Elements of Design for Containers and Solutions in the LinBox library

Brice Boyer[*] Jean-Guillaume Dumas [†]

Abstract

We develop in this paper design techniques used in the C++ exact linear algebra library LinBox. They are intended to make the library safer and easier to use, while keeping it generic and efficient.

First, we review the new simplified structure of the containers, based on the *mother model* (cf. [1]). Namely, vectors and matrix are all templated by a field and a storage type. Matrix interface all agree with the same minimal blackbox interface. We explain the design choices and their impact on coding.

Then we present a variation of the *strategy* design pattern that is comprised of a controller–plugin system: the controller (solution) chooses among plug-ins (algorithms) and the plug-ins always call back the solution so a new choice can be made by the controller. We give examples using the solution `mul`. We will describe serveral of the new containers, especially our sparse and dense matrices storages as well as their `apply` method and compare to previous implementations.

Finally we present a benchmark architecture that serves two purposes. The first one consists in providing the user with an easy way to produces graphs using C++. The second one is to create a framework for tuning the library and doing regression testing.

References

[1] J.-G. Dumas, T. Gautier, C. Pernet, and B. D. Saunders. LinBox founding scope allocation, parallel building blocks, and separate compilation. In K. Fukuda, J. Van Der Hoeven, and M. Joswig, editors, *Proceedings of the Third International Congress Conference on Mathematical Software*, volume 6327 of *ICMS'10*, pages 77–83, Berlin, Heidelberg, Sept. 2010. Springer-Verlag.

[*] bbboyer@ncsu.edu; this material is based on work supported in part by the National Science Foundation under Grant CCF-1115772 (Kaltofen)

[†] Jean-Guillaume.Dumas@imag.fr