J., Oates, C. J., Sullivan, T. J., and Girolami, M. (2017). "Probabilistic numerical methods for PDE-constrained Bayesian inverse problems." In Verdoolaege, G. (ed.), *Proceedings of the 36th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 1853 of *AIP Conference Proceedings*, 060001-1-060001-8. MR1962611. doi: https://doi.org/10.1063/1.4985359. 987

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). "MCMC methods for functions: modifying old algorithms to make them faster." *Statistical Science*, 28(3): 424–446. MR3135540. doi: https://doi.org/10.1214/13-STS421. 986

Fortuna, Z. (1977). "Superlinear convergence of conjugate gradient method in Hilbert space." In *Theory of nonlinear operators (Proc. Fourth Internat. Summer School, Acad. Sci., Berlin, 1975)*, Abh. Akad. Wiss. DDR Abt. Math.-Natur.-Tech., 1977, 1, 313–318. Akademie-Verlag, Berlin. MR0487705. 986

Fortuna, Z. (1979). "Some convergence properties of the conjugate gradient method in Hilbert space." *SIAM Journal on Numerical Analysis*, 16(3): 380–384. MR0530475. doi: https://doi.org/10.1137/0716031. 986

Málek, J. and Strakoš, Z. (2015). *Preconditioning and the conjugate gradient method in the context of solving PDEs*, volume 1 of *SIAM Spotlights*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. Chapter 5. MR3307335. doi: https://doi.org/10.1137/1.9781611973846.ch5. 986

Oates, C. J. and Sullivan, T. J. (2019). "A modern retrospective on probabilistic numerics." *Statistics and Computing*. To appear. https://arxiv.org/abs/1901.04457. 985

Owhadi, H. (2015). "Bayesian numerical homogenization." *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal*, 13(3): 812–828. MR3369060. doi: https://doi.org/10.1137/140974596. 987

Owhadi, H. (2017). "Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games." *SIAM Review*, 59(1): 99–149. MR3605827. doi: https://doi.org/10.1137/15M1013894. 987

Schober, M., Duvenaud, D. K., and Hennig, P. (2014). "Probabilistic ODE solvers with Runge–Kutta means." In *Advances in Neural Information Processing Systems 27*. https://papers.nips.cc/paper/5451-probabilistic-ode-solvers-with-runge-kutta-means. 985

Acknowledgments

TJS is supported by the Freie Universität Berlin within the Excellence Strategy of the German Research Foundation (DFG), including the ECMath/MATH+ transition project CH-15 and project TrU-2 of the Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID 390685689).

Contributed Discussion

Daniela Calvetti*

The authors address the task of solving large linear systems in a Bayesian framework, with particular emphasis on the conjugate gradient method and some generalizations. Given the growing interest in using the Bayesian framework to solve large scale linear, or linearized, inverse problems, where iterative solvers are the methods of choice, the authors' effort to bridge numerical linear algebra and Bayesian inference is very timely contribution. Systematic efforts to tear down the cultural wall between these two communities have been going on for at least a decade and a half. The desire to provide a friendly introduction to Bayesian scientific computing that would bring the two communities closer was the main reason behind Calvetti and Somersalo (2007), but unfortunately many of the contributions at the interface of these research areas seem to be falling still between the cracks. This is confirmed by the paper's paucity of references to the literature on Bayesian interpretation of preconditioners and on the construction of preconditioners from a Bayesian viewpoint, as well as to other numerical contributions similar to those being proposed. In the following, I will provide a brief summary of some results quite close to those in the paper that have been published in the literature.

1 Preconditioners from a Bayesian Perspective

Over the last 15 years, the analysis of statistically inspired and motivated preconditioners has led to families of computationally efficient algorithms for posterior modes, even when the dimensionality of the data and the unknown differ, thus breaking away from the need to have a square forward matrix. Below is an overview of literature on Bayesian iterative solvers, including but not limited to preconditioners, that fills the gap in the history of the topic, so as to make it easier to assess and compare the different approaches to selecting preconditioners.

In numerical analysis, linear preconditioners are usually selected so as to reduce the number of iterations needed for approximating accurately the solution, thus targeting transformations that cluster the spectrum and make the linear operator close to the identity. The genesis of preconditioners for inverse problems followed a different route. Once the similarities between Tikhonov regularized solution and the solution computed with an iterative linear solver, e.g., the Conjugate Gradient for Least Squares (CGLS) or the Generalized Minimum RESidual (GMRES) methods, equipped with a suitable stopping rule were observed, see, e.g., Calvetti and Somersalo (2007), preconditioners coming from Tikhonov regularization operators started to gain popularity. As the Bayesian framework for the solution of inverse problems started gaining acceptance in the inverse problems community, and with it came the need to design efficient numerical

*Department of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University, Cleveland, OH, USA, dxc57@case.edu

methods for the solution of problems that did not admit analytic solutions, preconditioners started to be looked at in a Bayesian way. The contributions of Calvetti and Somersalo (2005a); Calvetti (2007) are to connect left and, especially, right preconditioners to the covariance matrices of the noise and prior, in the simplified framework where the noise is additive Gaussian and the prior is Gaussian. In the general Bayesian setting, the prior expresses what is believed about the unknown before taking the data into consideration. Therefore, setting the prior should be independent on how data are collected, or if data are collected at all. This is not the case in the current paper which, on the contrary, proposes to set the covariance of the Gaussian prior starting from a classical preconditioner of the forward linear operator. If the spoon is the proof of the pudding, the draws are the proof of the prior: I would imagine that draws from a prior built on a classical preconditioner will be more representative of the characteristics of the forward model than of the expected traits of the solution. Over the course of the last decade and a half, preconditioners have been used to express many properties believed to be had by the solutions, ranging from its behavior at the boundary of the domain (Calvetti and Somersalo, 2005b), to its sparsity (Calvetti and Somersalo, 2008).

2 Statistically Inspired Stopping Rules and Statistically Interpreted Error

Stopping rules are essential when using preconditioned Krylov subspace iterative methods to solve linear inverse problems. Classical stopping rules are based on the discrepancy principle, Generalized Cross Validation (GCV), and L-curve. The stopping rule for CGLS proposed in Calvetti et al. (2017), which is statistically motivated, seems rather close to the criterion advocated in the paper. It would be very interesting to see whether and how the two are related.

3 EIT and Preconditioned CGLS

Bayes-Krylov preconditioned iterative methods for posterior mode computations have been used in different contexts, including in applications to electrical impedance tomography (EIT). EIT is one of the most popular examples of PDE based inverse problems on which different algorithms are tested, and there is a rich literature on Bayesian methods from decades ago, see, e.g. Nicholls and Fox (1997, 1998); Kolehmainen et al. (1997); Somersalo et al. (1997) for some of the earliest reports on Bayesian EIT solutions, with more systematic summaries in Kaipio et al. (2000); Kaipio and Somersalo (2006). The claim made by the authors that (Dunlop and Stuart, 2015) is where the Bayesian approach has been "formalized", gives a somewhat flawed view of the research in this field, in particular since the example in the present paper considers EIT in the discretized outset with a FEM based forward map. In that context a reference to inverse problems in the Hilbert space setting does not add anything to what was already thoroughly understood in the previous works on EIT and Bayesian analysis. What seems to be more relevant on the other hand is the use of priorconditioned CGLS in connection with the EIT problem in Calvetti et al. (2012).

