- CS Login: **dspatel6**

- Wisc ID: **9085310937**

- Email: **dspatel6@wisc.edu /** [**dspatel6@cs.wisc.edu**](mailto:dspatel6@cs.wisc.edu)

**CPU Specifications**

**- Processor 13th Gen Intel(R) Core(TM) i5-1350P 1.90 GHz**

**- Installed RAM 16.0 GB (15.4 GB usable)**

**- Total Cores 12**

**- Performance-cores 4**

**- Efficient-cores 8**

**- Total Threads 16**

**- Max Turbo Frequency 4.70 GHz**

**- Performance-core Max Turbo Frequency 4.70 GHz**

**- Efficient-core Max Turbo Frequency 3.50 GHz**

**Memory bandwidth**

**DDR5-4800 (Dual-Channel)**

**Memory speed: 4800 MT/s**

**Bus width per DIMM: 64 bits (8 bytes)**

**Channels: 2 (dual-channel)**

**4800 × 8 × 2 = 76.8 GB/s**

**Compiler**

**```bash**

**$ g++ --version**

**g++ (Ubuntu 14.2.0-16ubuntu1) 14.2.0**

**Copyright (C) 2024 Free Software Foundation, Inc.**

**This is free software; see the source for copying conditions. There is NO**

**warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.**

**```**

**OS**

**```bash**

**$ cat /etc/os-release**

**PRETTY\_NAME="Ubuntu Plucky Puffin (development branch)"**

**NAME="Ubuntu"**

**VERSION\_ID="25.04"**

**```**

**Compilation command**

**```bash**

**$ make**

**```**

**Code Changes**

**float Norm(const float (&x)[XDIM][YDIM][ZDIM])**

**{**

**#ifdef DO\_NOT\_USE\_MKL**

**float result = 0.;**

**#pragma omp parallel for reduction(max:result)**

**for (int i = 1; i < XDIM-1; i++)**

**for (int j = 1; j < YDIM-1; j++)**

**for (int k = 1; k < ZDIM-1; k++)**

**result = std::max(result, std::abs(x[i][j][k]));**

**return result;**

**#else**

**return std::abs(((const float \*)x)[cblas\_isamax(XDIM\*YDIM\*ZDIM, (const float\*) x, 1)]);**

**#endif**

**}**

**float InnerProduct(const float (&x)[XDIM][YDIM][ZDIM], const float (&y)[XDIM][YDIM][ZDIM])**

**{**

**#ifdef DO\_NOT\_USE\_MKL**

**double result = 0.;**

**#pragma omp parallel for reduction(+:result)**

**for (int i = 1; i < XDIM-1; i++)**

**for (int j = 1; j < YDIM-1; j++)**

**for (int k = 1; k < ZDIM-1; k++)**

**result += (double) x[i][j][k] \* (double) y[i][j][k];**

**return result;**

**#else**

**return cblas\_sdot(XDIM\*YDIM\*ZDIM, (const float \*)x, 1, (const float \*)y, 1);**

**#endif**

**}**

**void Copy(const float (&x)[XDIM][YDIM][ZDIM], float (&y)[XDIM][YDIM][ZDIM])**

**{**

**#ifdef DO\_NOT\_USE\_MKL**

**#pragma omp parallel for**

**for (int i = 1; i < XDIM-1; i++)**

**for (int j = 1; j < YDIM-1; j++)**

**for (int k = 1; k < ZDIM-1; k++)**

**y[i][j][k] = x[i][j][k];**

**#else**

**cblas\_scopy(**

**XDIM \* YDIM \* ZDIM, // Length of vectors**

**&x[0][0][0], // Input vector x**

**1, // Use step 1 for x**

**&y[0][0][0], // Output vector y**

**1 // Use step 1 for y**

**);**

**#endif**

**}**

**LaplaceSolver\_1\_4**

**### LaplaceSolver\_1\_4**

**```bash**

**$ make**

**$ ./laplace\_solver**

**[Initialization : 33307.1ms]**

**Residual norm (nu) after 0 iterations = 1**

**Residual norm (nu) after 1 iterations = 0.720741**

**Residual norm (nu) after 2 iterations = 0.503969**

**Residual norm (nu) after 3 iterations = 0.316996**

**Residual norm (nu) after 4 