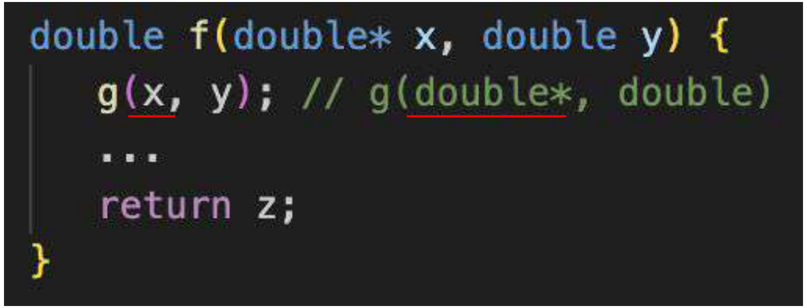
Wrap-Up:
Improve automatic differentiation of object-oriented paradigms using Clad

Petro Zarytskyi
Google Summer of Code
Julius-Maximilians-Universität, Germany
Mentors: Vassil Vassilev, David Lange

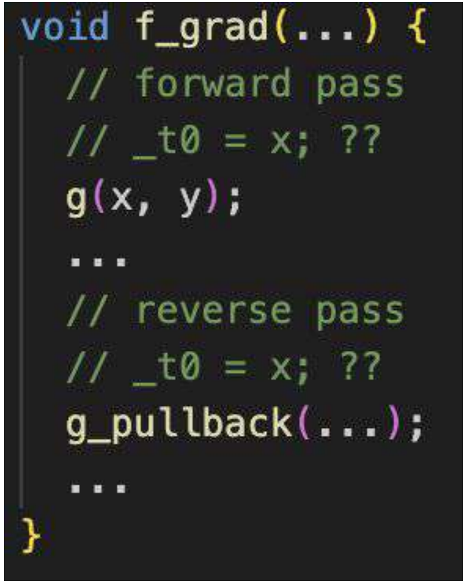
Recap: what didn’t work

Original code



```
double f(double* x, double y) {
    g(x, y); // g(double*, double)
    ...
    return z;
}
```

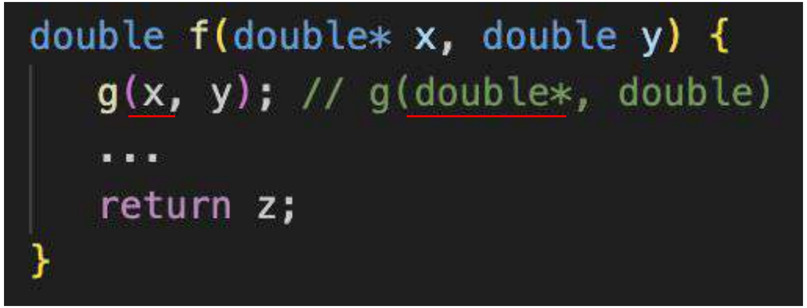
Code differentiated by Clad



```
void f_grad(...) {
    // forward pass
    // _t0 = x; ??
    g(x, y);
    ...
    // reverse pass
    // _t0 = x; ??
    g_pullback(...);
    ...
}
```

Recap: what didn’t work

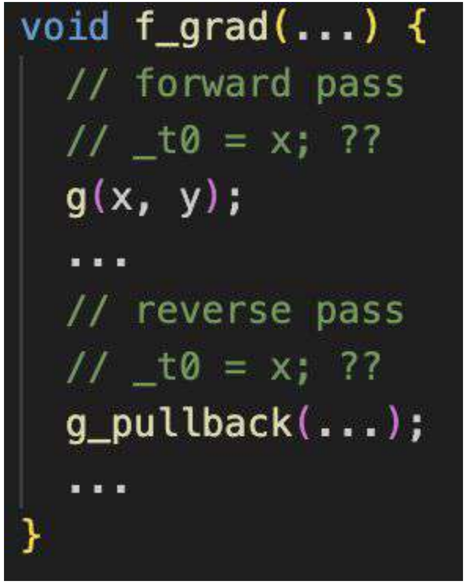
Original code



```
double f(double* x, double y) {
    g(x, y); // g(double*, double)
    ...
    return z;
}
```

The argument is not fully copyable!

