ARTICLE IN PRESS

Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A numerical method to compute derivatives of functions of large complex matrices and its application to the overlap Dirac operator at finite chemical potential

Matthias Puhr *, Pavel Buividovich

Institute of Theoretical Physics, Regensburg University, D-93053 Regensburg, Germany

ARTICLE INFO

Article history:
Received 2 May 2016
Received in revised form
12 July 2016
Accepted 11 August 2016
Available online xxxx

Keywords:
Chiral lattice fermions
Finite density QCD
Krylov subspace methods
Numerical differentiation

ABSTRACT

We present a method for the numerical calculation of derivatives of functions of general complex matrices. The method can be used in combination with any algorithm that evaluates or approximates the desired matrix function, in particular with implicit Krylov–Ritz-type approximations. An important use case for the method is the evaluation of the overlap Dirac operator in lattice Quantum Chromodynamics (QCD) at finite chemical potential, which requires the application of the sign function of a non-Hermitian matrix to some source vector. While the sign function of non-Hermitian matrices in practice cannot be efficiently approximated with source-independent polynomials or rational functions, sufficiently good approximating polynomials can still be constructed for each particular source vector. Our method allows for an efficient calculation of the derivatives of such implicit approximations with respect to the gauge field or other external parameters, which is necessary for the calculation of conserved lattice currents or the fermionic force in Hybrid Monte-Carlo or Langevin simulations. We also give an explicit deflation prescription for the case when one knows several eigenvalues and eigenvectors of the matrix being the argument of the differentiated function. We test the method for the two-sided Lanczos approximation of the finite-density overlap Dirac operator on realistic $SU(3)$ gauge field configurations on lattices with sizes as large as 14×14^3 and 6×18^3.

1. Introduction

Over the last decade matrix valued functions of matrices have become an essential tool in a variety of sub-fields of science and engineering [1]. An important application for matrix functions in the field of lattice QCD is the so-called overlap Dirac operator, which is a discretisation of the Dirac operator that respects the properly defined lattice chiral symmetry (Ginsparg–Wilson relations) and is free of doublers. Therefore the overlap Dirac operator is well suited for the non-perturbative study of strongly interacting chiral fermions. At finite chemical potential μ the overlap Dirac operator is defined as [2]

$$D_{\text{ov}} := \frac{1}{a}\left(\mathbb{1} + \gamma_5 \operatorname{sgn}\left[H(\mu)\right]\right), \tag{1}$$

where $H(\mu) := \gamma_5 D_{\text{w}}(\mu)$, $D_{\text{w}}(\mu)$ is the Wilson–Dirac operator at chemical potential μ, sgn is the matrix sign function and a stands for the lattice spacing. An explicit form of $D_{\text{w}}(\mu)$ is given in Appendix A.

At finite chemical potential H is a non-Hermitian matrix with complex eigenvalues. Since the size of the linear space on which H is defined is typically very large ($n \sim 10^4 \ldots 10^7$), it is not feasible to evaluate the matrix sign function exactly. While at $\mu = 0$ one can efficiently approximate the sign function of the Hermitian operator H by polynomials or rational functions [3–6], at nonzero μ the operator H becomes non-Hermitian and such approximations typically become inefficient. However, having in mind that in practice the sign function $\operatorname{sgn}[H(\mu)]$ is applied to some source vector, one can still construct an efficient polynomial approximation for each particular source vector by using Krylov subspace methods, such as Krylov–Ritz-type approximations. One of the practical Krylov–Ritz-type

* Corresponding author.
E-mail addresses: Matthias.Puhr@physik.uni-regensburg.de (M. Puhr), Pavel.Buividovich@physik.uni-regensburg.de (P. Buividovich).

http://dx.doi.org/10.1016/j.cpc.2016.08.011
0010-4655/

approximations which are suitable for the finite-density overlap Dirac operator is the two-sided Lanczos (TSL) algorithm, developed in [7,8]. The efficiency of the TSL approximation can be further improved by using a nested version of the algorithm [9].

Many practical tasks within lattice QCD simulations require the calculation of the derivatives of the lattice Dirac operator with respect to the gauge fields or some other external parameters. For example, conserved lattice vector currents and fermionic force terms in Hybrid Monte-Carlo simulations involve the derivatives of the Dirac operator with respect to Abelian or non-Abelian gauge fields. Also, electric charge susceptibilities which are used to quantify electric charge fluctuations in quark–gluon plasma involve derivatives of the Dirac operator with respect to the chemical potential.

