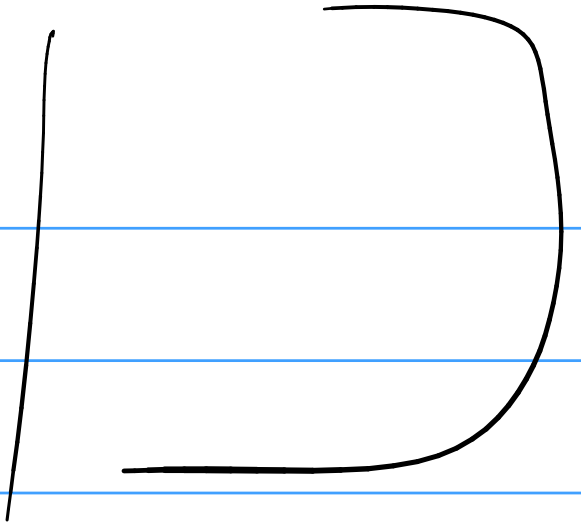
Lecture 9



- no class Friday
- HW1 grading
- HW2: Python vec. intrinsics
- HW2 deadline?

- Laugh at C
- Laugh at GPUs

Arrays vs Abstraction

Arrays-of-Structures or Structures-of-Arrays? What's the difference? Give an example.

Language aspects of the distinction? Salient example?

- Complex numbers

C and Multi-Dimensional Arrays: A Saving Grace

```
// YES:
void f(int m, int n, double (*)[m][n]);

// NO:
struct ary {
  int m;
  int n;
  double (*array)[m][n];
};

// YES:
struct ary {
  int m;
  int n;
  double a[];
};
```

SIMD

Name language mechanisms for SIMD:

Demo: machabstr/Ways to SIMD

Outer-Loop/inner-Loop Vectorization

Contrast *outer-loop* vs *inner-loop vectorization*.

Side q: Would you consider GPUs outer- or inner-loop-vectorizing?

Alignment: How?

The old way:

```
int __attribute__ ((aligned (8))) a_int;
```

Difference between these two?

```
int __attribute__ ((aligned (8))) * ptr_t_1;
int *__attribute__ ((aligned (8))) ptr_t_2;
```

The 'new' way (C/C++11):

```
struct alignas(64) somestruct_t { /* ... */ };
struct alignas(alignof(other_t))
  somestruct_t { /* ... */ };
struct
  alignas(
    std::hardware_destructive_interference_size)
    somestruct_t { /* ... */ };
```

What is *constructive interference*?

Alignment: Why?

What is the concrete impact of the constructs on the previous slide?