HPC Assignment 1 Durbin optimization

https://github.com/II-castor/hpc-lab-assignment-1

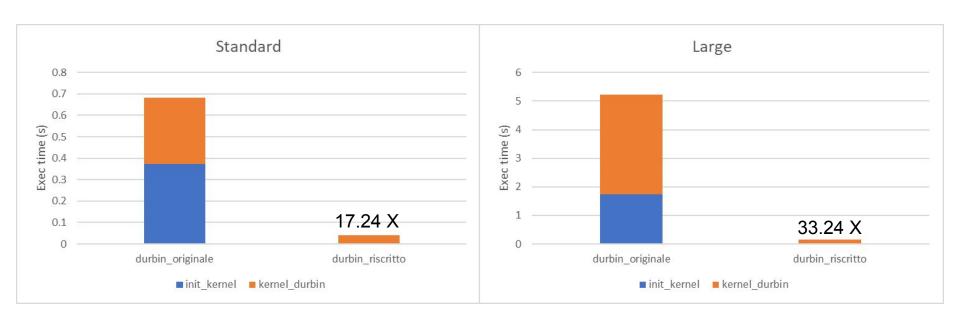
Balma Giovanni; Castorini Francesco; Rossi Lorenzo

Original code vs rewritten code

```
#pragma scop
 v[0][0] = r[0];
 beta[0] = 1;
 alpha[0] = r[0];
 for (k = 1; k < PB N; k++)
   beta[k] = beta[k - 1] - alpha[k - 1] * alpha[k - 1] * beta[k - 1];
   sum[0][k] = r[k]:
   for (i = 0; i \le k - 1; i++)
     sum[i + 1][k] = sum[i][k] + r[k - i - 1] * y[i][k - 1];
   alpha[k] = -sum[k][k] * beta[k];
   for (i = 0; i \le k - 1; i++)
     y[i][k] = y[i][k - 1] + alpha[k] * y[k - i - 1][k - 1];
   v[k][k] = alpha[k];
 for (i = 0; i < PB N; i++){}
   out[i] = y[i][_PB_N - 1];
```

```
int i, k;
 DATA_TYPE sum, beta, alpha;
 DATA_TYPE y[2][N];
#pragma scop
 y[0][0] = r[0];
 beta = 1;
 alpha = r[0];
 for (k = 1; k < PB N; k++)
   beta = beta - alpha * alpha * beta;
   sum = r[k];
   for (i = 0; i \le k - 1; i++)
     sum += r[k - i - 1] * y[(k - 1) % 2][i];
   alpha = -sum * beta;
   for (i = 0; i \le k - 1; i++)
     y[k \% 2][i] = y[(k - 1) \% 2][i] + alpha * y[(k - 1) \% 2][k - i - 1];
   y[k \% 2][k] = alpha;
 for (i = 0; i < PB N; i++)
   out[i] = y[(_PB_N - 1) \% 2][i];
```

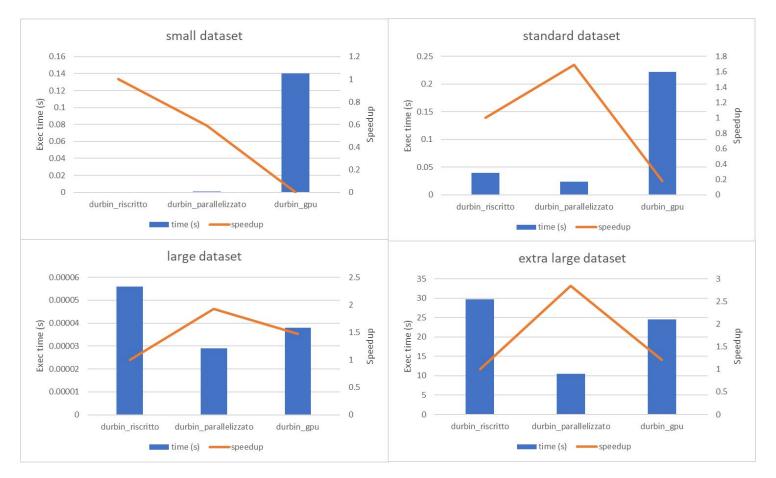
Performance boost



Parallel code

```
y[0][0] = r[0];
beta = 1;
alpha = r[0];
#pragma omp parallel default(none) firstprivate(i, k, alpha, beta, r, n, out) shared(sum, y) num threads(NTHREADS)
  for (k = 1; k < PB N; k++)
    beta = beta - alpha * alpha * beta;
    sum = r[k];
    #pragma omp for reduction(+:sum)
    for (i = 0; i <= k - 1; i++)
      sum += r[k - i - 1] * y[(k - 1) % 2][i];
    alpha = -sum * beta;
    #pragma omp for
    for (i = 0; i <= k - 1; i++)
     y[k \% 2][i] = y[(k - 1) \% 2][i] + alpha * y[(k - 1) % 2][k - i - 1];
    y[k \% 2][k] = alpha;
  #pragma omp for
  for (i = 0; i < PB N; i++)
    out[i] = y[(PB N - 1) \% 2][i];
```

Kernel Durbin



Optional assignment