Code differentiated by Clad



```
void f_grad(...) {
    // forward pass
    // _t0 = x; ??
    g(x, y);
    ...
    // reverse pass
    // _t0 = x; ??
    g_pullback(...);
    ...
}
```

What doesn’t work

The same goes with all non-copyable types:

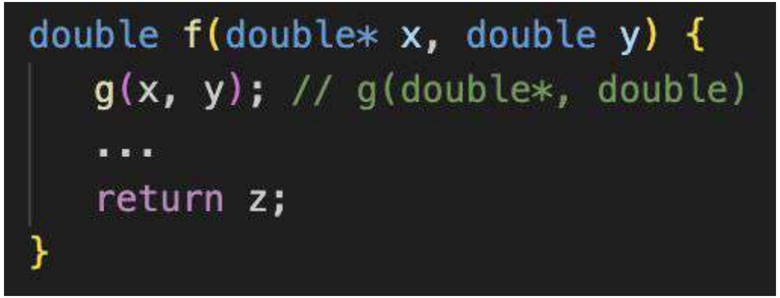
- std::initializer_list
- std::unique_ptr, std::shared_ptr, etc.
- Other STL and user-defined types.

Note: Methods of such classes follow the same logic as plain functions.

Implemented solution

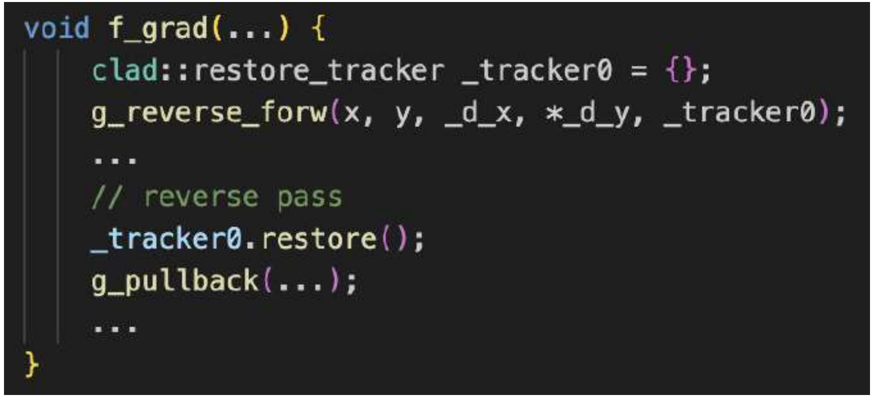
We introduced clad::restore_tracker (previously, clad::smart_tape) that can automatically store and restore multiple objects at the same time

Original code



```
double f(double* x, double y) {
    g(x, y); // g(double*, double)
    ...
    return z;
}
```

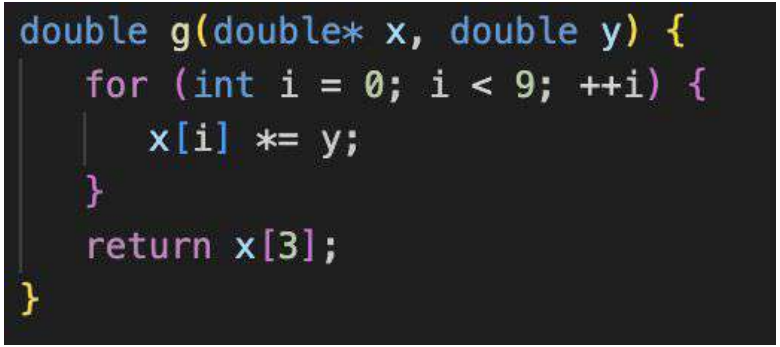
Code differentiated by Clad



```
void f_grad(...) {
    clad::restore_tracker _tracker0 = {};
    g_reverse_forw(x, y, _d_x, *_d_y, _tracker0);
    ...
    // reverse pass
    _tracker0.restore();
    g_pullback(...);
    ...
}
```

