Parallel Proper Orthogonal Decomposition Study and Implementation

Govind Gopakumar

Indian Institute of Technology, Kanpur

Roll number 11284

November 13, 2014

Overview

Proper Orthogonal Decomposition

- The Proper Orthogonal Decomposition - famous statistical procedure.
- Attempts to decompose a set of data into a different coordinate system, with the aim of compressing the data variance.
- Given a set of M x N data (M data points, N variables), it is possible to reduce to a set of M x P ($P \leq N$) with very low loss in accuracy.

Theory behind the POD

- Orthogonal linear transformation of dataset to new coordinate axis.
- New coordinate system such that first coordinate axis holds greatest variance of data, second holds second greatest, and so on.
- Projection of data is uncorrelated, and we can choose to ignore the latter axes.

Applications of POD

- In Computer Science, Statistics, and Data Analysis - Used for reducing dimensionality.
- Chaotic systems dynamics - Reduced order modelling enables easier computation of accurate solution.
- In Aerodynamics, POD helps characterise flows. POD modes can be used to generate flow models, up to reasonable accuracy.

Computing the POD

Suppose we have a set of data given by $x(t) \in R^n$ with $0 \leq t \leq T$. We seek a projection $P_r : R^n \rightarrow R^r$ that minimises the total error

$$\int_0^T (||x(t) - P_r x(t)||)^2 dt \tag{1}$$

To solve this problem, we introduce the n x n matrix :

$$R = \int_0^T x(t)x(t)^* dt \tag{2}$$

where * denotes the transpose, and find the eigenvalues and eigen vectors of R, given by,

$$R\phi_k = \lambda_k \phi_k, \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \tag{3}$$

Computing the POD

Since R is symmetric, positive-semidefinite, all the eigenvalues λ_k are real and non-negative. The eigen values may be chosen to be orthonormal. This gives us an optimal subspace spanned by $\phi_1, \ldots, \phi_r$ and the optimal projection given by:

$$P_r = \sum_{k=1}^{r} \phi_k \phi_k^* \tag{4}$$

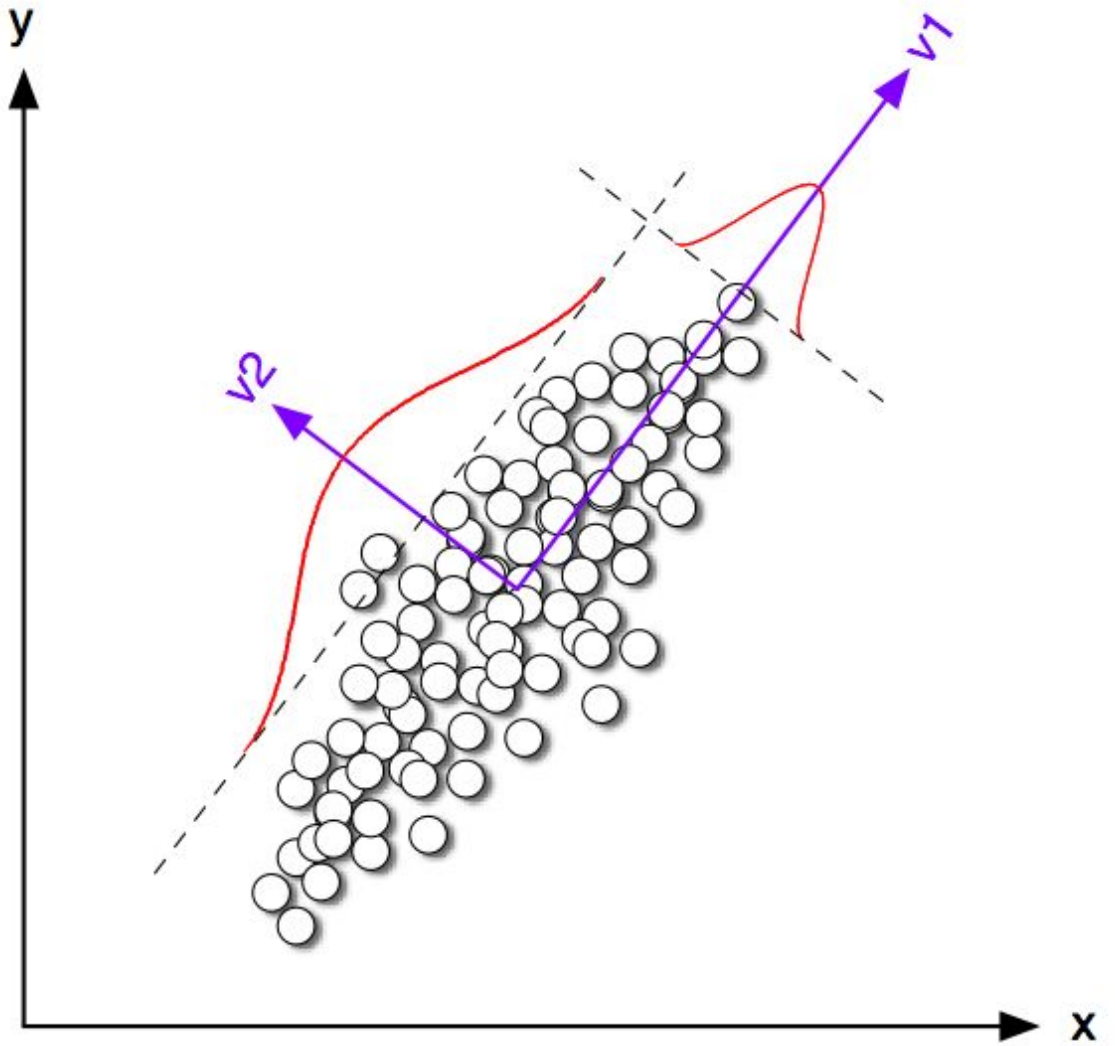
The vectors ϕ_k are called POD modes.

