Exercise 3: Exploring Parallelism
July 9, 2014

1. What are the different ways of exploring parallelism?

2. What is the granularity of a parallelized computation? How to control it using loop optimizations?

3. What is the safety constraint of loop parallelization?

4. What is a private variable?

5. What is a reduction variable?

6. How can each of the following optimizations be used to control the granularity of parallelism of a program??
   (a) Loop parallelization via OpenMP parallel for
   (b) Loop vectorization via SIMD vector registers
   (c) Loop pipelining
   (d) Loop strip-mining
   (e) Loop interchange
   (f) Loop fusion
   (g) Loop skewing

Try parallelize each of the following code fragments via combinations of loop optimizations.

1.
```
void dgemm(double *a,double *b,
     double *c, double beta, int n)
{
  int i,j,k;
  for (j = 0; j < n; j ++)
  for (i = 0; i < n; i ++)
     c[j*n+i] = beta*c[j*n+i];
  for (k = 0; k < n; k ++)
  for (j = 0; j < n; j ++)
  for (i = 0; i < n; i ++)
    c[j*n+i] +=
           a[k*n+i] * b[j*n+k];
}
```

2.
```
DO J = 1, JMAXD
   DO I = 1, IMAXD
      F(I, J, 1) = F(I, J, 1) * B(1)
DO K = 2, N-1
   DO J = 1, JMAXD
      DO I = 1, IMAXD
         F(I,J,K)=(F(I,J,K) - A(K)*F(I,J,K-1))*B(K)
DO J = 1, JMAXD
   DO I = 1, IMAXD
      TOT(I, J) = 0.0
DO J = 1, JMAXD
   DO I = 1, IMAXD
      TOT(I, J) = TOT(I, J) + D(1) * F(I, J, 1)
DO K = 2, N-1
   DO J = 1, JMAXD
      DO I = 1, IMAXD
         TOT(I, J) = TOT(I, J) + D(K) * F(I, J, K)
```

3.
```
DO I = 1, N
 DO K = 1, N
  A(I) = A(I) + X(I,K)
 ENDDO
 DO J = 1, M
   DO K = 1, N
     B(J,K) = B(J,K) + A(I)
   ENDDO
   C(J,I) = B(J,N)/A(I)
 ENDDO
ENDDO
```