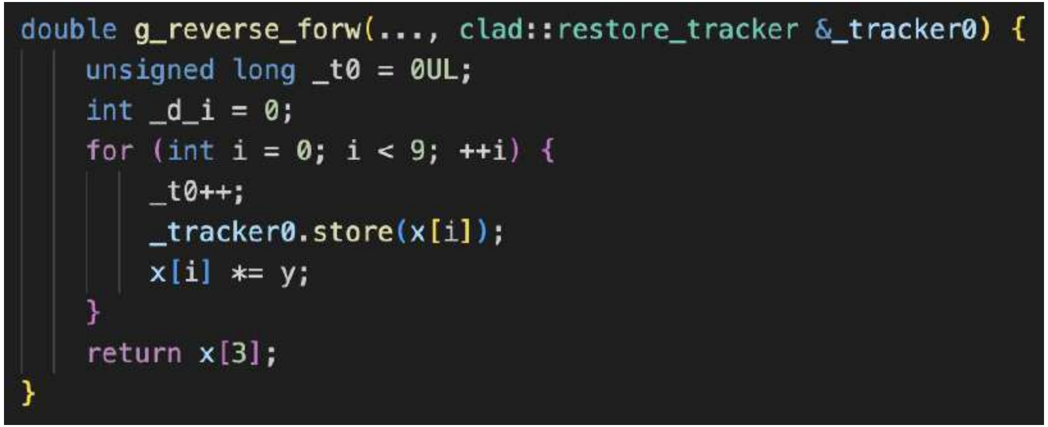
Implemented solution

Original code



```
double g(double* x, double y) {
    for (int i = 0; i < 9; ++i) {
        x[i] *= y;
    }
    return x[3];
}
```

Code differentiated by Clad



```
double g_reverse_forw(..., clad::restore_tracker &_tracker0) {
    unsigned long _t0 = 0UL;
    int _d_i = 0;
    for (int i = 0; i < 9; ++i) {
        _t0++;
        _tracker0.store(x[i]);
        x[i] *= y;
    }
    return x[3];
}
```

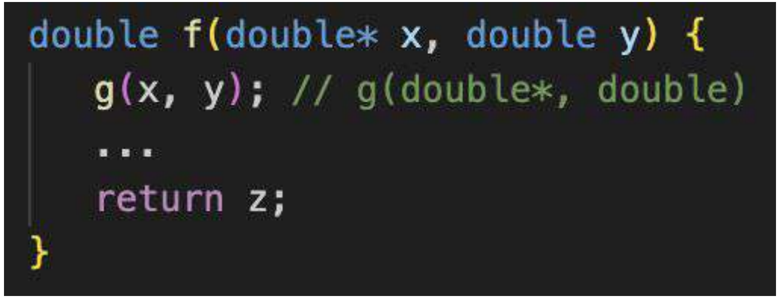
Implemented solution

- Already on Clad’s master
- Well-tested with nested calls and local variables
- Fixed an example from ROOT
- The system is attached to TBR analysis to avoid unnecessary uses

The second originally planned approach

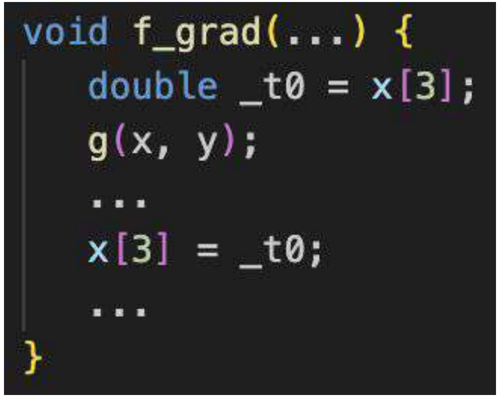
It was meant to cover “nice” cases when the argument can be stored easily outside of the original function.