References

Calvetti, D. (2007). "Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective." *Journal of Computational and Applied Mathematics* 198: 378–395. MR2260675. doi: https://doi.org/10.1016/j.cam.2005.10.038. 991

Calvetti, D., McGivney, D., and Somersalo, E. (2012) "Left and right preconditioning for electrical impedance tomography with structural information." *Inverse Problems* 28: 055015. MR2923200. doi: https://doi.org/10.1088/0266-5611/28/5/055015. 991

Calvetti, D. and Somersalo, E. (2005a). "Priorconditioners for linear systems." *Inverse Problems* 21: 1397. MR2158117. doi: https://doi.org/10.1088/0266-5611/21/4/014. 991

Calvetti, D. and Somersalo, E. (2005b). "Statistical elimination of boundary artefacts in image deblurring." *Inverse Problems* 21: 1697–1714. MR2173418. doi: https://doi.org/10.1088/0266-5611/21/5/012. 991

Calvetti, D. and Somersalo, E. (2007). *An Introduction to Bayesian Scientific Computing*. Springer-Verlag, New York, NY. MR2351679. 990

Calvetti, D. and Somersalo, E. (2008). "Hypermodels in the Bayesian imaging framework." *Inverse Problems* 24: 034013. MR2421950. doi: https://doi.org/10.1088/0266-5611/24/3/034013. 991

Calvetti, D., Pitolli, F., Preziosi, J., Somersalo, E., and Vantaggi, B. (2017). "Priorconditioned CGLS-based quasi-MAP estimate, statistical stopping rule, and ranking of priors." *SIAM Journal of Scientific Computing*, 39: S477–S500. MR3716568. doi: https://doi.org/10.1137/16M108272X. 991

Dunlop, M. and Stuart, A. (2015). "The Bayesian formulation of EIT: analysis and algorithms." *Inverse Problems and Imaging* 10: 1007–1036. MR3610749. doi: https://doi.org/10.3934/ipi.2016030. 991

Kaipio, J. P., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. (2000). "Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography." *Inverse Problems* 16: 1487. MR1800606. doi: https://doi.org/10.1088/0266-5611/16/5/321. 991

Kaipio, J. P. and Somersalo, E. (2006). *Statistical and Computational Inverse Problems*. Springer-Verlag, New York, NY. MR2102218. 991

Kolehmainen, V., Somersalo, E., Vauhkonen, P. J., Vauhkonen, M., and Kaipio, J. P. (1997). "A Bayesian approach and total variation priors in 3D electrical impedance tomography." *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 2: 1028–1031. IEEE. 991

Nicholls, G. K. and Fox, C. (1997). "Sampling conductivity images via MCMC." *The art and science of Bayesian image analysis*, 91–100. 991

Nicholls, G. K. and Fox, C. (1998). "Prior modeling and posterior sampling in impedance imaging." *Computational, Experimental, and Numerical Methods for Solving Ill-Posed Inverse Imaging Problems: Medical and Nonmedical Applications*, 3459: 116–127. International Society for Optics and Photonics. 991

Somersalo, E., Kaipio, J. P., Vauhkonen, M. J., Baroudi, D., and Järvenpää, S. (1997). "Impedance imaging and Markov chain Monte Carlo methods." *Computational, Experimental, and Numerical Methods for Solving Ill-Posed Inverse Imaging Problems: Medical and Nonmedical Applications* 3171:175–186. International Society for Optics and Photonics. 991

Contributed Discussion

Simone Rossi*, Cristian Rusu†, Lorenzo A. Rosasco‡, and Maurizio Filippone*

We would like to congratulate with the Authors for this interesting development of probabilistic numerical methods applied to the ubiquitous problem of solving linear systems. We structured this discussion around two main points, namely the use of Bayesian Conjugate Gradient (BCG) for Gaussian processes (GPs), and the possibility to accelerate the solution of linear systems thanks to parallelization of BCG.

Bayesian Conjugate Gradient for Gaussian Processes

Consider a regression task where X and $\mathbf{y}$ denote the set of input points and the set of targets, respectively, and assume a GP with an RBF kernel to model the mapping between X and $\mathbf{y}$ (Rasmussen and Williams, 2006). We are going to assume that GP hyper-parameters are optimized through standard marginal likelihood optimization, although it is possible to reformulate the problem of optimizing GP hyper-parameters in terms of linear systems (Filippone and Engler, 2015) where BCG could be applied. We are going to focus on the predictive distribution and the additional uncertainty stemming from the use of BCG. The GP predictive distribution is $p(\tilde{\mathbf{y}}|X,\mathbf{y},\tilde{X},\boldsymbol{\alpha}) = \mathcal{N}\left(K_{\tilde{X}X}\boldsymbol{\alpha}, \Sigma_{\tilde{\mathbf{y}}}\right)$, where $\boldsymbol{\alpha}$ is the solution of the linear system $(K+\lambda I)\boldsymbol{\alpha} = \mathbf{y}$. As BCG provides a distribution over the solutions for $\boldsymbol{\alpha}$ (i.e. $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\alpha}_m, \Sigma_m)$), we can integrate out $p(\boldsymbol{\alpha})$ obtaining

$$p(\tilde{\mathbf{y}}|X,\mathbf{y},\tilde{X}) = \mathcal{N}\left(K_{\tilde{X}X}\boldsymbol{\alpha}_m, \Sigma_{\tilde{\mathbf{y}}} + K_{\tilde{X}X}\Sigma_m K_{\tilde{X}X}^{\mathrm{T}}\right) ,$$

The topic of preconditioning for solving linear systems involving kernel matrices is an active area of research (Cutajar et al., 2016; Rudi et al., 2017), so we can leverage this in BCG given the connections established in the paper between Σ_0 and preconditioners.

We report the test MNLL and the test RMSE (20% of held-out data) as a function of BCG iterations for two datasets. Figure 1 shows that better preconditioners yield faster convergence. Figure 2 shows the error metrics as a function of time for GPs using BCG and sparse GPs (Matthews et al., 2017). There are configurations where BCG allows to reach better performance for a given computational budget, so this is an interesting possible application of this method.

Bayesian Model Averaging for multiple BCG solutions

One of the advantages that we see in the Bayesian formulation of conjugate gradient, is the possibility to speedup convergence through parallelization. To test this, we solve

*Department of Data Science, EURECOM, Sophia Antipolis, France, simone.rossi@eurecom.fr
†University of Genoa, LCLS – IIT, cristian.rusu@iit.it
‡University of Genoa, LCLS – IIT & MIT, lrosasco@mit.edu
*Department of Data Science, EURECOM, Sophia Antipolis, France, maurizio.filippone@eurecom.fr

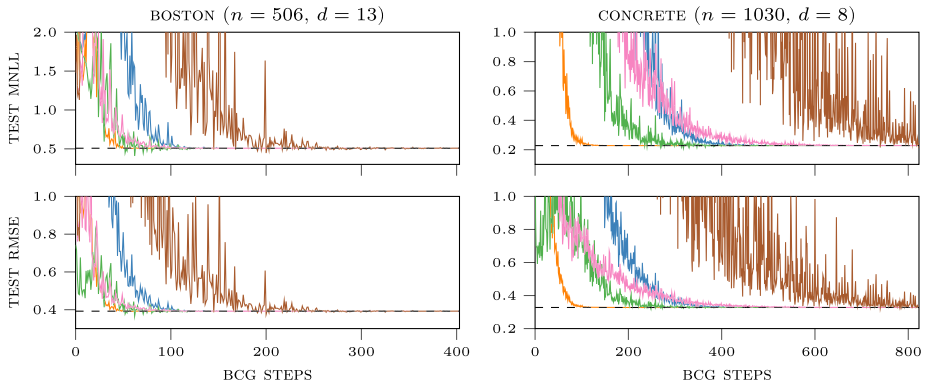


Figure 1: Comparison test MNLL and test RMSE for different priors (e.g. preconditioners) of BCG on two regression datasets. As preconditioners we consider Nyström (Williams and Seeger, 2000) with $\sqrt{n}$ (●) and $4\sqrt{n}$ (●) inducing points, PITC (Candela and Rasmussen, 2005) (●), and RANDOM SVD (Halko et al., 2011) (●). Experiment repeated 25 times.

multiple linear systems with BCG using different priors (possibly concurrently) and aggregate the solutions by means of Bayesian model averaging. Formally, let $\Sigma_0^{(i)}$ denote one of such multiple priors (corresponding to preconditioners) and let $p(\mathbf{x}_m|\Sigma_0^{(i)})$ be the solution at iteration m corresponding to the choice of the ith prior. Assuming a prior on the set of all $\Sigma_0^{(i)}$, the marginalization yields the mixture $p(\mathbf{x}_m) = \sum_i p(\mathbf{x}_m|\Sigma_0^{(i)})p(\Sigma_0^{(i)})$ We project this back to a Gaussian distribution on $p(\mathbf{x}_m)$ by moment matching. We assume a uniform prior for $p(\Sigma_0^{(i)})$, but we could think of relaxing this by setting a prior proportional to the complexity of (or time spent in) inverting the preconditioner.

