Two Steps Towards Automating Efficient Solution of Inverse Problems

Kent-Andre Mardal*, Bjørn Fredrik Nielsen†and Martin Alnæs‡

January 31, 2008

We have combined symbolic mathematics with code generation to create a user–friendly environment for specifying finite element methods. The motivation behind this approach is to automate the boring and error-prone task of implementing variational forms of PDEs, which is a cornerstone when developing finite element simulators. By employing a symbolic engine in a high–level language we allow the user to specify the weak form of the PDE in an abstract and user–friendly format. Furthermore, the symbolic framework allows us to do certain calculations like differentiation automatically. In addition to the symbolic framework, we use a high–level library for expressing linear algebra algorithms in terms of block matrices and block preconditioners.

Our efforts have resulted in the open source software package SyFi [3], which is part of the FEniCS project [9]. SyFi stands for Symbolic Finite elements and is implemented in C++ and Python. SyFi is built on top of the symbolic C++ library GiNaC [5] and uses its Python interface Swiginac [17]. In addition to SyFi, we employ the in-house code PyCC [16] for block matrices and preconditioners.

SyFi is largely divided in two parts, a kernel implementing finite elements etc. and a form compiler. The SyFi form compiler (SFC) takes as input variational forms and functionals on the following form (see also the FEniCS Form Compiler (FFC) [10, 11, 12, 13] and UFC (Unified Form-assembly Code) [1, 2]),

$$a_\Omega(u^0, \dots, u^{n-1}; w^0, \dots, w^{m-1}) \rightarrow \mathbb{R}, \tag{1}$$

where $0 \leq n \leq 2$, u^0 is the test function, u^1 is the trial function and $w^0, \dots, w^{m-1}$ are the coefficient functions (or prescribed functions), $u^k \in U_k$ and $w^l \in W^l$ where U^k and W^l are finite element spaces. The differentiation of variational

*Department of Scientific Computing, Simula Research Laboratory and Department of Informatics, University of Oslo, P.O. Box 134, NO-1325 Lysaker, Norway, e-mail: kent-and@simula.no

†Department of Scientific Computing, Simula Research Laboratory and Department of Informatics, University of Oslo, P.O. Box 134, NO-1325 Lysaker, Norway, e-mail: bjornn@simula.no

‡Department of Scientific Computing, Simula Research Laboratory P.O. Box 134, NO-1325 Lysaker, Norway, e-mail: martinal@simula.no

forms and functionals is performed as follows. Let $\Pi^j : W^j \to \mathbb{R}^{n_j}$, where n_j is the dimension of W^j, be the mapping which gives the degrees of freedom associated with W^j. Hence, $\{\Pi^j u\}_i = u_i^j$ where u_i^j is the i'th degree of freedom in space W^j. Then we may associate a tensor function

$$a^N : \mathbb{R}^{n_0} \times \ldots \times \mathbb{R}^{n_{m-1}} \to \mathbb{R}^{d_0} \times \ldots \times \mathbb{R}^{d_{n-1}}$$

with a_Ω as follows. For $u_{i_0}^0, \ldots, u_{i_{n-1}}^{n-1}$ in $U^0, \ldots, U^{n-1}$ the component $i_0, \ldots, i_{n-1}$ of the tensor function a^N is

$$a_{i_0,\ldots,i_{n-1}}^N(\Pi^0 w^0, \ldots, \Pi^{m-1} w^{m-1}) = a_\Omega(u_{i_0}^0, \ldots, u_{i_{n-1}}^{n-1}; w^0, \ldots, w^{m-1}) \in \mathbb{R}. \quad (2)$$

Since a^N is a tensor function, taking values in $\mathbb{R}^{n_0} \times \ldots \times \mathbb{R}^{n_{m-1}}$, it can be differentiated in a standard way. This differentiation is trivial with a symbolic engine, e.g.,

$$\frac{\partial}{\partial w_i^j} a_\Omega^N(\Pi^0 w^0, \ldots, \Pi^{m-1} w^{m-1}) \quad (3)$$

is the derivative with respect to the i'th degree of freedom in the coefficient space W^j. By running over all i, we obtain the derivative with respect to W^j. This corresponds directly to the Fréchet derivative with respect to the coefficient function space W^j. This differentiation can be obtained with SyFi.

