Computing With Tensors: Potential Applications of Physics-Motivated Mathematics to Computer Science

Martine Ceberio and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
emails mceberio@cs.utep.edu
vladik@utep.edu

1. Why Tensors

- *Modern computing* – main problems include:
  - large amounts of data;
  - long time required to process this data.
- *Similar situation* – 19 century physics:
  - large amounts of data;
  - long time required to process this data.
- *How the problem was solved then:* by using tensors
- *Natural idea:* let us use tensors to solve the problems with modern computing.

2. 19 Century Physics

- Physics starts with measuring and describing the values of different physical quantities.
- It goes on to equations which enable us to predict the values of these quantities.
- A measuring instrument usually returns a single numerical value.
- For some physical quantities (like mass m), the single measured value is sufficient to describe the quantity.
- For other quantities, we need several values.
- *Example:* three components E_x, E_y, and E_z describe the electric field.
- *Example:* to describe the tension inside a solid body, we need values σ_{ij}.

3. Problem and How Tensors Helped

- *19 century:* a separate equation for each component of the field.
- *Result:* equations cumbersome and difficult to solve.
- *Idea:* to describe all the components of a physical field as a single mathematical object:
  - a vector a_i,
  - or, more generally, a tensor a_{ij}, a_{ijk}, ...
- *Result:* simplified equations, faster computations.
- *Originally:* mostly vectors (rank-1 tensors) were used.
- *20 century:*
  - matrices (rank-2 tensors) in quantum physics,
  - higher-order tensors such as the rank-4 curvature tensor R_{ijkl} in relativity theory.

4. From Tensors in Physics to Computing with Tensors

- *Reminder:*
  - 19 century physics encountered a problem of too much data;
  - tensors helped.
- *Modern computing:* suffers from a similar problem.
- *Natural idea:* tensors can help.
- *Two examples* justifying our optimism:
  - modern algorithms for fast multiplication of large matrices;
  - quantum computing.
- *Comment:* detailed descriptions of these examples follow.

5. Modern Algorithm for Multiplying Large Matrices

- *Definition:*

$$\begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & \ldots & b_{1n} \\ \ldots & \ldots & \ldots \\ b_{n1} & \ldots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \ldots & \ldots & \ldots \\ c_{n1} & \ldots & c_{nn} \end{pmatrix};$$

$$c_{ij} = a_{i1} \cdot b_{1j} + \ldots + a_{ik} \cdot b_{kj} + \ldots + a_{in} \cdot b_{nj}.$$

- *Problem:* for large n, no space for both A and B in the fast (cache) memory.
- *Result:* lots of time-consuming data transfers ("cache misses") between different parts of the memory.
- *Solution:* represent each matrix as a matrix of blocks:

$$A = \begin{pmatrix} A_{11} & \ldots & A_{1m} \\ \ldots & \ldots & \ldots \\ A_{m1} & \ldots & A_{mm} \end{pmatrix},$$

$$C_{\alpha\beta} = A_{\alpha 1} \cdot B_{1\beta} + \ldots + A_{\alpha\gamma} \cdot B_{\gamma\beta} + \ldots + A_{\alpha m} \cdot B_{m\beta}.$$

6. Modern Algorithm for Multiplying Large Matrices: Tensor Interpretation

- *Main idea:*
  - we start with a large matrix A of elements a_{ij};
  - we represent it as a matrix consisting of block sub-matrices $A_{\alpha\beta}$.
- *Tensor interpretation:*
  - each element of the original matrix is now represented as
  - an (x, y)-th element of a block $A_{\alpha\beta}$,
  - i.e., as an element of a rank-4 tensor $(A_{\alpha\beta})_{xy}$.
- *Fact:* an increase in rank improves efficiency.
- *Analogy:* a representation of a rank-1 vector as a rank-2 spinor works in relativistic quantum physics.

7. Quantum Computing as Computing with Tensors

- *Classical bit:* a system with two states 0 and 1.
- *Quantum bit (qubit):* superposition principle – we can have states $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$.
- *Probabilities:* $\text{Prob}(0) = |c_0|^2$ and $\text{Prob}(1) = |c_1|^2$, hence $|c_0|^2 + |c_1|^2 = 1$.
- *n-(qu)bit system:* a general state is

  $$c_{0\ldots 00} \cdot |0\ldots 00\rangle + c_{0\ldots 01} \cdot |0\ldots 01\rangle + \ldots + c_{1\ldots 11} \cdot |1\ldots 11\rangle.$$

- *Conclusion:* each state is a tensor $c_{i_1\ldots i_n}$ of rank n.
- *Advantage:* store the entire tensor in only n (qu)bits.
- *Resulting efficiency of quantum computing:*
  - search in an unsorted array of size n in $\sqrt{n}$ time (Grover);
  - factoring large integers in polynomial time (Shor).

8. New Idea: Tensors to Describe Constraints

- *A general constraint* between n real-valued quantities is a subset $S \subseteq R^n$.
- *A natural idea:* represent this subset block-by-block – by enumerating sub-blocks that contain elements of S.
- *Fact:* each block $bi_1 \ldots i_n$ can be described by n indices $i_1, \ldots, i_n$.
- *Result:* we can describe a constraint by a boolean-valued tensor $t_{i_1 \ldots i_n}$ for which:
  - $t_{i_1 \ldots i_n} =$"true" if $b_{i_1, \ldots, i_n} \cap S \neq \emptyset$; and
  - $t_{i_1 \ldots i_n} =$"false" if $b_{i_1, \ldots, i_n} \cap S = \emptyset$.
- *Fact:* processing such constraint-related sets can also be naturally described in tensor terms.
- *Fact:* this speeds up computations.

9. Computing with Tensors Can Also Help Physics

- *So far:* we have shown that tensors can help computing.
- *New idea:* relation between tensors and computing can also help physics.
- *Example:* Kaluza-Klein-type high-dimensional space-time models of modern physics.
- *Einstein's idea:* use "tensors" with integer or circular values.
- *From the mathematical viewpoint:* such "tensors" are unusual.
- *In computer terms:* integer or circular data types are very natural.
- *Fact:* integers and circular data are even more efficient to process than standard real numbers.

10. Remaining Open Problem

- *Example:* tensors naturally appear in an efficient Taylor series approach to uncertainty propagation.
- *Detail:* the dependence of the result y on the inputs $x_1, \ldots, x_n$ is approximated by the Taylor series:

$$y = c_0 + \sum_{i=1}^{n} c_i \cdot x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_i \cdot x_j + \ldots$$

- *Specifics:* the resulting tensors $c_{i_1 \ldots i_r}$ are symmetric: $c_{i_1 \ldots i_r} = c_{\pi(i_1) \ldots \pi(i_r)}$ for each permutation π.
- *Result:* the standard computer representation leads to a $r!$ duplication.
- *Problem:* how to decrease this duplication.

11. Acknowledgments

This work was supported in part:

- by NSF grant HRD-0734825 and
- by Grant 1 T36 GM078000-01 from the National Institutes of Health.

The authors are thankful to Lenore Mullin for her encouragement.