In Figure 1, the line (●) shows this result. Using the same setup as before, we infer the posterior distribution of a GP using a Bayesian averaging of 16 independent

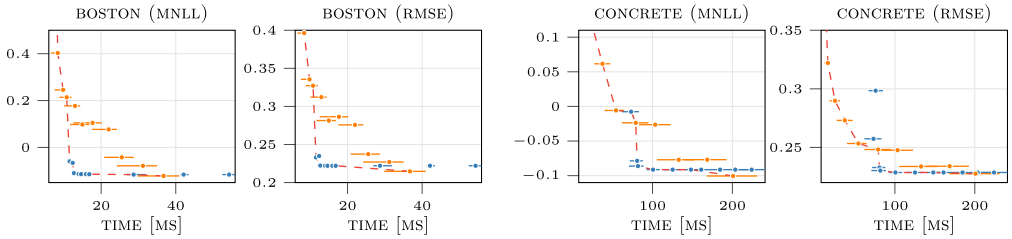


Figure 2: Analysis of the Pareto front (- -) of inference time vs error metric for full GP with BCG (● – Nyström preconditioner is assumed to be precomputed) and sparse GP (●). Points corresponds to different amount of BCG iterations and number of inducing points (with their kernel parameters optimized). Experiment repeated 500 times.

solutions with $\sqrt{n}$ random centers for the Nyström preconditioner (the comparison is with ●). This suggests that it is possible to benefit from combining multiple intermediate solutions of BCG, and this is rather intuitive in the context of Bayesian model averaging.

References

Candela, J. Q. and Rasmussen, C. E. (2005). "A Unifying View of Sparse Approximate Gaussian Process Regression." *Journal of Machine Learning Research*, 6: 1939–1959. MR2249877. 995

Cutajar, K., Osborne, M., Cunningham, J., and Filippone, M. (2016). "Preconditioning Kernel Matrices." In Balcan, M.-F. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 2529–2538. JMLR.org. 994

Filippone, M. and Engler, R. (2015). "Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE)." In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, July 6–11, 2015*. 994

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions." *SIAM Rev.*, 53(2): 217–288. MR2806637. doi: https://doi.org/10.1137/090771806. 995

Matthews, A. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). "GPflow: A Gaussian process library using TensorFlow." *Journal of Machine Learning Research*, 18(40): 1–6. MR3646635. 994

Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press. MR2514435. 994

Rudi, A., Carratino, L., and Rosasco, L. (2017). "FALKON: An Optimal Large Scale Kernel Method." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, 3888–3898. Curran Associates, Inc. 994

Williams, C. K. I. and Seeger, M. (2000). "Using the Nyström Method to Speed Up Kernel Machines." In Leen, T. K., Dietterich, T. G., Tresp, V., Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *NIPS*, 682–688. MIT Press. 995

Rejoinder

Jon Cockayne*, Chris J. Oates†, Ilse C.F. Ipsen‡, and Mark Girolami§

The authors are grateful to each of the discussants of our paper, "A Bayesian Conjugate Gradient Method". These provide valuable insight beyond our areas of expertise and have served to highlight aspects of the method that were not discussed in detail in the original manuscript. Below, please find our author responses to the points that have been raised. These are structured as follows: In Section 1 we address points relating to theoretical analysis of the method and in Section 2 the computational cost of the method is discussed. Then, in Section 3 the prior construction is discussed and implications for uncertainty quantification are explored in Section 4. Some extensions of the method are raised in Section 5 and other related matters are contained in Section 6. Finally, we summarise the rejoinder in Section 7.

1 Theoretical Results

Several discussants commented on the theoretical results presented in Cockayne et al. (2019a). In particular, the invited discussion from Xudong Li and Ethan Fang demonstrated an alternative method for deriving many of the theoretical results therein, while the contributed discussion from Tim Sullivan suggested several other theoretical developments that would be of interest. Relatedly, the contributed discussion from Daniela Calvetti commented on the stopping criteria we described in Section 5 of Cockayne et al. (2019a).

1.1 Properties of BayesCG

Li and Fang noted in their discussion that the posterior mean from the Bayesian conjugate gradient method (BayesCG) is equivalent to the iterate produced by applying the conjugate gradient method (CG) to the reformulated linear system $M\tilde{\boldsymbol{x}}^* = \boldsymbol{b}$, where $M = A\Sigma_0 A^\top$ and $\tilde{\boldsymbol{x}}^* = (\Sigma_0 A^\top)^{-1}\boldsymbol{x}^*$ (i.e. by using $\Sigma_0 A^\top$ as a right-preconditioner). This perspective dramatically simplifies the proofs that appeared in our paper, in that we may appeal to classical results for preconditioned CG in order to prove the results concerning convergence of $\boldsymbol{x}_m$ and its optimality properties (Propositions 9 and 10 from Cockayne et al. (2019a)). We would like to thank Li and Fang for pointing out this approach; it is always illuminating to learn about alternative representations and proofs of a problem, and it would have significantly simplified the writing of the paper had we observed this ourselves.

*The Alan Turing Institute, British Library, 96 Euston Road, London, NW1 2DB, jcockayne@turing.ac.uk, url: http://www.joncockayne.com

†School of Mathematics and Statistics, Herschel Building, Newcastle University, NE1 7RU, chris.oates@ncl.ac.uk

‡Department of Mathematics, North Carolina State University, Raleigh, NC, 27695-8205, ipsen@ncsu.edu

§Engineering Department, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, mag92@eng.cam.ac.uk

Sullivan made several interesting comments on the theoretical results reported. Specifically, he asked:

1. About further properties of Σ_m, such as its rank and bounds for its trace.
2. Whether the posterior distribution from BayesCG can be obtained by an orthogonal projection of the prior.
3. The relationship between the nullspace of Σ_m and the Krylov space from Proposition 9.

To directly address these questions, we include four additional theoretical results below, proofs of which can be found in Appendix A:

Proposition 1 (Rank of Σ_m)**.** *Suppose that $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m$ are linearly independent and recall from Cockayne et al. (2019a) that Σ_0 is assumed to be positive-definite. Then it holds that* $\operatorname{rank}(\Sigma_m) = d - m$.

Proposition 2 (Trace of Σ_m)**.** *It holds that* $\operatorname{trace}(\Sigma_m) \leq \operatorname{trace}(\Sigma_0)^{\frac{1}{2}}(d-m)^{\frac{1}{2}}$.

For the next proposition we must introduce some notation. For a Krylov space $K_m(M, \boldsymbol{v})$ with $M \in \mathbb{R}^{d\times d}$ and $\boldsymbol{v} \in \mathbb{R}^d$, and a matrix $S \in \mathbb{R}^{d\times d}$ define the affine space $SK_m(M, \boldsymbol{v})$ as

$$SK_m(M, \boldsymbol{v}) = \operatorname{span}(S\boldsymbol{v}, SM\boldsymbol{v}, \ldots, SM^{m-1}\boldsymbol{v}).$$

Proposition 3 (Posterior as Projection)**.** *Let $\mathcal{K}_m = \Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$. Let P denote an orthogonal projector onto $\mathcal{K}_m$ with respect to the inner product induced by Σ_0^{-1} and let $P^\perp$ denote an orthogonal projector onto $\mathcal{K}_m^\perp$ with-respect-to the same inner product. Let $\mu_0 \in \mathcal{P}(\mathbb{R}^d)$ denote the prior and $\mu_m \in \mathcal{P}(\mathbb{R}^d)$ denote the posterior from m iterations of BayesCG. Then we have that $P_\# \mu_m = \delta(P(\boldsymbol{x}^*))$ (i.e. a Dirac on $P(\boldsymbol{x}^*)$) and $P^\perp_\# \mu_m = P^\perp_\# \mu_0$.*

Note that Proposition 1 can also be obtained as a direct corollary of Proposition 3 and the rank-nullity theorem, since $\operatorname{rank}(\Sigma_m) = \operatorname{rank}(P^\perp) = \dim(\mathbb{R}^d) - \operatorname{nullity}(P^\perp) = d - m$.

Proposition 4 (Null-Space of Σ_m)**.** *The null space of Σ_m is $A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$.*

This last result is somewhat counterintuitive as the null-space is not $\Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$. However, we note that the null-space of Σ_m is orthogonal to $\Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$ with-respect to the inner product induced by Σ_0^{-1}.

1.2 Stopping Criteria

The names of Thomas Bayes and Alexei Krylov were previously brought together in Calvetti et al. (2018), in which the authors developed efficient iterative solvers for deployment in a more traditional Bayesian parameter inference context (i.e. inverse problems involving noisy data). The authors are grateful to Daniela Calvetti for participating in

this discussion and for highlighting the potential for diverse applications of numerical linear algebra within statistics.