Original code



```
double f(double* x, double y) {
    g(x, y); // g(double*, double)
    ...
    return z;
}
```

Code differentiated by Clad



```
void f_grad(...) {
    double _t0 = x[3];
    g(x, y);
    ...
    x[3] = _t0;
    ...
}
```

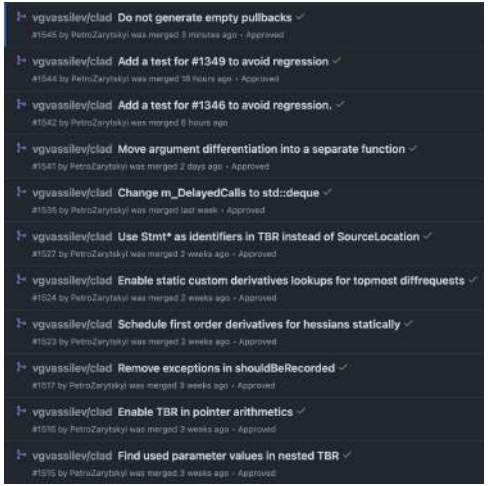
The second originally planned approach

- We decided that this feature is not currently needed as there are more relevant OOP-related issues
- Although we are halfway there as the approach relies on TBR analysis and most of the necessary improvements there were made

So what was done instead?

So what was done instead?

47 merged Pull Requests



vgvassilev/clad Do not generate empty pullbacks ✓
#1545 by PetroZarytskyi was merged 3 minutes ago · Approved

vgvassilev/clad Add a test for #1349 to avoid regression ✓
#1544 by PetroZarytskyi was merged 18 hours ago · Approved

vgvassilev/clad Add a test for #1346 to avoid regression. ✓
#1542 by PetroZarytskyi was merged 6 hours ago

vgvassilev/clad Move argument differentiation into a separate function ✓
#1541 by PetroZarytskyi was merged 2 days ago · Approved

vgvassilev/clad Change m_DelayedCalls to std::deque ✓
#1535 by PetroZarytskyi was merged last week · Approved

vgvassilev/clad Use Stmt* as identifiers in TBR instead of SourceLocation ✓
#1527 by PetroZarytskyi was merged 2 weeks ago · Approved

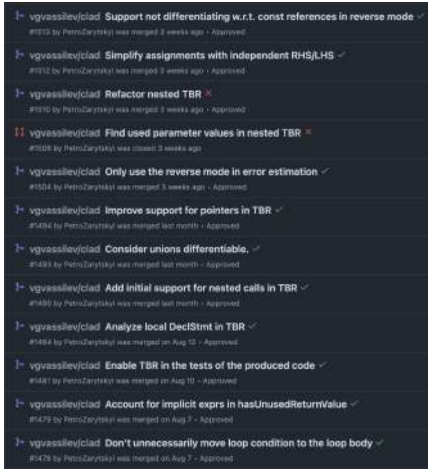
vgvassilev/clad Enable static custom derivatives lookups for topmost diffrequests ✓
#1524 by PetroZarytskyi was merged 2 weeks ago · Approved

vgvassilev/clad Schedule first order derivatives for hessians statically ✓
#1523 by PetroZarytskyi was merged 2 weeks ago · Approved

vgvassilev/clad Remove exceptions in shouldBeRecorded ✓
#1517 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Enable TBR in pointer arithmetics ✓
#1516 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Find used parameter values in nested TBR ✓
#1515 by PetroZarytskyi was merged 3 weeks ago · Approved



vgvassilev/clad Support not differentiating w.r.t. const references in reverse mode ✓
#1513 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Simplify assignments with independent RHS/LHS ✓
#1512 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Refactor nested TBR ✗
#1510 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Find used parameter values in nested TBR ✗
#1508 by PetroZarytskyi was closed 3 weeks ago

vgvassilev/clad Only use the reverse mode in error estimation ✓
#1504 by PetroZarytskyi was merged 3 weeks ago · Approved

vgvassilev/clad Improve support for pointers in TBR ✓
#1494 by PetroZarytskyi was merged last month · Approved

vgvassilev/clad Consider unions differentiable. ✓
#1493 by PetroZarytskyi was merged last month · Approved

