LAPP
Laboratoire d'Annecy de Physique des Particules

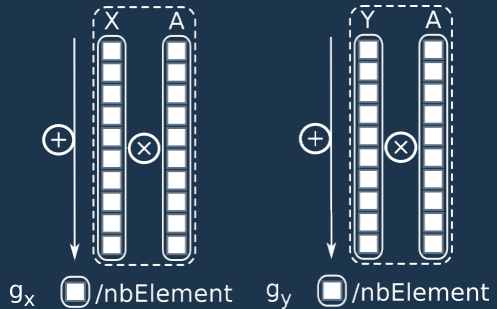
Optimisation : Barycentre

Pierre Aubert

LISTIC

The Barycentre 2d

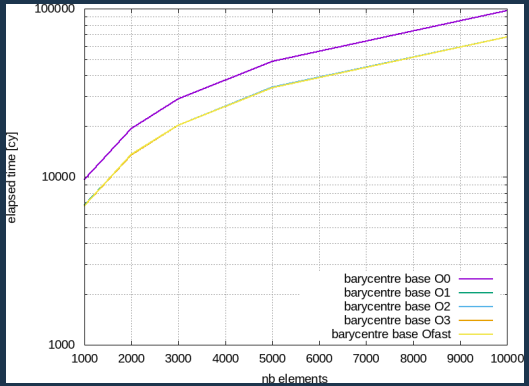
$$(g_x, g_y) = \frac{1}{N}\sum_{i=1}^{N}(x_i, y_i)\cdot a_i$$



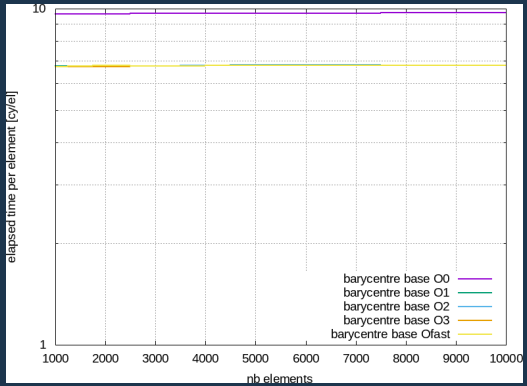

```
void barycentre(float & gx, float & gy, const float * tabX, const float* tabY, const float* tabA, long unsigned int nbElement){
	gx = 0.0f;
	gy = 0.0f;
	for(long unsigned int i(0lu); i < nbElement; ++i){
		gx += tabX[i]*tabA[i];
		gy += tabY[i]*tabA[i];
	}
	gx /= (float)nbElement;
	gy /= (float)nbElement;
}
```

The barycentre : performances

Total Elapsed Time (cy)

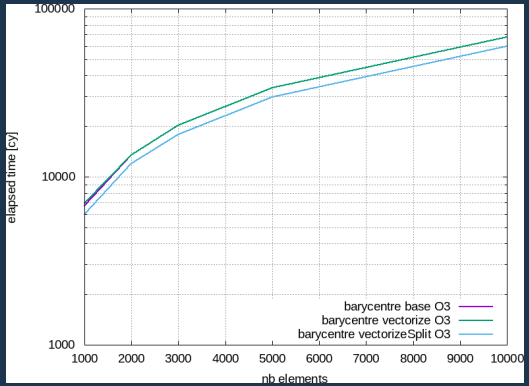


Elapsed Time per element (cy/el)

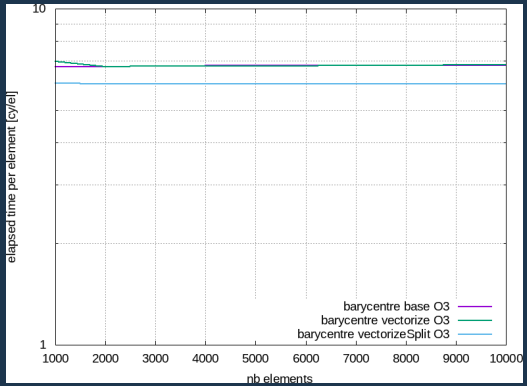


The barycentre : vectorize

Total Elapsed Time (cy)