In the following we will consider systems of linear PDEs arising from two specific optimal control problems with linear and elliptic PDE constraints. These are the model problem studied by Borzi, Kunisch and Kwak [7] and an inverse problem arising in connection with electrocardiography [15]. Each of the entries in the systems is a variational form for which the code for the element tensor calculations is generated by SyFi. After the assembly of the sub matrices and sub vectors they are put into the block matrix and preconditioner framework in PyCC.

To describe both model problems we present them in a common abstract setting, even though we need to be concrete when implementing these model problems in SyFi and PyCC. To this end, let U_1, U_2 and U_3 be Hilbert spaces with inner products $(\cdot,\cdot)_{U_1}$, $(\cdot,\cdot)_{U_2}$, $(\cdot,\cdot)_{U_3}$, norms $\|\cdot\|_{U_1}$, $\|\cdot\|_{U_2}$, $\|\cdot\|_{U_3}$ and dual spaces U_1', U_2' and U_3'. Our parameter identification problem can be written on the form

$$\min_{v \in U_1} \left\{ \frac{1}{2} \|Tu - d\|_{U_3}^2 + \frac{1}{2} \alpha \|v - v_{\text{prior}}\|_{U_1}^2 \right\} \quad (4)$$

subject to

$$Au = -Bv + g \quad \text{(state equation)}, \quad (5)$$

where

$$\begin{aligned} &A : U_2 \to U_2', \\ &B : U_1 \to U_2', \\ &T : U_2 \to U_3 \quad \text{(observation operator)} \end{aligned}$$

are bounded linear operators, $d \in U_3$ and $g \in U_2'$ are given quantities, and $\alpha > 0$ is a regularization parameter. Notice that, from an implementational point of view, the above (and below) operators are represented by their weak forms. The associated Lagrangian L_α reads

$$\begin{aligned} L_\alpha(v,u,w) &= \frac{1}{2}\|Tu-d\|_{U_3}^2 + \frac{1}{2}\alpha\|v-v_{\text{prior}}\|_{U_1}^2 \\ &\quad +\langle Au,w\rangle + \langle Bv,w\rangle - \langle g,w\rangle \end{aligned}$$

for $v \in U_1$ and $u, w \in U_2$, with Fréchet derivatives

$$\begin{aligned} \langle \frac{\partial L_\alpha}{\partial v}, \phi\rangle &= \alpha(v-v_{\text{prior}},\phi)_{U_1} + \langle B\phi, w\rangle, \quad \phi \in U_1, \\ \langle \frac{\partial L_\alpha}{\partial u}, \phi\rangle &= (Tu-d, T\phi)_{U_3} + \langle A\phi, w\rangle \quad \phi \in U_2, \\ \langle \frac{\partial L_\alpha}{\partial w}, \phi\rangle &= \langle Au, \phi\rangle + \langle Bv,\phi\rangle - \langle g,\phi\rangle \quad \phi \in U_2. \end{aligned}$$

From the first order necessary condition

$$\frac{\partial L_\alpha}{\partial v} = 0, \quad \frac{\partial L_\alpha}{\partial u} = 0, \quad \frac{\partial L_\alpha}{\partial w} = 0$$

we can write the linear system to be solved as: Find $(v,u,w) \in U_1 \times U_2 \times U_2$ such that

$$\begin{bmatrix} \alpha L & 0 & B' \\ 0 & K & A' \\ B & A & 0 \end{bmatrix} \begin{bmatrix} v \\ u \\ w \end{bmatrix} = \begin{bmatrix} \alpha L v_{\text{prior}} \\ Qd \\ g \end{bmatrix}, \tag{6}$$

where A' and B' denote the dual operators of A and B, respectively, and

$$L : U_1 \to U_1', \quad v \to (v,\phi)_{U_1} \ \forall \phi \in U_1, \tag{7}$$

$$K : U_2 \to U_2', \quad u \to (Tu, T\phi)_{U_3} \ \forall \phi \in U_2, \tag{8}$$

$$Q : U_3 \to U_2', \quad d \to (d, T\phi)_{U_3} \ \forall \phi \in U_2. \tag{9}$$

The linear system (6) is a saddle-point problem. Preconditioners for saddle-point problems have been studied extensively the last decades, see e.g. [6] for a review. Our work is based on the approach suggested in [4, 14], where the saddle point problem is considered as an isomorphism between an appropriate Hilbert space and its dual space, provided that the Babuška-Brezzi conditions [8] are satisfied. By letting the preconditioner be an isomorphism mapping the dual space back to the Hilbert space, the preconditioned system becomes well-conditioned, and in the discrete case one obtains conditions numbers which can be bounded independently of the mesh parameter h.

