

Towards Matrix-Free AD-Based Preconditioning of KKT Systems in PDE-Constrained Optimization

Roland Griesse^{*1} and Andrea Walther²

¹ Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz, Austria

² Institute of Scientific Computing, Technische Universität Dresden, Dresden, Germany

The presented approach aims at solving an equality constrained, finite-dimensional optimization problem, where the constraints arise from the discretization of some partial differential equation (PDE) on a given space grid. For this purpose, a stationary point of the Lagrangian is computed using Newton's method, which requires the repeated solution of KKT systems. The proposed algorithm focuses on two topics: Firstly, Algorithmic Differentiation (AD) will be used to evaluate the necessary computations of gradients, Jacobian-vector products, and Hessian-vector products, so that only the objective $f(y, u)$ and the PDE constraint $e(y, u) = 0$ have to be specified by the user. Secondly, we solve the KKT system iteratively using the QMR algorithm, with preconditioning provided by a multigrid approach. We wish to explore whether the Jacobian-vector products provided by AD are sufficient to construct suitable multigrid preconditioners. Our approach is then embedded into a globalized optimization routine. Numerical results for optimization problems involving a nonlinear reaction-diffusion model will be given.

Copyright line will be provided by the publisher

1 Introduction

We consider a finite-dimensional optimization problem of the form

$$\begin{aligned} \text{Minimize} \quad & f(y, u) \\ \text{subject to} \quad & e(y, u) = 0. \end{aligned}$$

The equality constraint $e(y, u) = 0$ represents the discretization of some stationary partial differential equation (PDE), where y denotes the state variable and u the control variable, respectively. With the Lagrangian of the problem defined by $\mathcal{L}(y, u, \lambda) = f(y, u) + \lambda^\top e(y, u)$, the stationarity condition $\nabla \mathcal{L}(y, u, \lambda) = 0$ is solved using an inexact variant of Newton's method. Hence, one repeatedly has to solve KKT systems of the form

$$\nabla^2 \mathcal{L} \cdot p = \begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & e_y^\top \\ \mathcal{L}_{uy} & \mathcal{L}_{uu} & e_u^\top \\ e_y & e_u & 0 \end{pmatrix} \begin{pmatrix} \delta y \\ \delta u \\ \delta \lambda \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_y \\ \mathcal{L}_u \\ e \end{pmatrix} \quad (1)$$

to compute the Newton steps p . A number of optimization algorithms employ the so-called reduced space approach for the efficient calculation of p , making use of the block structure of the KKT matrix $\nabla^2 \mathcal{L}$, see, e.g., [8, 9]. However, the reduced space approach usually requires the exact and thus expensive solution of the linearized state and adjoint equations at each iteration. Therefore, we consider a full space approach, the Lagrange-Newton SQP method, which aims at solving the whole KKT system by a preconditioned iterative Krylov method. Here, structural information about the optimization problem is used to provide an efficient preconditioner based on the control space Schur complement. A similar solution procedure was proposed for example in [3].

In the project presented in this paper, we focus on the efficient solution of subproblems (1) in the context of PDE-constrained optimization. Our work is based on the paradigm that only the objective $f(y, u)$ and the PDE constraint $e(y, u) = 0$ should have to be specified. Such an approach appears especially useful when solvers for the underlying PDE, its linearization and its adjoint are not available but when a rapid glance at the optimal solution is sought without the error-prone process of coding these solvers first. Hence, any additional information beyond f and e needed by the optimization routine should be generated automatically. To this end, we apply Algorithmic Differentiation (AD) to evaluate the products of derivative matrices and vectors. A comprehensive introduction to this method for computing exact derivative information can be found in [5]. Furthermore, we want to avoid the costly procedures of assembling and factoring any derivative matrices as far as possible. In particular, we shall employ the second order adjoint mode of AD to evaluate Hessian-vector products without evaluating the Hessian itself. To accelerate the iterative solution of the usually very ill-conditioned KKT system (1), we use a suitable preconditioner which, in part, is constructed from a geometric multigrid method with adaptive Richardson smoothing. Note that following the above design paradigm, this multigrid preconditioner must also be constructed automatically. Then, a LU decomposition of the constraint Jacobian e_y is required only on the coarsest grid, on which the complexity of assembling and

* Corresponding author: e-mail: roland.griesse@oeaw.ac.at, Phone: +43 (0)732 2468 5218, Fax: +43 (0)732 2468 5212

factoring e_y is considerably smaller than on fine grids. A first feasibility study is given for a nonlinear 2D reaction-diffusion problem where the proposed algorithm shows some promising behavior.

