The `Procrustes()` kernel was analyzed as shown in the following:

- Memory Traffic β: Inside the loop, for each iteration $n = 1, 2, \ldots, N$, there are 4 reads from memory: $x[i].x$, $x[i].y$, $y[i].x$, and $y[i].y$. Outside the loop, local variables $X1$, $X2$, $Y1$, $Y2$, Z, $C1$, and $C2$ are read and updated. Hence the total number of reads and writes are $(4N + 7)$ reads $+ 7$ writes $= 4N + 14$. Because we are using double precision numbers, the total memory traffic is $(4N + 14) \times 8\text{Byte} = 32N + 112$ Byte.
- FLOPs π: Inside the loop, for each iteration $n = 1, 2, \ldots, N$, there are 4 addition operations for $X1$, $X2$, $Y1$, $Y2$, 2 multiplication and addition operations for Z, 4 multiplication and 2 addition operations for $C1$, and 4 multiplication and 1 subtraction operation for $C2$. Hence the total FLOPs is $\pi = 17N$. Outside the loop, there is a one-time matrix inversion and matrix multiplication operation, which involves around 160 FLOPs. If N is large, this part becomes negligible.
- Operational Intensity I: Given the above computations, we find the operational intensity to be 0.53 Flops/Byte.
- Performance: We then measured the runtime of both `Procrustes()` and `ProcrustesV()` and computed the performance for each kernel. For a data size of 10^8, $T_s = 41.1$ ms and $T_p = 25.0$ ms. Hence the performances of the serial and the optimized function are 4.1 GFlops/s and 6.0 GFlops/s, respectively.