The problem studied in this abstract is different from those analyzed in [4, 14] because (4)-(5) typically is ill-posed for $\alpha = 0$. Furthermore, even though the regularized problem ($\alpha > 0$) is well-posed, the Babuška-Brezzi conditions will involve constants that depend on α. This dependency of the regularization parameter causes the condition number of the associated matrix to increase as the regularize parameter decreases towards zero. However, we will demonstrate that efficient solution methods can be constructed in spite of this dependency. More specifically, in the severely ill-posed case, it turns out that the number of iterations needed by the preconditioned minimal residual method can not grow any faster than of order $O((\ln(\alpha))^2)$, provided that the convergence is measured in the energy norm. Moreover, numerical experiments indicate that this theoretical worst case scenario bound is rather pessimistic.

Our preconditioner is block diagonal where each block is a standard "off-the-shelves" preconditioner. Since the resulting preconditioner is symmetric and positive, the minimal residual method is used. Our scheme is implemented in PyCC by employing block matrices and block preconditioners. The preconditioner can be written:

$$\mathcal{B}_\alpha^{-1} = \begin{bmatrix} \alpha L & 0 & 0 \\ 0 & \alpha A + K & 0 \\ 0 & 0 & \frac{1}{\alpha} A \end{bmatrix}. \tag{10}$$

We will present both theoretical results and a number of numerical experiments addressing both the model problem studied by Borzi, Kunisch and Kwak [7] and an inverse problem arising in connection with electrocardiography [15]. We will further explain how differentiation of variational forms and functionals can be done in SyFi.

References

[1] M. Alnæs, H.-P. Langtangen, A.Logg, K.-A. Mardal, and O. Skavhaug, *UFC Specification and User Manual*, 2007. URL: urlhttp//www.fenics.org/ufc/.

[2] M. Alnæs, H.-P. Langtangen, A. Logg, K.-A. Mardal, and O. Skavhaug, *Unified framework for finite element assembly*, 2008. Submitted to journal.

[3] M. Alnæs and K.-A. Mardal, *SyFi*, 2006. URL: `http://www.fenics.org/syfi/`.

[4] D. Arnold, R. Falk, and R. Winther, *Preconditioning discrete approximations of the reissner–mindlin plate model*, Mathematical Modeling and Numerical Analysis, 31 (1997), pp. 517–557.

[5] C. Bauer, C. Dams, A. Frink, V. V. Kisil, R. Kreckel, A. Sheplyakov, and J. Vollinga, *GiNaC*, 2006. http://www.ginac.de.

[6] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[7] A. Borzì, K. Kunisch, and D. Y. Kwak, *Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system*, SIAM Journal on Control and Optimization, 41 (2003), pp. 1477–1497.

[8] F. Brezzi, *On the existence, uniqueness and approximation of saddle–point problems arising from lagrangian multipliers*, RAIRO numerical analysis, 8 (1974), pp. 129–151.

[9] FEniCS, *FEniCS project*. URL: urlhttp//www.fenics.org/, 2007.

[10] R. C. Kirby and A. Logg, *A compiler for variational forms*, ACM Transactions on Mathematical Software, 32 (2006), pp. 417–444.

[11] ——, *Efficient compilation of a class of variational forms*, ACM Transactions on Mathematical Software, 33 (2007).

[12] A. Logg, *FFC user manual*, 2007. URL urlhttp//www.fenics.org/ffc/.

[13] A. Logg et al., *FFC*. URL: urlhttp//www.fenics.org/ffc/, 2006.

[14] K.-A. Mardal and R. Winther, *Uniform preconditioners for the time dependent stokes problem.*, Numerische Mathematik, 98 (2004), pp. 305–327.

[15] B. F. Nielsen, X. Cai, and O. M. Lysaker, *On the possibility for computing the transmembrane potential in the heart with a one shot method; an inverse problem*, Mathematical Biosciences, 210 (2007), pp. 523–553.

[16] *PyCC*, 2007. Software framework under development. URL: `http://www.simula.no/pycc`.

[17] O. Skavhaug and O. Certik, *Swiginac*, 2006. http://swiginac.berlios.de/.