2 Solving the KKT System

The KKT matrix $\nabla^2 \mathcal{L}$ is known to be symmetric but indefinite. Therefore, one could apply MINRES, an iterative method that exploits the symmetry. However, preconditioners used in a MINRES iteration have to be positive definite. To overcome this restriction, we use a symmetric variant of QMR as an iterative solver, which requires a symmetric preconditioner [4]. To motivate the choice of the preconditioner, we consider the LU factorization of the KKT matrix matrix $\nabla^2 \mathcal{L}$,

$$\nabla^2 \mathcal{L} = \begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & e_y^\top \\ \mathcal{L}_{uy} & \mathcal{L}_{uu} & e_u^\top \\ e_y & e_u & 0 \end{pmatrix} = \begin{pmatrix} \mathcal{L}_{yy} e_y^{-1} & 0 & I \\ \mathcal{L}_{uy} e_y^{-1} & I & e_u^\top e_y^{-\top} \\ I & 0 & 0 \end{pmatrix} \begin{pmatrix} e_y & e_u & 0 \\ 0 & H_R & 0 \\ 0 & B & e_y^\top \end{pmatrix}$$

with the reduced Hessian H_R and B defined as

$$H_R = (-e_u^\top e_y^{-\top} \quad I) \begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} \\ \mathcal{L}_{uy} & \mathcal{L}_{uu} \end{pmatrix} \begin{pmatrix} e_y^{-1} e_u \\ I \end{pmatrix} \quad \text{and} \quad B = (I \quad 0) \begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} \\ \mathcal{L}_{uy} & \mathcal{L}_{uu} \end{pmatrix} \begin{pmatrix} e_y^{-1} e_u \\ I \end{pmatrix}.$$

Throughout, I denotes the unit matrix of suitable size. Following the ideas presented in [3], the preconditioner P_2 is defined by canceling the terms involving second-order derivatives except H_R in the factors of $\nabla^2 \mathcal{L}$, i.e.,

$$P_2 = \begin{pmatrix} 0 & 0 & I \\ 0 & I & e_u^\top e_y^{-\top} \\ I & 0 & 0 \end{pmatrix} \begin{pmatrix} e_y & e_u & 0 \\ 0 & H_R & 0^\top \\ 0 & 0 & e_y^\top \end{pmatrix} = \begin{pmatrix} 0 & 0 & e_y^\top \\ 0 & H_R & e_u^\top \\ e_y & e_u & 0 \end{pmatrix}. \quad (2)$$

This preconditioner in combination with the iterative solution of (1) by GMRES was analyzed in [1]. However, since P_2 is symmetric, one may also employ the symmetric QMR method. In contrast to the standard QMR method, it requires only one matrix–vector product per iteration [4]. The application of the preconditioner P_2 from the left requires the evaluation of products of the form $P_2^{-1}v$. By forming the inverse of P_2 , one finds that each multiplication with P_2^{-1} amounts to one solve with e_y^\top and one solve with e_y (hence the name P_2 was chosen in [3]), plus one solve with H_R . Note that this is true only for left-preconditioning, i.e., for the solution of the system $P_2^{-1} \nabla^2 \mathcal{L} \cdot p = -P_2^{-1} \nabla \mathcal{L}$. For left-right preconditioning, the number of solves involving e_y or e_y^\top increases to four.

In this first feasibility study, we use the rather crude approximation $\tilde{H}_R = I$ of the reduced Hessian H_R , hence the dominant cost for evaluating the product $P_2^{-1}v$ is given by the two system solves involving e_y and e_y^\top , i.e., by the solution of the linearized and the adjoint PDE. To give the full picture, we expect that much can be gained by a careful selection of a preconditioner for H_R , but we postpone this discussion to a forthcoming extended version of this paper. In order to reduce the overall cost of applying the preconditioner, we wish to replace e_y and e_y^\top by approximations/preconditioners \tilde{e}_y and \tilde{e}_y^\top , respectively, which are much cheaper to invert. One preconditioning approach that is compatible with the design requirement that the preconditioner should be constructed automatically with as little user interaction as possible is the geometric multigrid method [6, 7]. Hence the method proposed here differs from the examples in [3] in that we do not require problem-dependent preconditioners for e_y and e_y^\top , and from [1] in allowing inexact inverses for e_y and e_y^\top . How we use this approach in our context is explained in more detail in section 3.

Summarizing, the method outlined so far aims at solving the KKT system (1) inexactly and at relatively low numerical cost. In order to globalize the overall optimization algorithm, the inexact solution p can be used as a search direction in a line search context. This technique is used for the results in Section 4 in order to illustrate the performance of our method.

3 Approximation of Jacobian Information

In this section, we describe the construction of geometric multigrid preconditioners \tilde{e}_y and \tilde{e}_y^\top for the linearized and adjoint PDE problems involving e_y and e_y^\top , respectively, which appear in each application of the preconditioner P_2 . The following requirements must be met by the preconditioner in order to fit into the framework set forth in the introduction:

1. It should be constructed with little involvement by the user, who so far only needed to specify the objective f and the discretized PDE constraint e .
2. It should not need any access to elements of the constraint Jacobian e_y but make do with matrix–vector products alone.
3. Its accuracy should be controllable.

