Use of automatic differentiation (AD) in OPM

Atgeirr Flø Rasmussen

SINTEF ICT, Dept. Applied Mathematics

12th March 2015

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

What next?

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

What next?

What does AD provide

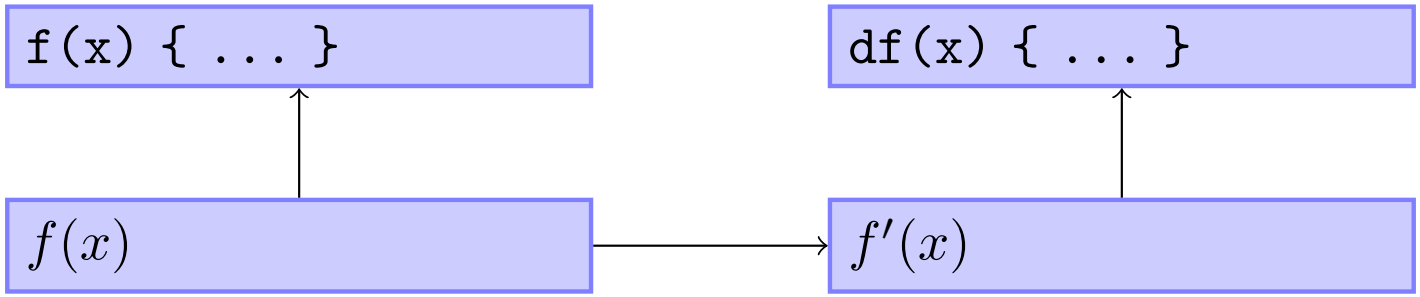
```
f(x) { ... }
```

```
df(x) { ... }
```

$f(x)$

$f'(x)$

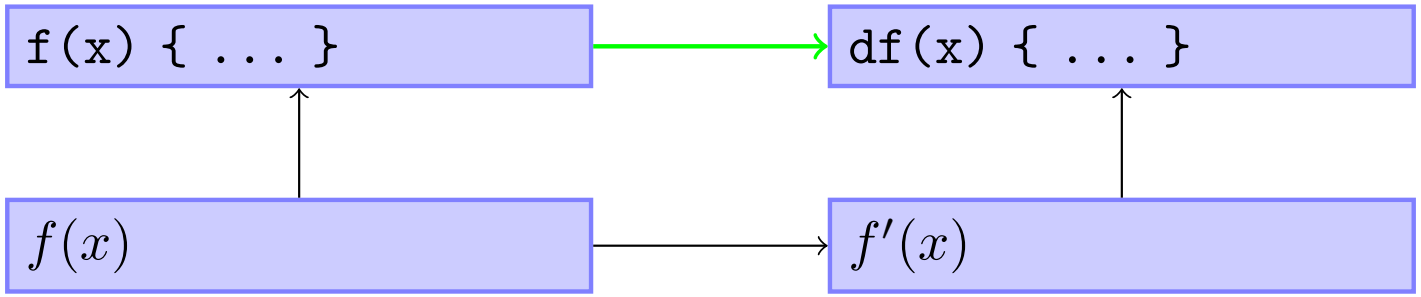
What does AD provide



Traditional Process

- Human implements code to evaluate $f(x)$
- Manual or symbolic calculation to derive $f'(x)$
- Human implements code to evaluate $f'(x)$

What does AD provide



Traditional Process

- Human implements code to evaluate $f(x)$
- Manual or symbolic calculation to derive $f'(x)$
- Human implements code to evaluate $f'(x)$

Automatic Differentiation (AD)

- Human implements code to evaluate $f(x)$
- Computer code to evaluate $f'(x)$ is automatically generated

Benefits of using AD

AD makes it easier to create simulators:

- only specify nonlinear residual equation
- automatically evaluates Jacobian
- sparsity structure of Jacobian automatically generated

Note that AD is *not* the same as finite differencing!

- no need to define a 'small' epsilon
- as precise as hand-made Jacobian
- ... but much less work!

Performance (of equation assembly) will usually be somewhat slower than a *good* hand-made Jacobian implementation.