In her discussion, Calvetti asked whether the proposed stopping rule from Section 5 of Cockayne et al. (2019a) is related to the stopping rule presented in Calvetti et al. (2017). This rule, termed the $\max_{\chi^2}$ stopping criterion, is derived in the context of noisy linear Bayesian inversion problems, in which we wish to solve $A\boldsymbol{\theta} = \boldsymbol{y} + \boldsymbol{\xi}$ for $\boldsymbol{\theta}$. Here $\boldsymbol{\xi} \in \mathbb{R}^n$ is a random variable representing observational noise, $\boldsymbol{y} \in \mathbb{R}^n$ is a datum, $\boldsymbol{\theta} \in \mathbb{R}^d$ is a parameter of interest and $A \in \mathbb{R}^{n\times d}$ is a parameter-to-observation map. Importantly the context for that paper is one in which $d > n$, so that the problem is underdetermined. This setting is different to that in Cockayne et al. (2019a), in that our paper assumes A to be invertible and the observations to be noiseless; nevertheless it is useful to comment on those aspects of the work that are related.

In the notation of our paper, the criterion proposed in Calvetti et al. (2017) is as follows: Let $Q(t,k) = \text{Prob}(T > 2t)$ for $T \sim \chi^2_{2k}$, $t_m = \|\boldsymbol{r}_m\|_2^2$ and $p_k = Q(\frac{t_m}{2}, \frac{d-m}{2})$. The authors then reason that, once sufficient iterations have been performed so that the signal from the data has been extracted and only noise remains, it should hold that $t_m \sim \chi^2_{d-m}$. The proposed $\max_{\chi^2}$ stopping criterion states that the iteration should stop at the iteration m^*, where

$$m^* \in \operatorname*{arg\,max}_{1\leq m\leq d} p_m.$$

In practice of course p_m is not computed for $m = 1, \ldots, d$, but rather the algorithm tracks p_m and terminates when it starts to decrease. Note that the rationale for the distribution of p_m assumes that the algorithm is applied in the setting of noisy observations.

The termination criterion studied[1] in Cockayne et al. (2019a, Section 5) is to terminate the algorithm when

$$(d-m)\nu_m < \epsilon$$

for some $\epsilon > 0$, where $\nu_m = \frac{1}{m}\|S_m^\top \boldsymbol{r}_0\|_2^2$. To compare the two, using the conjugacy properties of the algorithm it is straightforward to show by following the arguments in Cockayne et al. (2019b, Section S2) that the BayesCG search directions satisfy

$$s_m^\top \boldsymbol{r}_0 = \boldsymbol{s}_m^\top \boldsymbol{r}_{m-1} = \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}{\|\tilde{\boldsymbol{s}}_m\|_{A\Sigma_0 A^\top}}.$$

Thus it holds that

$$\nu_m = \frac{1}{m}\sum_{i=1}^{m} \frac{\|\boldsymbol{r}_{m-1}\|_2^2}{\|\tilde{\boldsymbol{s}}_m\|_{A\Sigma_0 A^\top}}.$$

If we suppose, in a similar vein to Calvetti et al. (2017), that $\boldsymbol{r}_m \sim \mathcal{N}(0, \|\tilde{\boldsymbol{s}}_m\|_Q^2 I)$ (note the scaling on the variance) then we would have that $m\nu_m \sim \chi^2_m$, so it is conceivable that a similar statistical test could be applied in the present setting. On the other hand,

[1] It is important to note that the criterion we studied in the paper is not actually being recommended as it was found not to be a good proxy for low error having been achieved.

the rationale for such a supposition is less clear in the present noise-free regime. Nevertheless, it would be interesting to discuss further whether a probabilistically-motivated termination criterion for BayesCG can be constructed by following similar arguments to Calvetti et al. (2017) in the setting of Cockayne et al. (2019a).

2 Computational Cost

Li and Fang commented that in the case of a sparse matrix A, our statements about the computational cost of BayesCG in Cockayne et al. (2019a) — namely, that the algorithm has a computational complexity only a constant factor higher than CG — may need to be reconsidered. We agree, and in hindsight should have been more specific that our comments referred to the cost of CG in the case of dense matrices A and Σ_0. Since CG is often applied to sparse matrices, a more careful consideration of the cost in this setting is due.

When A is sparse the cost of CG is only $\mathcal{O}(\mathrm{nnz}(A))$ per-iteration where $\mathrm{nnz}(A)$ is the number of nonzero entries of A. BayesCG requires two applications of A and one application of Σ_0 per-iteration, where the sparsity pattern depends of Σ_0 on the complexity of the prior. Of course in the case when $\Sigma_0 = I$, or any other diagonal matrix, the cost of an application of Σ_0^{-1} is much lower than the cost of an application of A; similarly if Σ_0 is dense the cost would be higher, as noted in Li and Fang's discussion.

However the choice of Σ_0 that we advocate in Cockayne et al. (2019a, Section 4.1) is to use a prior based on a preconditioner of A. It is difficult to make general statements about how high a cost this will incur, as preconditioners vary in sparsity from diagonal through to dense. For the preconditioner we have adopted in Cockayne et al. (2019a, Section 6) — based on an incomplete Cholesky factorisation of A with zero fill-in — assuming the incomplete factor is precomputed, the cost incurred from an application of Σ_0 is the cost of one pass of sparse forward substitution and one pass of sparse backward substitution. Since the sparsity pattern of the factors L is, in this case, chosen to be the same as the lower-triangular part of A, the cost incurred is of the same computational order as *two* applications of A. Thus, while the cost incurred in BayesCG is a more subtle matter than portrayed in the paper, for this particular preconditioner prior the cost is only a constant factor higher than that of BayesCG even for sparse A.

An alternative perspective is to think of BayesCG with a prior covariance based on a preconditioner as a constant factor more expensive than *preconditioned* CG. Preconditioned CG requires one application of A and one of P^{-1} per-iteration, whereas BayesCG with a preconditioner prior requires two applications of A and of P^{-1}. This reasoning holds regardless of the cost of applying A and the preconditioner.

3 Prior Construction

An important aspect of BayesCG that we highlighted in Section 4 of Cockayne et al. (2019a) is the choice of prior, particularly of the matrix Σ_0. Several of the discussion articles also highlighted this.

3.1 Matrix-vs-Solution Prior

In the thoughtful and illuminating invited discussion from Philipp Hennig, the approach taken in the present paper, in which a prior is placed on $\boldsymbol{x}$, is compared and contrasted with Hennig (2015) and Bartels and Hennig (2016), in which a prior was instead placed on A^{-1}. We agree that there are certainly arguments for both endowing A^{-1} (or A) and $\boldsymbol{x}$ with a prior distribution. Among these is that, as Hennig comments, the solution of linear systems based on A is only *one* possible use of the matrix A, and other uses such as calculation of determinants or spectra cannot be realised with a prior on $\boldsymbol{x}$. We have focussed on the problem of solving $A\boldsymbol{x} = \boldsymbol{b}$, which makes the inference framework presented in Cockayne et al. (2019a) less widely applicable than in Hennig (2015).

In the context of solution of a linear system we concur that incorporating more information about A into the prior is required. To some extent our use of priors based on preconditioners accomplishes this, but that approach is still somewhat of a black-box unless the preconditioner is constructed carefully. A more structured prior on $\boldsymbol{x}$ using information about A such as its sparsity pattern and spectrum might yield better calibrated UQ, while also achieving faster convergence rates. As noted in the comment, our joint work in Bartels et al. (2019) goes some way to bridging this gap, so to some extent the benefits of constructing priors on A^{-1} can be realised in the setting when the prior is placed on $\boldsymbol{x}$.

The discussion contributed by Francois-Xavier Briol, Alejandra DiazDelaO and Peter Hristov also advocated for placing a prior on A or on A^{-1}. One of their arguments for this is that it is common in related infinite-dimensional problems to place a prior on a latent quantity rather than the quantity of interest, and that this typically accelerates computation in those settings. However we would argue that the appropriate analogy with the finite-dimensional setting *is* to place a prior on $\boldsymbol{x}$, and not to place the prior on A; in solution of a partial differential equation (PDE), for example, no existing work on probabilistic numerical methods has advocated for placing a prior on the inverse of the differential operator. Of course this does not mean that it should not be attempted, however we would suggest that the accelerated convergence the authors refer to could therefore be achieved with a prior on $\boldsymbol{x}$ rather than on A.

They also argue that in practice much is often known of the properties of A, owing to the fact that it arises from a discretisation of a PDE. For example, its sparsity properties or spectrum may be known, and this information could be encoded into the prior. We agree that strong prior information on such properties of A is more challenging to encode directly in $\boldsymbol{x}$. However we note that, since a distribution on A^{-1} induces a marginal distribution on $\boldsymbol{x}$ (Bartels et al., 2019), we expect that construction of matrix-valued priors that encode the features of the problem would be directly applicable to BayesCG.