Eigenvalue based POD

Easy way to understand in terms of data :

- The matrix we introduce captures the covariance of the data matrix.
- Computing the eigenvector decomposition of the matrix will capture the maximum variance.
- Ordering the eigenvectors by eigenvalues will order in decreasing variance.
- Equivalent to line fitting and dropping off-line terms in 2D.

POD illustration



Overview of CUDA

CUDA : Compute Device Unified Architecture - This is NVIDIA corp.'s proprietary platform for parallel computation.
Advantages of CUDA over traditional parallel processing:

- Hundreds of compute cores on a single device.
- Massively parallelized architecture - heavily optimized for managing data and processes in parallel.
- Ideal for numerous small tasks.

CUDA in Science

Currently, popularity in scientific community is low, but it is extensively used in :

- Graphics rendering and computations
- Linear algebra routines
- Cryptography

Drawbacks with CUDA

- Accuracy - Double precision is relatively new.
- Power of a single node is relatively low
- Getting started is slightly tough.

Parallelizing the POD

The state of the art is limited to the domain of computer science. The variant of POD, called Principal Component Analysis finds favour, and some work has been done on implementing it on GPUs.

- Work has been done on multi-core implementations of the POD
- GPU processing is still in its infancy - very little work has been done.
- SVD - a related decomposition on GPU is a widely explored area.

