**1. Case Study**
Turbulent Flow in Computational Fluid Dynamics
With Eric Arobone and Sutanu Sarkar
- A shear layer causes turbulent flow
- Statistics on the whole grid and on planes
- Build a Domain Specific Library
  - High-level primitives

**2. High Level Primitives**
Abstractions ease programming
- Domains
- Arrays (Domain \(\rightarrow\) Value)
- Iterators
- Functors
- Aggregators

**3. Queries can be composed, making it easy to build domain-specific abstractions.**

**4. Abstractions are expensive**
```
for(Iterator i = domain.begin();
i != domain.end(); ++i)
f(i);
```
Lots of code to implement iterator increment
```
++z;
if (z > max[2])
{
  ++y;
  z = max[2];
  if (y > max[1])
  {
    ++x;
    y = min[1];
    if (x > max[0])
    {
      DONE();
    }
  }
}
```

**5. Solution**
- Write a library which can be optimized.
- We pre-process library constructs and erase them before compilation.
- Utilize knowledge of the library semantics to improve performance.

**6. Performance**
- Three optimizations yield a 2.75x speedup
  1. Naïve: Saaz Directly
  2. Change loops using iterators to nested loops using integers
  3. Combine index calculation of multiple arrays
  4. Inline an expensive function

Support: NSF (OCE-0835839)
We can extend the planar averaging query
\(\Sigma\)
to compute the buoyancy flux
\(\Sigma \rho'w'\)
or compute it only in certain places
\(\lambda_2 < \epsilon\)