3.2 On Preconditioner Priors

In the context of selecting a preconditioner for use in the prior, we would like to thank Daniela Calvetti for providing in her discussion a summary of existing literature on Bayesian perspectives on preconditioning. Indeed, the use of the prior covariance as a

right-preconditioner for the linear system described in Calvetti and Somersalo (2005) resembles the alternative formulation of the linear system from Li and Fang's invited discussion that yields iterates identical to the sequence of posterior means from BayesCG.

One distinction that is of some significance is that the works cited in the discussion focus on construction of a prior over the solution to the *inverse problem*, whereas here we are attempting to construct priors over the solution to the *forward problem*. Mathematically speaking the differences are limited; we are simply interpreting the forward problem as a noiseless inverse problem, as described in Cockayne et al. (2019c). However the considerations that must be made when constructing the prior differ. Good priors for the parameter in the inverse problem are not guaranteed to be good priors for solving the linear system that arises from the forward problem. Thus we would agree with her comment that a preconditioner for the forward problem is likely to yield draws more representative of the characteristics of the forward problem, but also note that this is precisely our goal in the setting of the paper!

Also in this direction, Calvetti remarked that choosing a prior based on a preconditioner departs from the Bayesian paradigm, in the sense that the choice of prior is then dependent on how the data are collected. Our view is that this is not the case for BayesCG, since the data $y_i := \boldsymbol{s}_i^\top \boldsymbol{b}$ are not used to construct a preconditioner for A. In fact, we would go further and argue that preconditioner priors are especially natural in an inverse problem context: To be concrete, suppose data $\boldsymbol{y} \in \mathbb{R}^n$ arise as $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\xi}$ where $\boldsymbol{x}, \boldsymbol{\xi} \in \mathbb{R}^n$, and that $\boldsymbol{x}$ is related to a parameter θ of interest via a linear system $A_\theta \boldsymbol{x} = \boldsymbol{b}$ such that the matrix $A_\theta \in \mathbb{R}^{d \times d}$ is parametrised by θ, the vector $\boldsymbol{b} \in \mathbb{R}^d$ is known and $\boldsymbol{\xi}$ represents measurement noise. Such situations arise in inverse problems constrained by PDEs, where $\boldsymbol{x}$ represents a discrete approximation to the solution of the PDE (e.g. Biegler et al., 2003). In the context of the full inference problem for both the solution $\boldsymbol{x}$ to the forward problem and the parameter θ, one could think of a joint prior over the forward and inverse problems of the form

$$p(\boldsymbol{x}, \theta) = p(\theta) p(\boldsymbol{x} | \theta)$$

so that the prior over $\boldsymbol{x}$ is specified conditional on θ (Cockayne et al., 2016). This seems intuitively reasonable — in general there will be a dependence of $\boldsymbol{x}$ on θ — and a prior should acknowledge that. Thus, the use of a preconditioner prior $p(\boldsymbol{x}|\theta)$ based on A_θ provides an automatic mechanism to encode the dependence of $\boldsymbol{x}$ on θ.

Calvetti further commented on the lack of a comparison with conjugate gradient least squares (CGLS) methods (Calvetti et al., 2012) in Section 6.2 of Cockayne et al. (2019c). Calvetti et al. (2012) presented fast techniques for solving the linear systems arising in the inverse problem; these are based on using the prior covariance of a highly structured prior as a right preconditioner, and a decomposition of the noise covariance as a left preconditioner. As mentioned, this bears a strong resemblance to the observation from Li and Fang. However we feel that the experimental results in Section 6.2 seek to achieve something quite different to the goal of Calvetti et al. (2012). Our goal was to construct uncertainty estimates for the forward problem and propagate these to the inverse problem, rather than to produce a competitive new approach to solving the electrical impedance tomography (EIT) inverse problem. Thus we feel the benefit of a

computational comparison would be limited. However we would be interested to discuss the conceptual overlap between the two approaches further, and in particular to examine whether techniques for constructing preconditioners for the inverse problem could also be applied to the forward problem.

4 Uncertainty Quantification

Several authors have commented on the uncertainty quantification (UQ) issue highlighted in Section 6.1 of Cockayne et al. (2019a), namely that the UQ provided by the posterior appears to be poorly calibrated. This is owing to the fact that the search directions are constructed in a "data-driven" manner in that they depend on $\boldsymbol{x}^*$, and this induces nonlinearity in the information that is not acknowledged in the conditioning procedure we adopt. Indeed, the search directions $\boldsymbol{s}_i$ depend on $\boldsymbol{x}^*$; i.e. $\boldsymbol{s}_i = \boldsymbol{s}_i(\boldsymbol{x}^*)$, $i = 1, \ldots, d$. Thus the information $y_i \in \mathbb{R}^d$ can be considered as the output of a nonlinear map

$$y_i = \mathrm{y}_i(\boldsymbol{x}^*) = \boldsymbol{s}_i(\boldsymbol{x}^*)^\top A\boldsymbol{x}^*.$$

If the Gaussian prior $\mu \in \mathcal{P}(\mathbb{R}^d)$ has full support then the set $\mathcal{D} := \{\boldsymbol{x} \in \mathbb{R}^d : \mathrm{y}_i(\boldsymbol{x}) = y_i, i = 1, \ldots, m\}$ is a μ-null set. As such, "conditioning on $\mathcal{D}$" is not well-defined and measure-theoretic notions of disintegration of measure are required (Chang and Pollard, 1997). In particular, the "exact" posterior arises from disintegrating the map $\boldsymbol{x} \mapsto [\mathrm{y}_1(\boldsymbol{x}), \ldots, \mathrm{y}_m(\boldsymbol{x})]$. In BayesCG, the dependence of $\boldsymbol{s}_i$ on $\boldsymbol{x}^*$ is neglected. As such, the BayesCG output μ_m arises from disintegrating a linear map $\boldsymbol{x} \mapsto [\boldsymbol{s}_1^\top A\boldsymbol{x}, \ldots, \boldsymbol{s}_m^\top A\boldsymbol{x}]$, and this explains why the UQ provided by BayesCG is not well-calibrated.

Sullivan remarks that the additional cost of BayesCG over CG would be a small price to pay if the UQ were well-calibrated, and that it is indeed unfortunate that it proves not to be. In light of our comments that BayesCG is not strictly Bayesian, he asks whether it is possible to compare the distribution that forms the output of BayesCG to the *exact* Bayesian posterior using techniques from our earlier paper, Cockayne et al. (2019c). We agree that this would be a worthwhile experiment to perform. Unfortunately we have not been able to prepare results for this rejoinder, but we hope to present some results in this direction in a future publication.

Hennig (2015) also commented on the UQ provided. He notes that the problem with the hierarchical approach employed in Section 4.2 is that it violates the assumption of independent and identically distributed observations required by the hierarchical inference framework. Indeed, the search directions, and thus the observations, are correlated through their recursive definition, but they also represent nonlinear information owing to their dependence on $\boldsymbol{x}^*$. Thus we are sceptical than the UQ can be repaired within the Bayesian framework, since we have already departed from that framework by ignoring this nonlinearity. The difficulty of this is further highlighted by the theoretical results highlighted in the paper; Proposition 3 of Cockayne et al. (2019a) shows that a measure of the size of Σ_m converges at a linear rate, while Proposition 10 shows that the posterior mean converges exponentially fast.

This does not eliminate use of a more *ad-hoc* calibration procedure, such as the procedure in Section 2.1 of his response that significantly expands on an approach we

sketched in Section S4.3 of the supplement. The approach he outlines looks promising and certainly, from the perspective of the matrix elements, looks to produce quite well calibrated UQ. We would be interested in collaborating to develop these ideas further, particularly since — as noted above — such efforts would be of value both in the prior on A^{-1} and prior on $\boldsymbol{x}$ scenarios.

4.1 BayesCG and Filtering

Sullivan further asked whether BayesCG can be interpreted as a filtering method. This is a direction that we have investigated to some extent. Certainly, in the case that the search directions are chosen *independently* of $\boldsymbol{x}^*$, the posterior distribution from Cockayne et al. (2019a, Proposition 1) is identical to what would be obtained from application of a Kalman filter, owing to the equivalence between conditioning Gaussians sequentially and in a single batch procedure. However, given the nonlinearity induced by the dependence of the search directions on $\boldsymbol{x}^*$, a natural question is whether any of the *non*linear filters that exist in the literature could be applied to correct the UQ and also obtain fast convergence. In particular we have investigated the extended Kalman filter (EKF; see Law et al., 2015), and we outline our analysis here.