Methodology

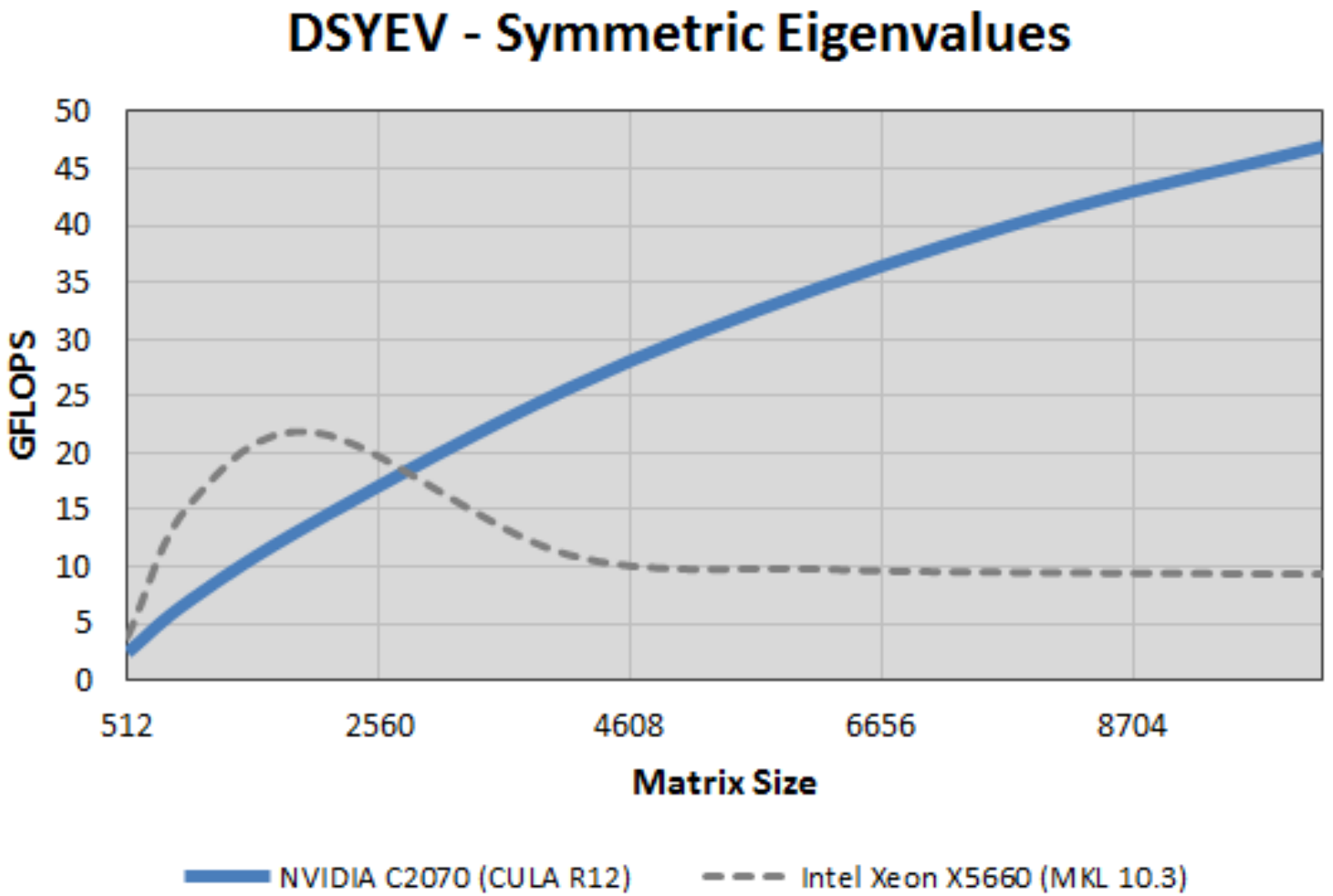
- CULA - A linear algebra library.
- Langauge agnostic - can be used in C++, C, Fortran
- Extensions to MATLAB, Python almost trivial
- Accelerated routines available commercially, at no cost to academics.

Context : Three different kinds of results :

- Standard results published by CULA authors : NVIDIA C2070 GPU vs Intel Xeon X5660 hexa core
- HPCL PC : NVIDIA Quadro K2000 vs Intel Xeon E5-2690 20 core
- Consumer PC : NVIDIA GeForce GTX580 vs Intel Core i7 3770 quad core