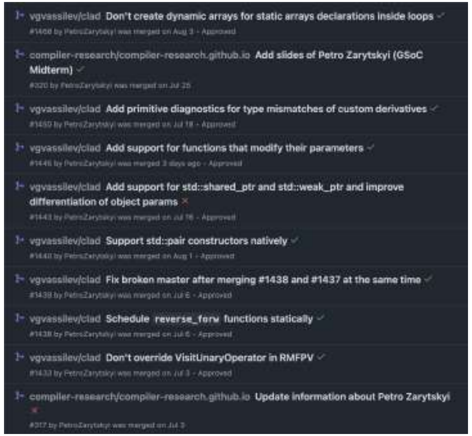
vgvassilev/clad Add initial support for nested calls in TBR ✓
#1490 by PetroZarytskyi was merged last month · Approved

vgvassilev/clad Analyze local DeclStmt in TBR ✓
#1484 by PetroZarytskyi was merged on Aug 12 · Approved

vgvassilev/clad Enable TBR in the tests of the produced code ✓
#1481 by PetroZarytskyi was merged on Aug 10 · Approved

vgvassilev/clad Account for implicit exprs in hasUnusedReturnValue ✓
#1479 by PetroZarytskyi was merged on Aug 7 · Approved

vgvassilev/clad Don't unnecessarily move loop condition to the loop body ✓
#1478 by PetroZarytskyi was merged on Aug 7 · Approved



vgvassilev/clad Don't create dynamic arrays for static arrays declarations inside loops ✓
#1468 by PetroZarytskyi was merged on Aug 3 · Approved

compiler-research/compiler-research.github.io Add slides of Petro Zarytskyi (GSoC Midterm) ✓
#320 by PetroZarytskyi was merged on Jul 25

vgvassilev/clad Add primitive diagnostics for type mismatches of custom derivatives ✓
#1450 by PetroZarytskyi was merged on Jul 18 · Approved

vgvassilev/clad Add support for functions that modify their parameters ✓
#1446 by PetroZarytskyi was merged 3 days ago · Approved

vgvassilev/clad Add support for std::shared_ptr and std::weak_ptr and improve differentiation of object params ✗
#1443 by PetroZarytskyi was merged on Jul 16 · Approved

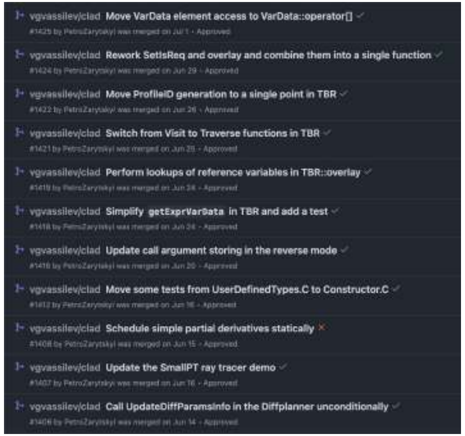
vgvassilev/clad Support std::pair constructors natively ✓
#1440 by PetroZarytskyi was merged on Aug 1 · Approved

vgvassilev/clad Fix broken master after merging #1438 and #1437 at the same time ✓
#1439 by PetroZarytskyi was merged on Jul 6 · Approved

vgvassilev/clad Schedule `reverse_forw` functions statically ✓
#1438 by PetroZarytskyi was merged on Jul 6 · Approved

vgvassilev/clad Don't override VisitUnaryOperator in RMFPV ✓
#1433 by PetroZarytskyi was merged on Jul 3 · Approved

compiler-research/compiler-research.github.io Update information about Petro Zarytskyi ✗
#317 by PetroZarytskyi was merged on Jul 3



vgvassilev/clad Move VarData element access to VarData::operator[] ✓
#1425 by PetroZarytskyi was merged on Jul 1 · Approved

vgvassilev/clad Rework SetIsReq and overlay and combine them into a single function ✓
#1424 by PetroZarytskyi was merged on Jun 29 · Approved

vgvassilev/clad Move ProfileID generation to a single point in TBR ✓
#1422 by PetroZarytskyi was merged on Jun 26 · Approved

vgvassilev/clad Switch from Visit to Traverse functions in TBR ✓
#1421 by PetroZarytskyi was merged on Jun 25 · Approved

vgvassilev/clad Perform lookups of reference variables in TBR::overlay ✓
#1419 by PetroZarytskyi was merged on Jun 24 · Approved

vgvassilev/clad Simplify `getExprVarData` in TBR and add a test ✓
#1418 by PetroZarytskyi was merged on Jun 24 · Approved