In the EKF the information functional $\mathrm{y}_{m+1} : \mathbb{R}^d \to \mathbb{R}$ is approximated by a linearisation about the point $\tilde{\boldsymbol{x}}$, of the form

$$\bar{\mathrm{y}}_{m+1}(\boldsymbol{x}; \tilde{\boldsymbol{x}}) = \mathrm{y}_{m+1}(\boldsymbol{x}_m) + \frac{\partial \mathrm{y}_{m+1}}{\partial \boldsymbol{x}}(\tilde{\boldsymbol{x}})^\top (\boldsymbol{x} - \tilde{\boldsymbol{x}}) + \epsilon_m$$

where ϵ_m is noise introduced to relax the problem and account for the linearisation error. The (intractable) exact posterior at iteration $m+1$ is then recursively approximated by the distribution

$$\bar{\mu}_{m+1} := \bar{\mu}_m | \{\boldsymbol{x} : \bar{\mathrm{y}}_{m+1}(\boldsymbol{x}; \bar{\boldsymbol{x}}_m) = y_{m+1}\}$$

where $\bar{\mu}_0 = \mu_0$ and $\bar{\boldsymbol{x}}_m$ is the mean of $\bar{\mu}_m$. Note the incorrect conditioning here; we are conditioning on the value y_{m+1} arising from the true information functional $\mathrm{y}_{m+1}(\boldsymbol{x}^*)$, but relating it to $\bar{\mu}_m$ through the linearised information operator $\bar{\mathrm{y}}_{m+1} \neq \mathrm{y}_{m+1}$. This can be viewed as a specific instance of an approximate likelihood.

The derivatives $\frac{\partial \mathrm{y}_m}{\partial \boldsymbol{x}}$ can be computed in closed-form and thus the above method can be implemented and empirically assessed. Unfortunately, the results were negative; while the UQ provided by $\bar{\mu}_m$ does appear to be well-calibrated in the sense of Section 6.1 of Cockayne et al. (2019a), the mean $\bar{\boldsymbol{x}}_m$ does not appear to have the exponential convergence property exhibited by BayesCG. Moreover, the posterior distribution $\bar{\mu}_{m+1}$ does not appear to coincide in any sense with a classical numerical method. This does not preclude the possibility that other variants of filters might yield more practical numerical methods, however this is not an avenue that we are currently exploring.

5 Extensions and Generalisations

Several of the discussants asked about generalisations or presented interesting possible extensions of Cockayne et al. (2019a). We are pleased that our paper has already inspired

such interesting new work and some brief responses to the discussants' proposals are presented next.

5.1 Singular Matrices

While A is assumed to be invertible in Cockayne et al. (2019a), Sullivan asked how the algorithm performs in the case when A is indefinite; either because it is square but not invertible, or because it is rectangular. Naturally, in this setting the problem is ill-posed, and so the system either has no solution or has infinitely-many solutions. In the latter case a natural thing to do is to regularise the problem in some way to obtain a least-squares solution to the problem.

We believe that BayesCG is robust in this setting, because the Bayesian formulation provides such a regularisation and also provides a natural representation of uncertainty in the solution that ought to arise from it not having a unique solution. To show that the posterior is still well-defined, suppose that $A \in \mathbb{R}^{d\times n}$ is an arbitrary, potentially singular matrix. The central object that must be examined is the Krylov space $K_m^* = \boldsymbol{x}_0 + K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$, which determines the behavior of the iterate and the posterior covariance as revealed in Proposition 3. Since $A\Sigma_0 A^\top$ is positive semidefinite irrespective of A the Krylov space will still be well-defined, though the dimension of $K^*_{\max(d,n)}$ may not be $\max(d,n)$. Furthermore we note that it may be the case that $\boldsymbol{x}_* \notin K^*_{\max(d,n)}$, but Proposition 9 of Cockayne et al. (2019a) nevertheless guarantees that the solution is optimal in that space according to the prior precision norm. Furthermore the projection properties described in Proposition 3 of this work are maintained, so that the posterior covariance will precisely be the prior uncertainty outside of the space that has been explored.

We further note that the singular setting is the setting considered in many of the works cited in Calvetti's discussion, in which similar results are obtained to those presented in our paper, but the matrix A is assumed to arise from some underdetermined inverse problem and thus to be rectangular. In particular, the priorconditioned CGLS algorithm for solving such underdetermined systems, introduced in Calvetti et al. (2017), reports similar results for the Krylov space occupied by the iterate in the algorithm presented therein, though we note that the results do not appear to be identical as the Krylov space reported is

$$\boldsymbol{x}_m \in K_m(A\Sigma_0 A^\top, AL\boldsymbol{b})$$

where L is the Cholesky factor of Σ_0. This does not precisely coincide with the Krylov space reported in our work, which is (assuming a zero-mean)

$$\boldsymbol{x}_m \in \Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{b}).$$

However some similarities are nevertheless apparent.

5.2 Precision Updates

Sullivan also asks whether the algorithm can be formulated in terms of *precision* updates of Σ_m^{-1} rather than covariance updates of Σ_m. We do not believe this to be the

case, owing to the fact that the Sherman-Morrison formula breaks down in the setting of noiseless observations in Cockayne et al. (2019a). At a more basic level, as shown in Section S3.1 of the supplement, when the prior $\Sigma_0 = (A^\top A)^{-1}$ is used BayesCG converges in a single iteration; thus, if a practical update of the precision matrix were possible it would yield an iterative method that converged in a single iteration for general linear systems, which would be quite remarkable if the cost remained $\mathcal{O}(d^2)$!

5.3 Multiple Related Linear Systems

Briol, DiazDelaO and Hristov provide an interesting examination of a generalisation of BayesCG to a setting where multiple related linear systems must be solved simultaneously[2], i.e.

$$A_j \boldsymbol{x}_j^* = \boldsymbol{b}_j$$

$j = 1, \ldots, J$. The approach they suggest is to "stack" the problems into a single large linear system $\underline{A}\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$ and apply BayesCG to this system in an approach they call the *multi-system BayesCG algorithm*.

This is a problem that we have also considered, and which we feel may be a compelling application of BayesCG. The multi-system approach they highlight is somewhat of an extreme, as it requires solving a single problem of significantly higher dimension than each individual problem, whereas in classical Krylov subspace recycling schemes the problems are generally assumed to be solved sequentially (see e.g. Parks et al., 2006). On the other hand, the stacked system is in some sense the prototypical problem, in that if some acceleration of convergence can be achieved by solving the sub-problems iteratively and recycling information between them then we ought to be able to observe accelerated convergence on this stacked problem, with a suitable choice of prior.

Relatedly, if the J linear systems arise in such a way that the $\boldsymbol{x}_j^*$ can be considered as approximately independent draws from some distribution in $\mathcal{P}(\mathbb{R}^d)$, then one may attempt to solve a small number $k \ll J$ of the linear systems to high accuracy and then to apply BayesCG to the remainder using a prior that is based on the empirical distribution of the $\boldsymbol{x}_k^*$ previously obtained. In general one can envisage a number of strategies to deal with related linear systems, up to and including the construction of a full Bayesian hierarchical model.

5.4 Application to Gaussian Process Regression

The contributed discussion from Simone Rossi, Christian Rusu, Lorenzo Rosasco and Maurizio Filippone investigated applying BayesCG to an inversion problem arising from Gaussian process regression for a function $f(t)$, namely that of calculating the predictive distribution for new observations of $f(\tilde{t})$ at points $\tilde{t}$ outside of the set of training design points. Using BayesCG to solve the linear system that arises in computing the posterior mean function, they show that it is possible to propagate the uncertainty from BayesCG

[2] As an aside, from correspondence with the authors of the comment we have determined that there is a typographical error in their response, regarding the equivalent of the equation below.

into the Gaussian process covariance explicitly, in much the same way as in the inverse problem we discuss in Section 6.2 of our paper. We note that this is also related to work in Bartels and Hennig (2016). The results they present investigate using different choices of preconditioner from the literature on Gaussian processes to construct the prior, and show impressive performance for the convergence of the posterior mean.