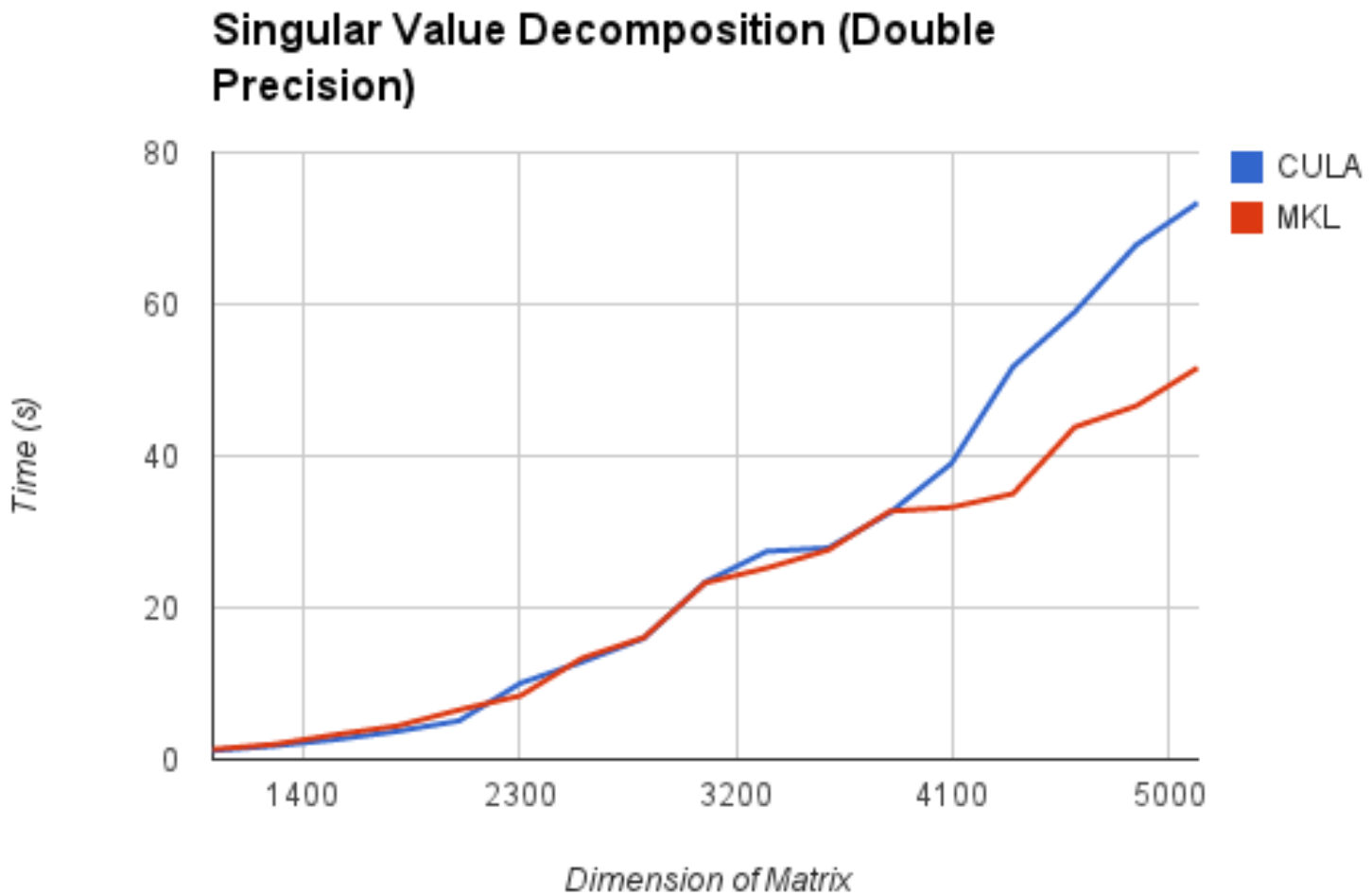
Benchmarking

Published benchmarks for eigenvector decomposition are available :