The geometric multigrid method meets these requirements under some precautions: In order to define meaningful prolongation and restriction operators, the ordering of the state variables y in the vector of unknowns must be known. In addition, the classical Gauss-Seidel or Jacobi smoothers cannot be used since they require the explicit knowledge of the diagonal elements of e_y , which are expensive to compute with a standard application of AD techniques. A follow-up paper may address these issues, but presently we confine ourselves to smoothers which work only with Jacobian–vector products such as Richardson’s iteration, see below. Let us denote by $0, 1, \dots, L$, a hierarchy of grid levels for the PDE at hand, where L refers to the finest level (at which the original PDE constraint e is given) and where each level is obtained by global refinement of the previous one. For simplicity, we describe the application of the multigrid preconditioner \tilde{e}_y^{-1} in the two-grid case $L = 1$ and for the V-cycle only. Given the right hand side b_1 on grid level 1, we have

$$\tilde{e}_y^{-1}b_1 = S_1^{\nu_2}((S_1^{\nu_1}(0; b_1) + P_0^1 A_0^{-1} R_1^0(b - A_1 S_1^{\nu_1}(0; b_1))); b_1) \quad (3)$$

where R_{l+1}^l and P_l^{l+1} denote restriction and prolongation operators between grid levels l and $l + 1$. In the example problem below, which is discretized by finite differences, we use linear prolongation and full weighted restriction operators [7]. Note that given the ordering of the state variables in the vector of optimization variables, these operators can be constructed automatically. In addition, $S_l^\nu(y_l; b_l)$ is the ν -fold application of a smoothing operator on level l with initial guess y_l and right hand side b_l . In the case of Richardson’s smoothing, we have

$$S_l^\nu(y_l, b_l) = (I - \omega_l A_l)^\nu y_l + \left[\sum_{j=0}^{\nu-1} (I - \omega_l A_l)^j \right] \omega_l b_l. \quad (4)$$

The coarse grid matrices A_l , $l > 0$, i.e., the approximations of e_y on the coarser grid levels, are formed by Galerkin projection of the original constraint Jacobian e_y provided by AD, i.e., $A_l = R_{l+1}^l \cdots R_L^{L-1} A_L P_{L-1}^L \cdots P_l^{l+1}$, and $A_L = e_y$. The damping factors ω_l can be chosen on each grid level l individually. Upon the first application of the preconditioner \tilde{e}_y , they are determined by analyzing some information on the eigenvalues of A_l and are then kept constant until moving to the next iterate with a different KKT system. The coarse grid matrix A_0 is of small size and can be assembled at reasonable cost by AD techniques and then factored to apply its inverse for the exact coarse grid correction on the coarsest level. Note that this factorization is required only once for every iteration in the outer Newton loop and is then used every time the preconditioner is applied.

The preconditioner for the transpose e_y^\top is constructed in the same way by replacing the matrices A_l in (3) and (4) by A_l^\top . Evaluating the matrix–vector products involving A_l and A_l^\top require the application of the forward and reverse modes of AD, respectively. In order to retain symmetry of the preconditioner P_2 , \tilde{e}_y^\top must be the transpose of \tilde{e}_y . This is guaranteed if in the multigrid cycle for \tilde{e}_y^\top , the number of pre- and post-smoothing steps are switched in comparison to \tilde{e}_y , and if the restriction operators are multiples of the transposed prolongation operators, as is the case for the choice described above. Note that the preconditioner satisfies the requirements 1. and 2. set forth above. In order to control its accuracy, one may simply carry out several V-cycles (or other cycles) in a row, or perform an exact defect correction step not on the coarsest but on a finer level, as reported in the results in the following section.

4 Numerical Example

As a first numerical test for the proposed KKT preconditioner, we choose a semilinear reaction-diffusion model in two spatial dimensions on the unit square Ω . That is, we compute the concentration $y(x)$ of one species that diffuses and undergoes an autocatalytic reaction of second order at the rate $\alpha y(x)^2$. As boundary condition, we use $y|_{\text{boundary}} = 0$. To approach a desired distribution $y_d(x)$ of the species at the steady state we allow a distributed control by adding the species with the rate $\beta u(x)$. This optimization problem can be described by

$$\begin{aligned} \min_{y,u} \quad & \|y(x) - y_d(x)\|_{L^2(\Omega)}^2 + \gamma \|u(x)\|_{L^2(\Omega)}^2 \\ \text{s.t.} \quad & -\Delta y(x) + \alpha y(x)^2 = \beta u(x), \quad \forall x \in (0, 1)^2, \quad y|_{\text{boundary}} = 0. \end{aligned}$$

