Triangular Solve (STRSM)

for (j = 0; j < n; j++)
    for (k = 0; k < n; k++)
        if (B[j*n+k] != 0.0f) {
            for (i = k+1; i < n; i++)
                B[j*n+i] -= A[k * n + i] * B[j * n + k];
        }

Equivalent to:
cublasStrsm('l' /* left operator */, 'l' /* lower triangular */, 
    'N' /* not transposed */, 'u' /* unit triangular */, 
    N, N, alpha, d_A, N, d_B, N);

See: http://www.netlib.org/blas/strsm.f
A Few Details

- C stores multi-dimensional arrays in row major order
- Fortran (and MATLAB) stores multi-dimensional arrays in column major order

- **Confusion alert**: BLAS libraries were designed for FORTRAN codes, so column major order is implicit in CUBLAS!
Dependences in STRSM

for (j = 0; j < n; j++)
    for (k = 0; k < n; k++)
        if (B[j*n+k] != 0.0f) {
            for (i = k+1; i < n; i++)
                B[j*n+i] -= A[k * n + i] * B[j * n + k];
        }

Which loop(s) “carry” dependences?
Which loop(s) is(are) safe to execute in parallel?
Assignment

• Details:
  – Integrated with simpleCUBLAS test in SDK
  – Reference sequential version provided

1. Rewrite in CUDA
2. Compare performance with CUBLAS library
Performance Issues?

- + Abundant data reuse
- - Difficult edge cases
- - Different amounts of work for different \( <j,k> \) values
- - Complex mapping or load imbalance