It would be interesting to see how this compared to using standard CG, as well as preconditioned CG, to solve the inversion problem, and also to see how well-calibrated the posterior covariance in their expression for $p(\tilde{\boldsymbol{y}}|X, \boldsymbol{y}, \tilde{X})$ is. One would expect, given the poor calibration for BayesCG in general, that it would be a relatively poor predictor of the uncertainty in the predicted values, but this would be useful to check. One minor criticism of this uncertainty propagation procedure is that, unless we are mistaken, the same linear system must be solved to compute the matrix $\Sigma_{\tilde{\boldsymbol{y}}}$ in their notation, and we do not believe the same uncertainty propagation can be accomplished in closed form in that case.

5.5 Model Averaging

Rossi, Rusu, Rosasco and Filippone also discussed averaging the posterior distribution produced from different choices of prior. They suppose that a distribution $p(\Sigma_0)$ is placed over different prior covariances that are derived, in their work, from different choices of centre for a Nyström preconditioner applied to an inversion problem from Gaussian process regression. Their simulations indicate that the convergence rate of the posterior mean is accelerated through this procedure.

This suggests a number of interesting possible extensions to BayesCG. Inspired by multigrid methods for PDEs, suppose that we have J preconditioners $P^1, \ldots, P^J$, where the computational cost of applying $(P^j)^{-1}$ is greater than that of applying $(P^{j-1})^{-1}$, for $j = 2, \ldots J$. Then, by placing an appropriately decaying probability distribution over $\{1, \ldots, J\}$ one might mix these preconditioners in a probabilistic framework. Whether this provides a particular advantage over the standard multigrid method is not immediately clear, but it would be interesting to investigate. Similarly, there is a literature on producing random preconditioners (e.g. Avron et al., 2010; Meng et al., 2014; Yang et al., 2015; Avron et al., 2017), and the randomness in the preconditioner could be elegantly incorporated into the solution of the linear system using this framework.

6 Other Remarks

Lastly, there are several comments from the discussions which do not fit into any of the sections above.

6.1 Active Probabilistic Solvers

Hennig (2015) made an important distinction in his comment between what he terms *active* probabilistic solvers, which attempt to use the probabilistic viewpoint to improve

upon point estimates from classical solvers (in some sense), and *passive* solvers that attempt to endow existing solvers with UQ. We concur that most existing work on probabilistic linear algebra, including Cockayne et al. (2019a), are passive solvers by this definition, though we do respectfully disagree with his statement in Section 3, that the "ambition should be to build *active* probabilistic linear solvers"; we believe that efficient passive solvers are equally of interest. Even should fast active solvers be discovered, incorporating UQ into code bases that exploit classical solvers represents much less of a drastic change to existing code if a passive solver replaces its classical counterpart than if an entirely new numerical method is introduced.

Constructing active solvers is an interesting proposition. In the context of BayesCG opportunities for constructing priors to accelerate convergence might come from knowledge about the provenance of the underlying problem. For example, if we know the problem to be derived from discretisation of a linear PDE, using a prior on $\boldsymbol{x}^*$ based on discretisation of a smooth Gaussian process is likely to yield a faster rate of convergence, such as those rates proven for this problem in the function-space setting in Cockayne et al. (2016). On the other hand, as Li and Fang elucidate in their discussion, since the iterate from BayesCG is equivalent to solution of a right-preconditioned problem with CG, this choice of prior is equivalent to using that information to construct a preconditioner to accelerate CG. Thus, merely constructing a more informative prior for BayesCG is insufficient to outperform classical methods; one would need to exploit the probabilistic perspective in a deeper and more subtle way.

From a wider perspective, the pursuit of active probabilistic linear solvers seems to be an interesting line of research, on which we would be interested to collaborate.

6.2 Miscellanea

Sullivan asked whether the square-root $M^{\frac{1}{2}}$ introduced before Proposition 5 needs to have a specific form, or if it is arbitrary. The choice of M is indeed arbitrary; a Cholesky factor is equally suitable as a symmetric square root.

Sullivan also objected to our use of the phrase "probabilistic termination criteria" in Section 5, since the criterion we use is in fact deterministic. We agree that this was a poor choice of words on our part; perhaps instead we should have called this a probabilistically *motivated* termination criterion.

Calvetti commented on the paucity of references to the existing literature on EIT as a Bayesian inversion problem in Section 6.2 of Cockayne et al. (2019a). We apologise for the lack of references here. Our focus was on demonstrating how the *probabilistic* UQ provided over the solution to the forward problem could be used within an inference problem, a topic which, to our knowledge, the existing literature on EIT has not considered. However our references around the history of EIT are certainly incomplete, and we thank her for highlighting them in her discussion.

7 Conclusion

We would like once again to thank all of the discussants for their valuable and insightful feedback. We are delighted to have provoked so much discussion and, indeed, original work. This process has raised several interesting new lines of research that are worthy of investigation, and we hope to pursue those new directions collaboratively going forward.

A Proofs

Proof of Proposition 1. Let $\Lambda_m = S_m^\top A \Sigma_0 A^\top S_m$. First recall from Cockayne et al. (2019a, Proposition 3) that $\mathrm{trace}(\Sigma_m \Sigma_0^{-1}) = d - m$. It is straightforward to show that $\Sigma_m \Sigma_0^{-1}$ is idempotent, since:

$$\begin{aligned}
\Sigma_m \Sigma_0^{-1} &= I - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \\
(\Sigma_m \Sigma_0^{-1})^2 &= (I - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A)(I - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A) \\
&= I - 2\Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A + \Sigma_0 A^\top S_m \Lambda_m^{-1} \underbrace{S_m^\top A \Sigma_0 A^\top S_m}_{=\Lambda_m} \Lambda_m^{-1} S_m^\top A \\
&= I - 2\Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A = \Sigma_m \Sigma_0^{-1}.
\end{aligned}$$

We therefore have that the eigenvalues of $\Sigma_m \Sigma_0^{-1}$ are either 0 or 1, and since the trace is the sum of the eigenvalues it holds that $\Sigma_m \Sigma_0^{-1}$ has rank $d - m$. Therefore $\Sigma_0 \Sigma_m \Sigma_0^{-1}$ has rank $d - m$ since Σ_0 is full-rank. Since this matrix is similar to Σ_m, it follows that Σ_m is also of rank $d - m$, which completes the proof. □

Proof of Proposition 2. We have that

$$\begin{aligned}
\mathrm{trace}(\Sigma_m) &= \mathrm{trace}(\Sigma_0 \Sigma_m \Sigma_0^{-1}) \\
&= \langle \Sigma_0, \Sigma_m \Sigma_0^{-1} \rangle_F
\end{aligned}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product, defined by $\langle A, B \rangle_F = \mathrm{trace}(A^\top B)$. By applying Cauchy–Schwarz we then have

$$\begin{aligned}
\mathrm{trace}(\Sigma_m) &\leq \|\Sigma_0\|_F \|\Sigma_m \Sigma_0^{-1}\|_F \\
&= \mathrm{trace}(\Sigma_0)^{\frac{1}{2}} \|\Sigma_m \Sigma_0^{-1}\|_F.
\end{aligned}$$

Lastly recall that $\|A\|_F^2$ is the sum of the squared singular values of A. In the present setting the squared singular values of $\Sigma_m \Sigma_0^{-1}$ are precisely its nonzero eigenvalues, and from Cockayne et al. (2019a, Proposition 3) we know the sum of these eigenvalues to be $d - m$. Hence

$$\mathrm{trace}(\Sigma_m) \leq \mathrm{trace}(\Sigma_0)^{\frac{1}{2}} (d - m)^{\frac{1}{2}}$$

as required. □

Proof of Proposition 3. Let S_m denote the BayesCG search directions and recall that $\text{range}(S_m) = K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$. Observe that since S_m are $A\Sigma_0 A^\top$-orthonormal, it follows that $\Sigma_0 A^\top S_m$ is a Σ_0^{-1}-orthonormal basis of $\mathcal{K}_m$. Further note that if Q denotes an orthogonal projector with-respect-to the standard Euclidean inner-product, then $P = Q\Sigma_0^{-1}$ denotes an orthogonal projector with-respect-to the inner product induced by Σ_0^{-1}. Thus the required orthogonal projection onto $\mathcal{K}_m$ is $P = \Sigma_0 A^\top S_m S_m^\top A$. Now note that:

$$\begin{aligned} P\Sigma_0 A^\top S_m &= \Sigma_0 A^\top S_m \underbrace{S_m^\top A \Sigma_0 A^\top S_m}_{=I} \\ &= \Sigma_0 A^\top S_m \end{aligned}$$

and furthermore

$$P^2 = P\Sigma_0 A^\top S_m S_m^\top A = P.$$

Now consider $P_{\#}\mu_m$. Owing to the conjugacy of Gaussian distributions with linear maps, it suffices to check that $P\boldsymbol{x}_m = P\boldsymbol{x}_0$ and $P\Sigma_m P^\top = P\Sigma_0 P^\top$. We have that

$$\begin{aligned} P\boldsymbol{x}_m &= P\boldsymbol{x}_0 + P\Sigma_0 A^\top S_m S_m^\top \boldsymbol{r}_0 \\ &= P\boldsymbol{x}_0 + \underbrace{\Sigma_0 A^\top S_m S_m^\top A}_{=P}(\boldsymbol{x}^* - \boldsymbol{x}_0) \\ &= P\boldsymbol{x}_0 + P\boldsymbol{x}^* - P\boldsymbol{x}_0 = P\boldsymbol{x}_* \\ P\Sigma_m P^\top &= P\Sigma_0 P^\top - P\Sigma_0 A^\top S_m S_m^\top A \Sigma_0 P^\top \\ &= P\Sigma_0 P^\top - P\Sigma_0 P^\top = 0 \end{aligned}$$

as required.