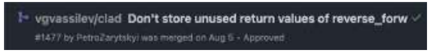
vgvassilev/clad Update call argument storing in the reverse mode ✓
#1416 by PetroZarytskyi was merged on Jun 20 · Approved

vgvassilev/clad Move some tests from UserDefinedTypes.C to Constructor.C ✓
#1412 by PetroZarytskyi was merged on Jun 16 · Approved

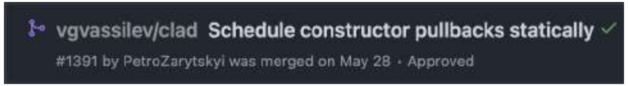
vgvassilev/clad Schedule simple partial derivatives statically ✗
#1408 by PetroZarytskyi was merged on Jun 15 · Approved

vgvassilev/clad Update the SmallPT ray tracer demo ✓
#1407 by PetroZarytskyi was merged on Jun 16 · Approved

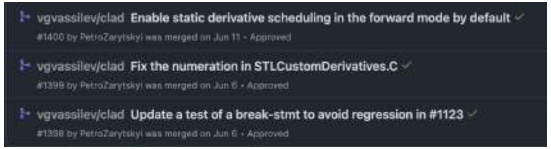
vgvassilev/clad Call UpdateDiffParamsInfo in the Diffplanner unconditionally ✓
#1406 by PetroZarytskyi was merged on Jun 14 · Approved



vgvassilev/clad Don't store unused return values of reverse_forw ✓
#1477 by PetroZarytskyi was merged on Aug 5 · Approved



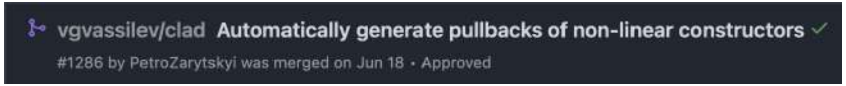
vgvassilev/clad Schedule constructor pullbacks statically ✓
#1391 by PetroZarytskyi was merged on May 28 · Approved



vgvassilev/clad Enable static derivative scheduling in the forward mode by default ✓
#1400 by PetroZarytskyi was merged on Jun 11 · Approved

vgvassilev/clad Fix the numeration in STLCustomDerivatives.C ✓
#1399 by PetroZarytskyi was merged on Jun 6 · Approved

vgvassilev/clad Update a test of a break-stmt to avoid regression in #1123 ✓
#1398 by PetroZarytskyi was merged on Jun 6 · Approved



vgvassilev/clad Automatically generate pullbacks of non-linear constructors ✓
#1286 by PetroZarytskyi was merged on Jun 18 · Approved

Summary:
Improvements in TBR analysis

- **Support for pointers**: new operators, address and dereference operators (& and *), pointers are no longer always to-be-recorded.
- **Support for nested function calls**: information about variables is propagated between functions
- **Improved OOP support** leading to **17% less produced code** and **up to 44% speedup** in OOP-specialized tests in Clad**.**
- **Support for local declarations**: in the reverse mode, they also require tapes and can be optimized by TBR
- **Major code refactoring** for better readability/maintainability (fixed some bugs automatically)

Summary:
Improvements in OOP

- **Automatic differentiation of constructors**: general constructors are now supported (before, only linear), base-class constructors, delegating constructors, move-constructors – **constructor-exprs are supported on the same level as function call-exprs**
- **Improved reverse_forw functionality**: fixed bugs and crashes, improved produced code, improved pointer support
- **Diagnostics for custom derivatives**: hints for the user if the custom derivative is written incorrectly and cannot be used by Clad.
- **Other bugs/features**: added support for **std::pair, std::unique_ptr, std::shared_ptr, std::weak_ptr**, etc.
- **Related code maintenance**: mostly moved to static scheduling of derivatives, etc.

Thank you!
Questions?