Laboratoire OpenMP

1 Extrait de code avec la boucle à paralléliser

```
{
  vrai_nb_threads = omp_get_num_threads();
  #pragma omp for schedule(static) reduction(+: somme)
  for ( int i = 0; i < nb_rectangles; i++ ) {
    double x = (i + 0.5) * largeur;   // Point milieu
    double fx = 4.0 / (1.0 + x*x);    // Hauteur au point milieu
    somme += fx;
  }
}

double pi = largeur * somme;
```

2 Fonctions coefficient et fois

```
static double coefficient( int i, Polynome p1, Polynome p2 )
{
  int expMinR1 = MAX( 0, i-p2.degre );
  int expMaxR1 = MIN( i, p1.degre );

  assert( expMinR1 <= expMaxR1 && "expMinR1 doit etre <= expMaxR1?

  double c = 0.0;
  # pragma omp parallel for reduction(+: c)
  for( int k = expMinR1; k <= expMaxR1; k++ ) {
    c += p1.coefficients[k] * p2.coefficients[i-k];
  }
  return( c );
}


Polynome fois( Polynome p1, Polynome p2 )
{
  Polynome p = allouer( p1.degre + p2.degre );

  # pragma omp parallel for schedule(dynamic)
  for( int i = 0; i <= p.degre; i++ ) {
    p.coefficients[i] = coefficient( i, p1, p2 );
  }
  return( p );
}
```