

HW3

Juan Li

jl10889@nyu.edu

1. Pitch your final project

We have already emailed you about our project proposal.

2. Approximating Special Functions Using Taylor Series & Vectorization

run original `sin(x)`

```
[jl10889@access1 hw3]$ ./fast-sin
Reference time: 17.4749
Taylor time:      2.4350      Error: 6.928125e-12
Intrin time:      0.6224      Error: 2.454130e-03
Vector time:      0.8839      Error: 2.454130e-03
```

After implementing the function `sin4_vec()` by adding more items, I got the accuracy to 12-digits.

```
[jl10889@access1 hw3]$ ./fast-sin
Reference time: 17.1321
Taylor time:      2.4583      Error: 6.928125e-12
Intrin time:      0.6104      Error: 2.454130e-03
Vector time:      0.7471      Error: 6.928125e-12
```

Extra credit

The properties of sin:

$$\begin{aligned}\sin(x) &= \sin(x + 2\pi i) \\ \sin(x) &= -\sin(-x)\end{aligned}$$

We can compute sine value in range $[-\pi/2, \pi/2]$, and get outside value by this.

$$\begin{aligned}\text{if } x \in [\pi/2 + 2\pi i, 3\pi/2 + 2\pi i], \sin(x) &= -\sin(x - 2\pi i - \pi) \\ \text{if } x \in [-\pi/2 + 2\pi i, \pi/2 + 2\pi i], \sin(x) &= -\sin(x - 2\pi i)\end{aligned}$$

Since the sin's taylor series is $\sin(x) = x - x^3/3! + x^5/5! - x^7/7!...$,

we can reverse the symbol in each part to get the minus value, $-\sin(x) = -x + x^3/3! - x^5/5! + x^7/7!...$

I implemented the $[-\pi/2, \pi/2]$ in fast-sin2.cpp (only the taylor and vec version.

and I implemented the $[\pi/2, 3\pi/2]$ in fast-sin3.cpp (only the taylor and vec version.

result:

```
[jl10889@access2 hw3]$ ./fast-sin2
Reference time: 16.4367
Taylor time:      2.4433      Error: 5.625877e-08
Intrin time:      0.5437      Error: 7.516768e-02
Vector time:      0.6842      Error: 5.625877e-08
```

```
[jl10889@access2 hw3]$ ./fast-sin3
Reference time: 33.2468
Taylor time:      2.5178      Error: 5.625877e-08
Vector time:      0.8960      Error: 5.625877e-08
```

3. Parallel Scan in OpenMP

The result is

threads	runtime
serial	0.75s
1	0.30s
2	0.247s
3	0.209s
4	0.177s
8	0.199s
16	0.199s

```
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=1
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.747851s
Number of threads: 1
parallel-scan   = 0.300385s
error = 0
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=2
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.752604s
Number of threads: 2
parallel-scan   = 0.247087s
error = 0
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=3
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.870833s
Number of threads: 3
parallel-scan   = 0.209445s
error = 0
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=4
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.732510s
Number of threads: 4
parallel-scan   = 0.177296s
error = 0
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=8
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.799525s
Number of threads: 8
parallel-scan   = 0.199154s
error = 0
[jl10889@access2 hw3]$ export OMP_NUM_THREADS=16
[jl10889@access2 hw3]$ ./omp-scan
sequential-scan = 0.909125s
Number of threads: 16
parallel-scan   = 0.199045s
error = 0
```

I run on NYU's cims linux machine, CPU: 4; Threads per core: 1.