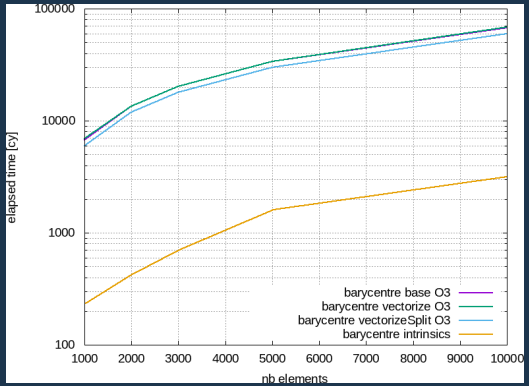


Elapsed Time per element (cy/el)

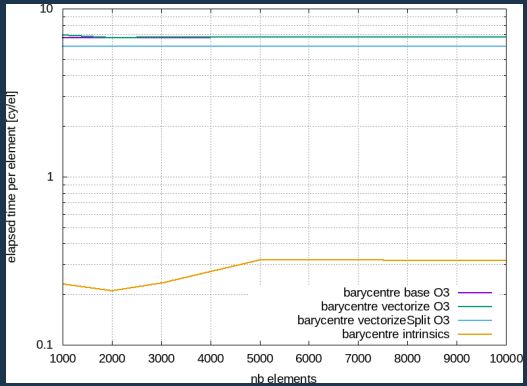


The barycentre : intrinsics

Total Elapsed Time (cy)



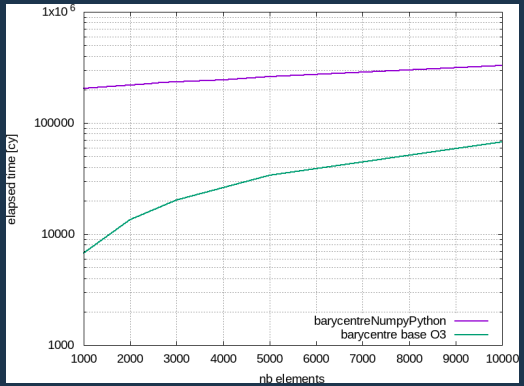
Elapsed Time per element (cy/el)



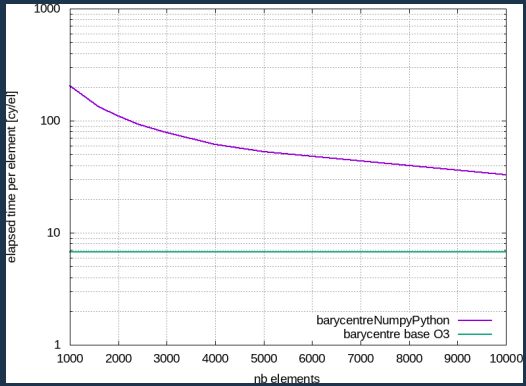
2000 elements, Intrinsics is **28** times faster than -**Ofast** vectorized

The barycentre : Python

Total Elapsed Time (cy)



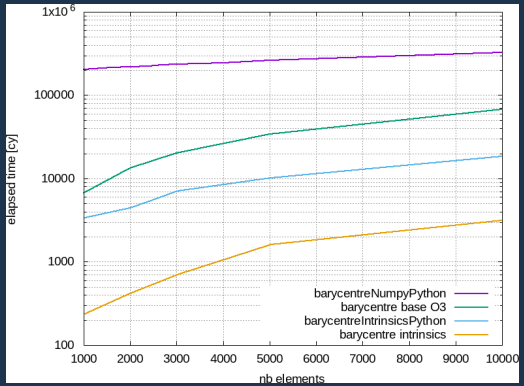
Elapsed Time per element (cy/el)



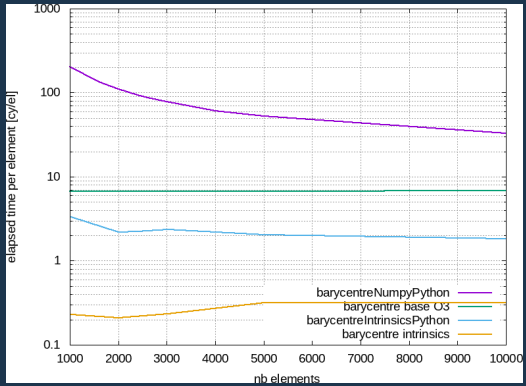
1000 elements, C++ O3 version is **5.7** times faster than **numpy** version

The barycentre : Python summary

Total Elapsed Time (cy)



Elapsed Time per element (cy/el)



1000 elements, our python version is **53** times faster than **numpy** version