

Gaussian Elimination using SSE

Harshvardhan Agrawal (hka25)

Chirag Jain (csj43)

We use the following procedure to bring parallelism for SSE using SSE. We are dealing with 128 but registers here. We specify the steps used to write the **gauss_eliminate_using_sse** function.

- We firstly copy the data from matrix A to matrix U. In order to achieve this we use two pointers **cp1** & **cp2** pointing at **A** & **U** respectively. We also create a dummy register filled with 0.0. We add the cp1 and dummy register and store the value in location pointed by cp2.
- Next, we make the **principal diagonal element 1** at an **index k** by dividing that row by the principal element.
- After making the principal element 1 for a particular **row j**, we make the elements of the **column k** as 0 for all rows pointed by **index m (j+1 upto MATRIX_SIZE.)**

We do this by storing the element in the **column k** in a register. We multiply it by the row pointed to by **j** and subtract it from the row pointed by **index m**.

We perform the operation **MATRIX_SIZE/4** times.

We gave a **warm start** to the system by running it for sizes **64x64**, **256x256** and **512x512**.

The execution time for serial execution was approximately **5.82 s** for a **2048 x 2048** matrix.

For SSE execution we were getting approximately **4.14 s** execution time.

Thus, **Speedup** is **1.4x**.