Chainable Stencil Operators

Current interface:

1. Define general stencil along 1 dimension

2. Instantiate wrapper class for a specific stencil and dimension
   Defines `T operator()(Index idx){}` on a field or another stencil instance

3. Potentially concatenate N of these stenils

Possible use case:

Generate Hessian with different numerical stencils along each dimension

Appendix: Code Snippets

Stencil Definition:

```
template<Dim D, typename T, class Callable>
inline T forward_stencil(const Index &idx, const T &hInv, const Callable &F){
    return 0.5 * hInv * (-3.0*F(idx) + 4.0*F(idx.get_shifted<D>(1)) - F(idx.get_shifted<D>(2)));
}
```

Chainable operator along dimension `D` with stencil `Diff`:

```
template<Dim D, typename T, DiffType Diff, class C>
class DiffOpChain{
    // [...]
    const inline T operator()(Index idx) const;
};
```