Accepted Manuscript

Extending an iterative orthogonal projection method towards least-squares solutions

Noreen Jamil, Farhaan Mirza, M. Asif Naeem, Nilufar Baghaei

PII: S0377-0427(18)30104-3
DOI: https://doi.org/10.1016/j.cam.2018.02.025
Reference: CAM 11531

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 26 March 2017
Revised date : 9 October 2017

Please cite this article as: N. Jamil, F. Mirza, M.A. Naeem, N. Baghaei, Extending an iterative orthogonal projection method towards least-squares solutions, *Journal of Computational and Applied Mathematics* (2018), https://doi.org/10.1016/j.cam.2018.02.025

Extending an iterative orthogonal projection method towards least-squares solutions

Noreen Jamil[3], Farhaan Mirza[1], M. Asif Naeem[2], Nilufar Baghaei[3]

[1,2] *School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology*
Private Bag 92006, Auckland, New Zealand.

[3] *Department of Computer Science*
Unitec Institute of Technology.

1. Introduction

The Kaczmarz method [1] is an iterative method for solving large systems of equations that projects iterate orthogonally onto the solution space of each equation. In contrast to direct methods such as Gaussian elimination or QR-factorization, this algorithm is efficient for problems with sparse matrices, as encountered in constraint-based user interface (UI) layout specifications.

Starting with an initial guess, the Kaczmarz algorithm selects a row index of the matrix and projects the current iterate onto the solution space of that equation, refining the solution until a sufficient precision is reached. The Kaczmarz algorithm does not need any pivot assignment, and it is ideal for highly over-determined linear systems, as in many linear problems including the constraint-based UI layout. In our paper [2], we proposed extensions to the linear relaxation method to deal with this issue. Our analysis showed that the Kaczmarz method is more suitable for solving non-square matrices as they occur in UI layout.

Despite Kaczmarz's efficiency for sparse systems, it is seldom used for generating constraint-based UI layouts. The reasons are as follows. First, constraint-based UI layout contains linear *equality* and *inequality constraints* for specifying relationships among objects such as "inside", "above", "below", "left-of",

[1] farhaan.mirza@aut.ac.nz
[2] mnaeem@aut.ac.nz
[3] nbaghaei@unitec.ac.nz, njamil@unitec.ac.nz