While explicit expressions for the derivatives of source-independent approximations of matrix functions are well known and are routinely used in practical lattice QCD simulations, differentiating the implicit source-dependent approximation appears to be a more subtle problem. In principle the algorithms for taking numerical derivatives of scalar functions, like the finite difference method or algorithmic differentiation, can be generalised to matrix functions and to matrix function approximation algorithms. It is easy to combine an approximation with the finite difference method, but finite differences are very sensitive to round-off errors and it is often not possible to reach the desired precision in the derivative using this method. For algorithmic differentiation the situation is more complicated. Depending on the approximation method used it might not be immediately clear how to apply algorithmic differentiation. Even if algorithmic differentiation can be implemented for the approximation method this might lead to a numerically unstable algorithm. Such a behaviour was observed when the TSL approximation was used in conjunction with algorithmic differentiation [10].

In this paper, we propose and test a practical numerically stable method which makes it possible to compute derivatives of implicit approximations of matrix functions to high precision. The main motivation for this work is the need to take derivatives of the overlap Dirac operator in order to compute conserved currents on the lattice.

The structure of the paper is the following: in Section 2 we state some general theorems about matrix functions and their derivatives, which provide the basis for our numerical method. In Section 3 we discuss how the calculation of the derivatives of matrix functions can be made more efficient by deflating a number of small eigenvalues of the matrix which is the argument of the function being differentiated. The deflation is designed with the matrix sign function and the TSL approximation in mind. Nevertheless we want to emphasise that the method is very general and can be applied to other matrix functions and different matrix function approximation schemes. In Section 4 we demonstrate how the method can be used in practice. First we discuss the efficiency and convergence of the TSL approximation. After that, as a test case, we compute $U(1)$ lattice vector currents, which involve the derivatives of the overlap Dirac operator with respect to background Abelian gauge fields, and demonstrate that they are conserved. Finally we summarise and discuss the advantages and disadvantages of our method in Section 5. Detailed calculations and derivations as well as a pseudo code implementation of the method can be found in the Appendices.

2. Matrix functions and numerical evaluation of their derivatives

For completeness we start this Section with a brief review of matrix functions. Let the function $f : \mathbb{C} \to \mathbb{C}$ be defined on the spectrum of a matrix $A \in \mathbb{C}^{n\times n}$. There exist several equivalent ways to define the generalisation of f to a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ [1,11]. For the purpose of this paper the most useful definition is via the Jordan canonical form. A well known Theorem states that any matrix $A \in \mathbb{C}^{n\times n}$ can be written in the Jordan canonical form

$$X^{-1}AX = J = \operatorname{diag}(J_1, J_2, \ldots, J_k), \tag{2}$$

where every Jordan block J_i corresponds to an eigenvalue λ_i of A and has the form

$$J_i = J_i(\lambda_i) = \begin{pmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \lambda_i & 1 \\ 0 & \cdots & 0 & 0 & \lambda_i \end{pmatrix} \in \mathbb{C}^{m_i\times m_i}, \tag{3}$$

with $m_1 + m_2 + \cdots + m_k = n$. The Jordan matrix J is unique up to permutations of the blocks but the transformation matrix X is not. Using the Jordan canonical form the matrix function can be defined as [1,11]

$$f(A) := Xf(J)X^{-1} = X\operatorname{diag}(f(J_i))X^{-1}. \tag{4}$$

The function of the Jordan blocks is given by

$$f(J_i) := \begin{pmatrix} f(\lambda_i) & f'(\lambda_i) & \ldots & \dfrac{f^{(m_i-1)}(\lambda_i)}{(m_i-1)!} \\ 0 & f(\lambda_i) & \ddots & \vdots \\ \vdots & \ddots & \ddots & f'(\lambda_i) \\ 0 & \cdots & 0 & f(\lambda_i) \end{pmatrix} \in \mathbb{C}^{m_i\times m_i}. \tag{5}$$

Note that this definition requires the existence of the derivatives $f^{(m_i-1)}(\lambda_i)$ for $i = 1, \ldots, k$. If A is diagonalisable every Jordan block has size one and Eq. (5) reduces to the so-called spectral form

$$f(A) = X\operatorname{diag}(f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_n))X^{-1}, \tag{6}$$

which does not depend on the derivatives of f. For instance, the matrix sign function $\operatorname{sgn}[H(\mu)]$ in (1) is defined by $\operatorname{sgn}(\lambda_i) = \operatorname{sgn}(\operatorname{Re}\lambda_i)$.

Finding the Jordan normal form (or the spectral decomposition) of a matrix is a computationally expensive task and takes $\mathcal{O}(n^3)$ operations. For large matrices it is therefore not feasible to compute the matrix function exactly. In practical calculations it is often sufficient

Please cite this article in press as: M. Puhr, P. Buividovich, A numerical method to compute derivatives of functions of large complex matrices and its application to the overlap Dirac operator at finite chemical potential, Computer Physics Communications (2016), http://dx.doi.org/10.1016/j.cpc.2016.08.011