A Query Language for a Computational Database
Alden King; Scott Baden, Advisor

What is a computational database?

- Computational science derives knowledge from large datasets.
- Relational databases are not appropriate for scientific data. Particularly, the relational database model doesn’t easily:
  - Express arbitrary aggregation functions
  - Preserve locality (stencil operators require this)
- Computational database for data analysis
  - Non-relational query language which combines queries with computation
  - Generate source code for optimized programs to do the analysis. Target the appropriate architecture, such as host CPUs, or compute cards like GPUs.

Identifying Vortices in Horizontal Sheers
With Sutanu Sarkar,
Hieu Pham, and Eric Arobone

Identification of vortices is done via the “delta-criterion”: \( \delta > \epsilon \).
We then want to evaluate a function over these points. This requires conditional execution:

```plaintext
{  delta(0,0,0) = nx,ny,nz;  /* compute f */  nx,ny,nz where delta > \epsilon
}
```

We can use this to optimize for \( \delta \) being sparse and for caching based on epsilon.
This is equivalent to the C version:

```plaintext
for (int i = 0; i < nx; ++i)
  for (int j = 0; j < ny; ++j)
    for (int k = 0; k < nz; ++k)
      if ( f(i, j, k) > delta ) /* compute f */
```
or the CUDA version:

```plaintext
__global__ void ComputeF_kernel( Mat* f, Mat* delta, float epsilon, ... ) {
  if ( f(threadIdx.x, threadIdx.y, threadIdx.z) > delta ) /* compute f */
}
```

why a new language?

We use generate source code for different targets, and they have different models for which we want to optimize. We can choose the appropriate representations of primitives automatically according to the target and the data itself. Generating source code lets us take advantage of existing compiler optimizations, as well as implement hand-tuning if necessary.

<table>
<thead>
<tr>
<th>Target</th>
<th>Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Flattened array (dense); Sparse-matrix</td>
</tr>
<tr>
<td>CUDA</td>
<td>Flattened array (small, dense); Blocked array (dense); Sparse-matrix</td>
</tr>
<tr>
<td></td>
<td>Kernel function and method call</td>
</tr>
</tbody>
</table>

Support: NSF (OCE-0835839)