Next consider $P_{\#}^\perp \mu_m = (I-P)_{\#}\mu_m$. Proceeding as above, we have:

$$\begin{aligned} (I-P)\boldsymbol{x}_m &= \boldsymbol{x}_m - P\boldsymbol{x}^* \\ &= \boldsymbol{x}_0 + \Sigma_0 A^\top S_m S_m^\top A(\boldsymbol{x}^* - \boldsymbol{x}_0) - P\boldsymbol{x}^* \\ &= \boldsymbol{x}_0 + P\boldsymbol{x}^* - P\boldsymbol{x}_0 - P\boldsymbol{x}^* = (I-P)\boldsymbol{x}_0 \\ (I-P)\Sigma_m(I-P)^\top &= (I-P)\Sigma_0(I-P)^\top \\ &\quad - (I-P)\Sigma_0 A^\top S_m S_m^\top A \Sigma_0 (I-P)^\top \\ (I-P)\Sigma_0 A^\top S_m S_m^\top A \Sigma_0 (I-P)^\top &= (I-P)P\Sigma_0(I-P^\top) \\ &= P\Sigma_0 - P^2\Sigma_0 - P\Sigma_0 P^\top + P^2\Sigma_0 P^\top \\ &= P\Sigma_0 - P\Sigma_0 - P\Sigma_0 P^\top + P\Sigma_0 P^\top = 0 \end{aligned}$$

where the last line follows from the fact that $P^2 = P$. Thus $P_{\#}^\perp \mu_m = P_{\#}^\perp \mu_0$, which completes the proof. □

Proof of Proposition 4. Let S_m denote the search directions from BayesCG. Then

$$\begin{aligned}\Sigma_m &= \Sigma_0 - \Sigma_0 A^\top S_m S_m^\top A \Sigma_0 \\ \Sigma_m A^\top S_m &= \Sigma_0 A^\top S_m - \Sigma_0 A^\top S_m S_m^\top A \Sigma_0 A^\top S_m \\ &= \Sigma_0 A^\top S_m - \Sigma_0 A^\top S_m = 0.\end{aligned}$$

Furthermore since $\operatorname{rank}(\Sigma_m) = d - m$ and $\operatorname{rank}(A^\top S_m) = m$, it must hold that $\operatorname{span}(A^\top S_m)$ is the entire null-space of Σ_m. □

References

Avron, H., Clarkson, K. L., and Woodruff, D. P. (2017). "Faster Kernel Ridge Regression Using Sketching and Preconditioning." *SIAM Journal on Matrix Analysis and Applications*, 38(4): 1116–1138. MR3713904. doi: https://doi.org/10.1137/16M1105396. 1007

Avron, H., Maymounkov, P., and Toledo, S. (2010). "Blendenpik: Supercharging LAPACK's Least-Squares Solver." *SIAM Journal on Scientific Computing*, 32(3): 1217–1236. MR2639236. doi: https://doi.org/10.1137/090767911. 1007

Bartels, S., Cockayne, J., Ipsen, I., and Hennig, P. (2019). "Probabilistic Linear Solvers: A Unifying View." *Statistics and Computing*. To appear. 1001

Bartels, S. and Hennig, P. (2016). "Probabilistic Approximate Least-Squares." In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51 of *JMLR Workshop and Conference Proceedings*, 676–684. URL http://jmlr.org/proceedings/papers/v51/bartels16.html 1001, 1007

Biegler, L. T., Ghattas, O., Heinkenschloss, M., and van Bloemen Waanders, B. (2003). "Large-scale PDE-constrained optimization: an introduction." In *Large-Scale PDE-Constrained Optimization*, volume 30 of *Lecture Notes in Computational Science and Engineering*, 3–13. Springer. MR2038928. doi: https://doi.org/10.1007/978-3-642-55508-4_1. 1002

Calvetti, D., McGivney, D., and Somersalo, E. (2012). "Left and right preconditioning for electrical impedance tomography with structural information." *Inverse Problems*, 28(5): 055015. MR2923200. doi: https://doi.org/10.1088/0266-5611/28/5/055015. 1002

Calvetti, D., Pitolli, F., Prezioso, J., Somersalo, E., and Vantaggi, B. (2017). "Priorconditioned CGLS-Based Quasi-MAP Estimate, Statistical Stopping Rule, and Ranking of Priors." *SIAM Journal on Scientific Computing*, 39(5): S477–S500. MR3716568. doi: https://doi.org/10.1137/16M108272X. 999, 1000, 1005

Calvetti, D., Pitolli, F., Somersalo, E., and Vantaggi, B. (2018). "Bayes meets Krylov: Statistically inspired preconditioners for CGLS." *SIAM Review*, 60(2): 429–461. MR3797727. doi: https://doi.org/10.1137/15M1055061. 998

Calvetti, D. and Somersalo, E. (2005). "Priorconditioners for linear systems." *Inverse Problems*, 21(4): 1397–1418. MR2158117. doi: https://doi.org/10.1088/0266-5611/21/4/014. 1002

Chang, J. T. and Pollard, D. (1997). "Conditioning as disintegration." *Statistica Neerlandica*, 51(3): 287–317. MR1484954. doi: https://doi.org/10.1111/1467-9574.00056. 1003

Cockayne, J., Oates, C., Ipsen, I., and Girolami, M. (2019a). "A Bayesian Conjugate Gradient Method." *Bayesian Analysis*. Advance publication. 997, 998, 999, 1000, 1001, 1003, 1004, 1005, 1006, 1008, 1009

Cockayne, J., Oates, C., Ipsen, I., and Girolami, M. (2019b). "Supplementary Material for "A Bayesian Conjugate-Gradient Method"." *Bayesian Analysis*. MR3577381. doi: https://doi.org/10.1214/16-BA1038. 999

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2016). "Probabilistic Numerical Methods for Partial Differential Equations and Bayesian Inverse Problems." ArXiv:1605.07811. MR3577382. doi: https://doi.org/10.1214/16-BA1017A. 1002, 1008

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2019c). "Bayesian probabilistic numerical methods." *SIAM Review*. To appear. 1002, 1003

Hennig, P. (2015). "Probabilistic Interpretation of Linear Solvers." *SIAM Journal on Optimization*, 25(1): 234–260. URL http://epubs.siam.org/toc/sjope8/25/1. MR3301314. doi: https://doi.org/10.1137/140955501. 1001, 1003, 1007

Law, K., Stuart, A., and Zygalakis, K. (2015). *Data Assimilation*. Springer International Publishing. MR3363508. doi: https://doi.org/10.1007/978-3-319-20325-6. 1004

Meng, X., Saunders, M. A., and Mahoney, M. W. (2014). "LSRN: A Parallel Iterative Solver for Strongly Over- or Underdetermined Systems." *SIAM Journal on Scientific Computing*, 36(2): C95–C118. MR3172249. doi: https://doi.org/10.1137/120866580. 1007

Parks, M. L., de Sturler, E., Mackey, G., Johnson, D. D., and Maiti, S. (2006). "Recycling Krylov Subspaces for Sequences of Linear Systems." *SIAM Journal on Scientific Computing*, 28(5): 1651–1674. MR2272183. doi: https://doi.org/10.1137/040607277. 1006

Yang, J., Chow, Y.-L., Ré, C., and Mahoney, M. W. (2015). "Weighted SGD for ℓp Regression with Randomized Preconditioning." In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. MR3478417. doi: https://doi.org/10.1137/1.9781611974331.ch41. 1007