For our numerical results, we set $\alpha = 20$, $\beta = 40$ and $\gamma = 0.01$. In Table 1, we report on the behavior of the symmetric QMR method without look-ahead [4] for various fine-grid levels. In every iteration of Newton’s method, the KKT system (1) is solved to a relative accuracy of 10^{-6} , i.e., the initial residual is reduced by this factor. As mentioned above, we use $\tilde{H}_R = I$ as a preconditioner for the reduced Hessian H_R . The coarsest grid (level 0) has 5–by–5 grid points, and the grid on level l has $4 \cdot 2^l + 1$ squared. Discretization of the PDE constraint $e(y, u) = 0$ is carried out by standard finite differences. The discretization dimension of the problem increases between consecutive rows in Table 1. The first column shows the number of state variables and the number of total variables (states, adjoints, and controls). As mentioned above, the individual rows correspond to fine grid levels $L \in \{2, 3, 4\}$, respectively. The number of Newton steps remains constant, as is expected from the known mesh independence of Newton’s method. In the next set of columns, the total number of QMR steps and the CPU time in seconds are shown. Each column corresponds to a different choice of the grid level on which the linear

defect correction problem within the multigrid preconditioner for e_y and e_y^\top is solved exactly, by LU decomposition. In the first column, the LU decomposition is carried out on the coarsest level and thus for a 5-by-5 matrix, while in the following columns, the multigrid V-cycle is stopped short before reaching level 0 and the problem involving A_l is solved exactly on the current level l . As l increases, the matrix that needs to be LU-decomposed grows in size, but at the same time, we obtain a more accurate preconditioning of e_y and e_y^\top . In terms of CPU time, the optimal choice for the current example and for the discretization levels shown is $l = 2$. A discussion of the question on how to choose this coarse grid level in general, based on complexity estimates, is also postponed to a forthcoming paper.

states	fine		Newton	LU on grid #				
	total	grid		0	1	2	3	
289	2		7	964	927	184		qmr
867				9	5	2		cpu
1089	3		7	1193	972	1053	185	qmr
3267				71	44	38	118	cpu
4225	4		7	1320	1135	1256	1124	qmr
12675				626	428	351	468	cpu

Table 1 Convergence results for the symmetric QMR version with our multigrid preconditioner.

5 Conclusion

In the present work, we have developed a preconditioner for KKT systems which arise in stationary PDE-constrained optimization. Our work is based on previous findings in [1, 3], but in extension we use Algorithmic Differentiation (AD) to provide all required derivatives and we wish to develop preconditioners for the linearized and adjoint PDE subproblems which are not problem-dependent but instead can be constructed automatically. We have suggested to use the geometric multigrid approach for this purpose, with an adaptive Richardson smoother. Details are postponed to a forthcoming paper.

In the future, the following open issues need to be addressed: What is the optimal choice for the coarse grid on which the linearized and adjoint PDEs are solved exactly or possibly inexactly? How to find cheap and yet good preconditioners for the reduced Hessian H_R ? How to automatically construct multigrid preconditioners for time-dependent PDEs?

One needs to keep in mind that in order to use our method, the user needs to specify only the objective and the discretized PDE constraint function (on the finest grid). This presents some advantage over problem-dependent preconditioners which, once devised, are of course expected to perform somewhat better. The first feasibility study presented in this paper shows some promising results which lead us to pursue further details of this method in a forthcoming paper.

References

- [1] A. Battermann and E. Sachs. An indefinite preconditioner for KKT systems arising in optimal control problems. Technical Report, University of Trier. To appear.
- [2] M. Benzi, G. Golub, and J. Liesen: Numerical solution of saddle point problems. *Acta Numerica* 14, pp. 1-137, 2005
- [3] G. Biros and O. Ghattas: Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM Journal on Scientific Computing*, to appear.
- [4] R. Freund and N. Nachtigal: An implementation of the QMR method based on coupled two-term recurrences. *SIAM J. Sci. Comput.* 15, pp. 313–337 (1994).
- [5] A. Griewank: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. *Frontiers in Appl. Math.* 19, SIAM, (2000).
- [6] W. Hackbusch: *Multi-grid methods and applications*, Series in Computational Mathematics 4, Springer (1985).
- [7] U. Trottenberg, C. Oosterlee, and A. Schüller: *Multigrid*. Academic Press (2001).
- [8] A. Wächter and L. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Technical Report RC 23149, IBM T. J. Watson Research Center, Yorktown, USA, 2004. To appear in *Mathematical Programming*.
- [9] R. Waltz and J. Nocedal. KNITRO user’s manual. Technical Report OTC 05/2003, Optimization Technology Center, Northwestern University, Evanston, IL 60208, USA, 2003.