

A Divide-and-Conquer Algorithm for Blade Factorization

Spencer T. Parkin

August 24, 2012

Abstract

The divide-and-conquer approach is taken to the problem of factoring blades in a geometric algebra.

Let \mathbb{G} denote a geometric algebra. Then for any non-zero blade $A \in \mathbb{G}$ of grade m , if we can find a set $\{a_k\}_{k=1}^m$ of m linearly independent vectors such that for all $v \in \{a_k\}_{k=1}^m$, we have $v \wedge A = 0$, then we have found, up to scale, a factorization of A . That is, for some scalar $\lambda \in \mathbb{R}$, the set of real numbers, we have

$$A = \lambda \bigwedge_{k=1}^m a_k.$$

A fast algorithm for finding such a factorization, and one that is numerically stable, can be found in (ref). Here we consider a divide-and-conquer approach to the problem. Doing so, we change the problem from that of directly finding a factorization of A to that of cutting the blade A in half. That is, find any two blades B and C , each of roughly the same grade, such that $A = B \wedge C$ or such that $B \wedge C$ is a non-zero scalar multiple of A . This is worth considering, provided that it is a significantly easier problem. That it might be such an easier problem is based upon the following observation.

For almost any blade D of grade $\lfloor m/2 \rfloor$, the blade $D \cdot A$, if not zero, is of grade $m - \lfloor m/2 \rfloor$ and must represent a vector sub-space of A . Furthermore, $(D \cdot A) \cdot A$ must represent the vector sub-space of A complement to $D \cdot A$. We may then let $B = D \cdot A$ and $C = (D \cdot A) \cdot A$. The degree to which B

and C are not zero is a point of numerical stability, but we will ignore this for the time being.

This problem, which we might refer to as indirect factorization, is easier than that of direct factorization, because the problem of finding the blade D is that of finding any set of $n = m - \lfloor m/2 \rfloor$ linearly independent vectors $\{d_k\}_{k=1}^n$ such that there does not exist $v \in \{d_k\}_{k=1}^n$ such that $v \cdot A = 0$. There is no requirement that any member of $\{d_k\}_{k=1}^n$ be in or out of the vector space represented by A . The only requirement is that for any integer $k \in [1, n]$, that d_k not be orthogonal to the sub-space represented by A , (i.e., d_k is not as far out of the vector space A as it can be.) (Now vector factor D can be in the dual space of A . Maybe this is also hard?)

The question now arises: how do we quickly find the blade D ? Interestingly, if we choose any blade D of grade n , the probability is that this blade will work. Therefore, two better questions are 1: how do we quickly find the best blade D , and 2: if we happen to incorrectly guess a blade D , how do we correct our guess?

Another problem with this approach is that we must calculate B and C each as a function of A and D , and this takes time. Furthermore, could this approach suffer from increasing numerical round-off error with each iteration of the algorithm?

More work needs to be done.