iterations = 0.167438**

**Residual norm (nu) after 5 iterations = 0.114786**

**Residual norm (nu) after 6 iterations = 0.0673758**

**Residual norm (nu) after 7 iterations = 0.0423407**

**Residual norm (nu) after 8 iterations = 0.0429277**

**Residual norm (nu) after 9 iterations = 0.0358568**

**Residual norm (nu) after 10 iterations = 0.0317483**

**Residual norm (nu) after 11 iterations = 0.0258011**

**Residual norm (nu) after 12 iterations = 0.0200023**

**Residual norm (nu) after 13 iterations = 0.0147499**

**Residual norm (nu) after 14 iterations = 0.0103051**

**Residual norm (nu) after 15 iterations = 0.00810947**

**Residual norm (nu) after 16 iterations = 0.00639989**

**Residual norm (nu) after 17 iterations = 0.00574772**

**Residual norm (nu) after 18 iterations = 0.00474403**

**Residual norm (nu) after 19 iterations = 0.00384194**

**Residual norm (nu) after 20 iterations = 0.00263516**

**Residual norm (nu) after 21 iterations = 0.0021436**

**Residual norm (nu) after 22 iterations = 0.00160884**

**Residual norm (nu) after 23 iterations = 0.00102262**

**Conjugate Gradients terminated after 23 iterations; residual norm (nu) = 0.000850818**

**[Total Laplacian Time : 3181.35ms]**

**[Total Saxpy Time : 523.062ms]**

**[Total Copy Time : 129.019ms]**

**[Total DotProduct Time : 328.117ms]**

**[Total Norm Time : 174.928ms]**

**```**

**LaplaceSolver\_1\_4 With MKL**

**### LaplaceSolver\_1\_4 MKL**

**```bash**

**$ make**

**$ ./laplace\_solver**

**[Initialization : 82132.1ms]**

**Residual norm (nu) after 0 iterations = 1**

**Residual norm (nu) after 1 iterations = 0.720746**

**Residual norm (nu) after 2 iterations = 0.503966**

**Residual norm (nu) after 3 iterations = 0.317025**

**Residual norm (nu) after 4 iterations = 0.167438**

**Residual norm (nu) after 5 iterations = 0.11479**

**Residual norm (nu) after 6 iterations = 0.0673855**

**Residual norm (nu) after 7 iterations = 0.042335**

**Residual norm (nu) after 8 iterations = 0.0428855**

**Residual norm (nu) after 9 iterations = 0.0358354**

**Residual norm (nu) after 10 iterations = 0.0317406**

**Residual norm (nu) after 11 iterations = 0.0257948**

**Residual norm (nu) after 12 iterations = 0.0199994**

**Residual norm (nu) after 13 iterations = 0.0147458**

**Residual norm (nu) after 14 iterations = 0.0103037**

**Residual norm (nu) after 15 iterations = 0.00810744**

**Residual norm (nu) after 16 iterations = 0.0063971**

**Residual norm (nu) after 17 iterations = 0.00574608**

**Residual norm (nu) after 18 iterations = 0.00474247**

**Residual norm (nu) after 19 iterations = 0.00384114**

**Residual norm (nu) after 20 iterations = 0.00263496**

**Residual norm (nu) after 21 iterations = 0.00214284**

**Residual norm (nu) after 22 iterations = 0.00160846**

**Residual norm (nu) after 23 iterations = 0.00102251**

**Conjugate Gradients terminated after 23 iterations; residual norm (nu) = 0.000850518**

**[Total Laplacian Time : 5612.97ms]**

**[Total Saxpy Time : 460.622ms]**

**[Total Copy Time : 96.4345ms]**

**[Total DotProduct Time : 220.16ms]**

**[Total Norm Time : 103.169ms]**

**```**