# ProtoX: A First Look

Het Mankad*, Sanil Rao*, Phillip Colella†, Brian Van Straalen†, Franz Franchetti*
* Carnegie Mellon University, † Lawrence Berkeley National Laboratory

## Problem

**Stencil Operations**: A key component in numerical solutions to partial differential equations (PDEs).

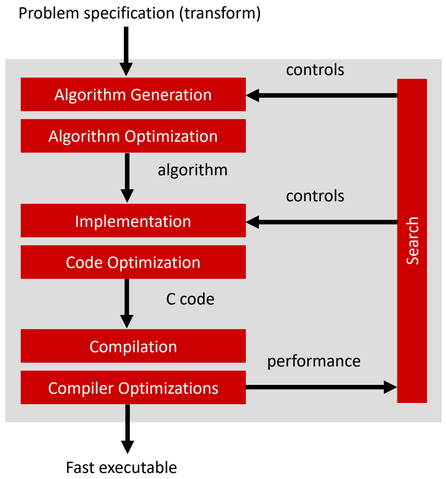
**Proto:** It is a domain specific library written in C++ that provides a high level of abstraction for solving PDEs using various numerical methods.

- **Shortcoming:** Abstraction fusion is something no compiler can easily perform.
- **Our Goal:** To interpret Proto as a Domain Specific Language with the help of SPIRAL [1],[2] and obtain better performance.

## SPIRAL

**Spiral:**
Complete automation of the implementation and optimization task
**Basic idea:**
Declarative representation of algorithms using the Signal Processing Language (SPL) and the Operator Language (OL)

Rewriting systems to generate and optimize algorithms

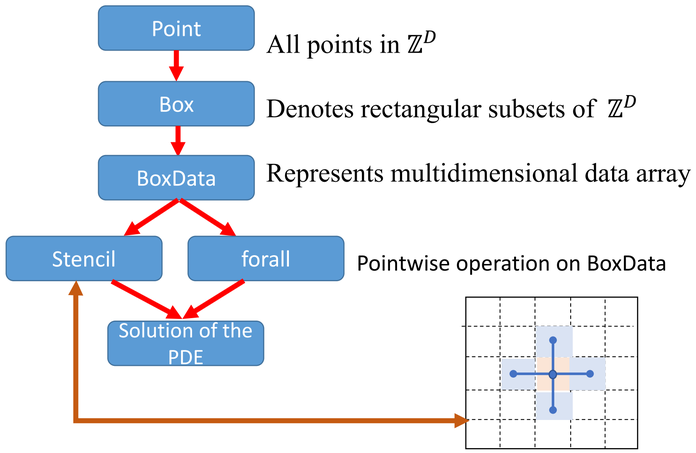



## Acknowledgment

This project is funded by the DOE Office of Advanced Scientific Computing Research the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

## Proto

**Design layout and class structure in Proto**




## ProtoX

2D Poisson Equation is given as

$$\Delta\phi(x,y) = \rho(x,y), \quad x,y \in \Omega := [0,1] \times [0,1],$$

where $\rho$ is the given function, $\phi$ is what we are solving for and $\Delta$ is the Laplace operator.

- The Jacobi iteration method is used to solved the Poisson equation.
- The SPL breakdown rules in SPIRAL to compute the Poisson problem with $n \times n$ domain and $m \times m$ interior elements are

$$\text{Poisson2D}^{\ell,w,a}_{n,m,t} \rightarrow \begin{bmatrix} \text{Jacobi}_{n,m,w,l} \\ \|.\|^{n,m,a}_{\infty} \end{bmatrix} \circ \left( \begin{bmatrix} \text{I}_{n^2} \\ \text{Laplace2D} \end{bmatrix} \oplus \text{I}_{n^2} \right),$$

$$\text{Laplace2D}_{n,m,t} \rightarrow \text{Scatter}_{n^2 \times m^2} \circ [\text{Filt}(t)]^{m^2}_{i=0},$$

$$\text{Jacobi}_{n,m,w,l} \rightarrow (1, w, -\lambda) \otimes \text{I}_{n^2},$$

$$\|.\|^{n,m,a}_{\infty} \rightarrow (0, 1/(a^2), -1) \otimes \text{I}_{n^2}.$$

## ProtoX: Sample Code

**Sample of SPIRAL generated code for the 2D Poisson problem**

```
void poisson2D(double   *Y, double   *X, double weight1, double lambda1, double   *rhs,
double a_h1, double  *retval1) {
    static double T1[4357];
    static double T2[13068];
    static double T3[8452];
    . . .
    // Computing the Laplacian
    for(int i13 = 0; i13 <= 4095; i13++) {
        int a691;
        a691 = ((66*(i13 / 64)) + (i13 % 64));
          T2[(a691 + 4423)] = ((T3[(a691 + 1)] - (4.0*T3[(a691 + 67)]))
                    + T3[(a691 + 66)] + T3[(a691 + 68)]
                    + T3[(a691 + 133)]);
    }
    . . .
    // Jacobi Iteration
    for(int i6 = 0; i6 <= 4355; i6++) {
        T1[i6] = ((T8[i6] + (weight1*T8[(i6 + 4356)]))
              - (lambda1*T8[(i6 + 8712)]));
    }
    . . .
    // Computing || ||_{inf}
    for(int i10 = 0; i10 <= 4355; i10++) {
        T14[i10] = fabs(((1 / (a_h1*a_h1))*T15[(i10 + 4356)])
                - T15[(i10 + 8712)]);
    }
    . . .
    for(int i2 = 0; i2 <= 4355; i2++) {
        t3 = ((((T13[i2] >= t3))) ? (T13[i2]) : (t3));
    }
}
```

**Speedup : ProtoX gives 6x speedup on CPU over the baseline Proto code.**

## Conclusion & Future Work

- A proof of concept of having SPIRAL generated code as the backend for Proto is shown.
- We can interpret Proto as a DSL by writing a Proto program as a SPIRAL specification.
- The future goal is to add more targets and make ProtoX interoperable with FFTX [3] to do cross library optimization.

## References

[1] M. Puschel et al., "SPIRAL: Code Generation for DSP Transforms," in Proceedings of the IEEE, vol. 93, no. 2, pp. 232-275, Feb. 2005.
[2] Franchetti et al., "Formal Loop merging for signal transforms", in Proceedings of ACM SIGPLAN, 40, 6 , pp. 315-326, June 2005.
[3] F. Franchetti *et al.*, "FFTX and SpectralPack: A First Look," 2018 IEEE 25th International Conference on High Performance Computing Workshops, 2018, pp. 18-27