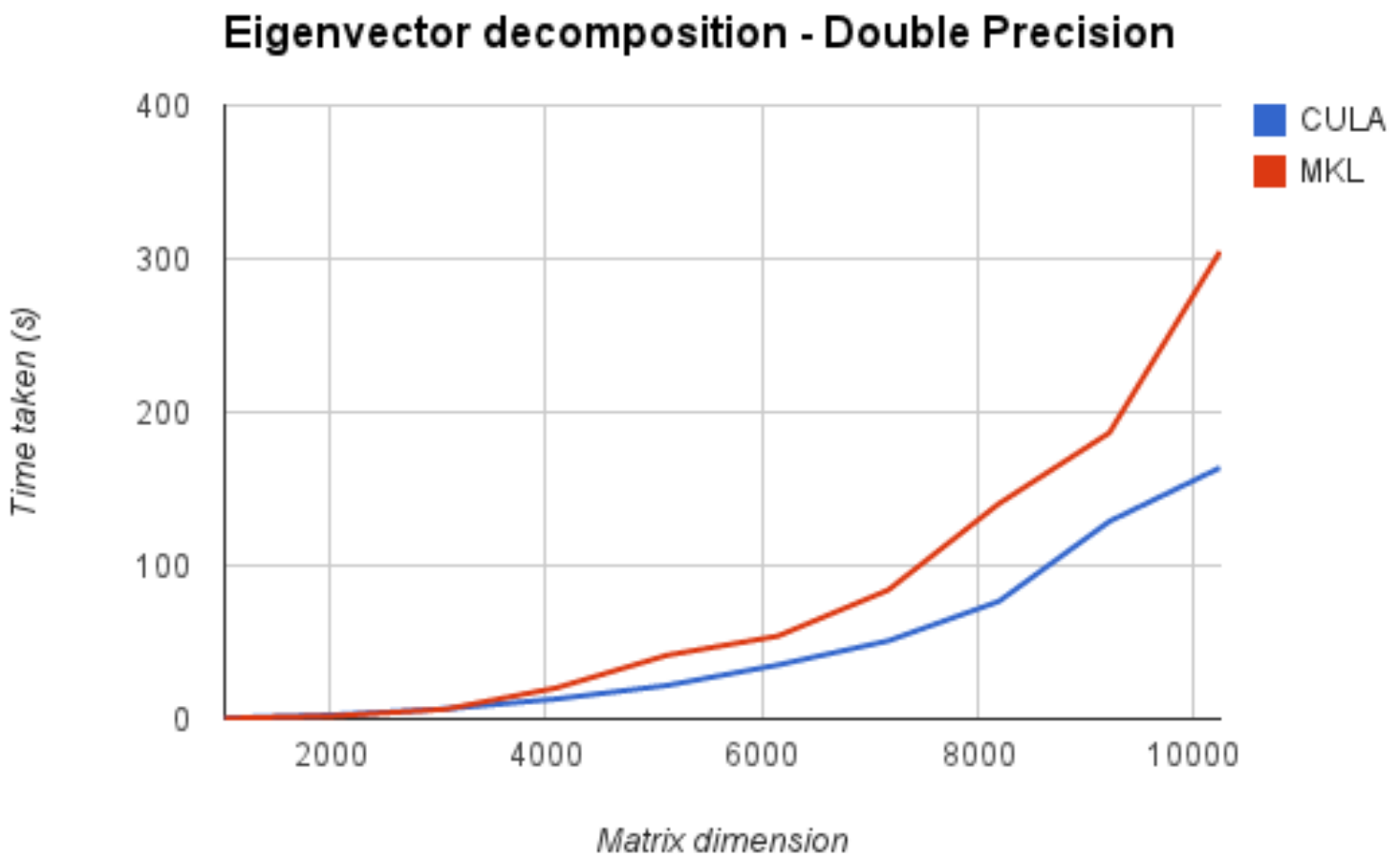
HPCL

Actual results from simulations on HPCL PC:



Consumer PC

Actual results from simulations on consumer PC:



Inference from performance

- Performance of CPUs is easily matched, or even eclipsed by a standard GPU.
- Speed-ups offered can be almost 2x, depending on CPU vs GPU comparison.
- Accuracy is maintained till double precision.
- Results in case of highly polar configurations still show GPUs matching CPU speed.

Concluding remarks

- Compelling use case when applicable
- Specifically, POD speed can be increased up to two fold, with a good GPU based system.
- General linear algebra routines too can be accelerated
- Speed up achieved was using stock configurations, with optimizations, more than double the performance of normal CPUs can be reached.

The End