Basic idea

A numeric computation $y = f(x)$ can be written (D = derivative)

$$
\begin{aligned}
y_1 &= f_1(x) & \frac{dy_1}{dx}(x) &= Df_1(x) \\
y_2 &= f_2(y_1) & \frac{dy_2}{dx}(x) &= Df_2(y_1) \cdot Df_1(x) \\
&\vdots \\
y &= f_n(y_{n-1}) & \frac{dy}{dx}(x) &= Df_n(y_{n-1}) \cdot Df_{n-1}(y_{n-2}) \cdots Df_1(x)
\end{aligned}
$$

Automatic Differentiation:

- make each line an elementary operation
- compute right derivative values as we go using chain rule

Implementation approaches

Two main methods:

Operator overloading

- requires operator overloading in programming language
- syntax (more or less) like before (non-AD)
- efficiency can vary a lot, depends on usage scenario
- easy to implement and experiment with
- Examples: opm-autodiff, Sacado (Trilinos), ADOL-C

Source transformation with AD tool

- can be implemented for almost any language
- may restrict language syntax or features used
- efficiency can be high (depends on AD tool)
- Examples: TAPENADE, OpenAD

Types of AD

Two different approaches.
(We compute $f(x)$, u is some intermediate variable.)

Forward Mode

Carry derivatives with respect to independent variables:

$$(u, \frac{du}{dx})$$

Reverse Mode

Carry derivatives with respect to dependent variables (adjoints):

$$(u, \frac{df}{du})$$

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

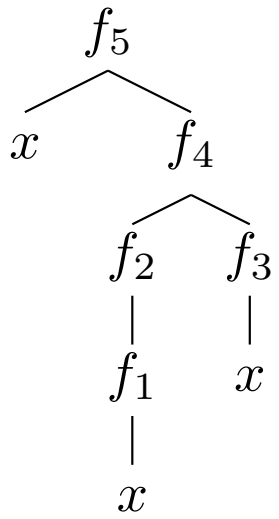
What next?

Forward AD example (1)

Example function: $f(x) = x(sin(x^2) + 3x)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

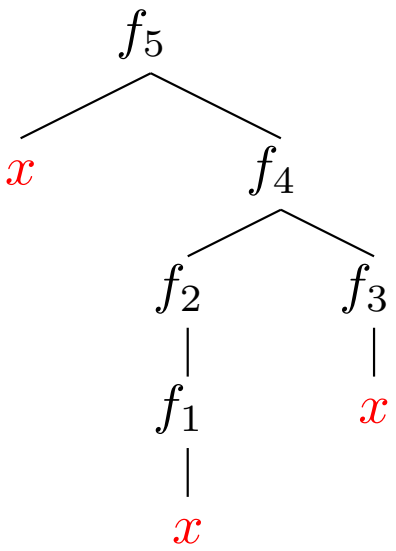
Rewritten:
$f(x) = f_5(x, f_4(f_2(f_1(x)), f_3(x)))$

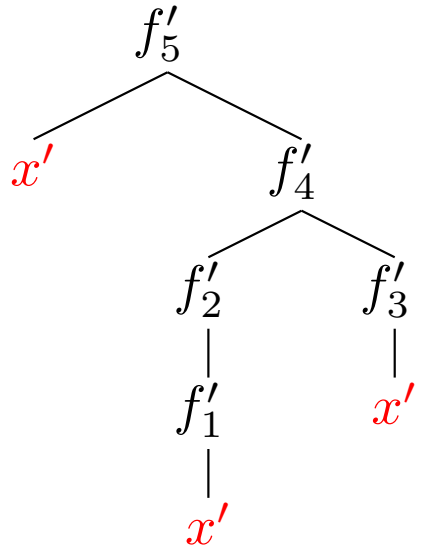


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

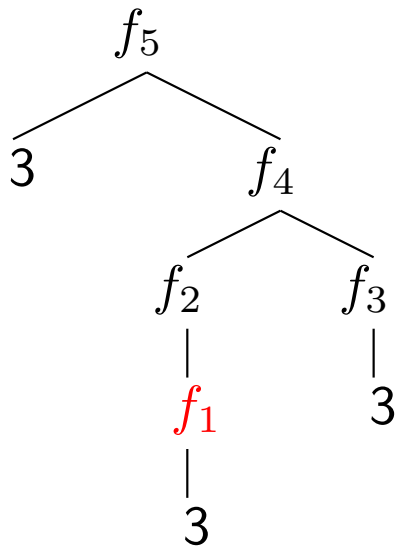


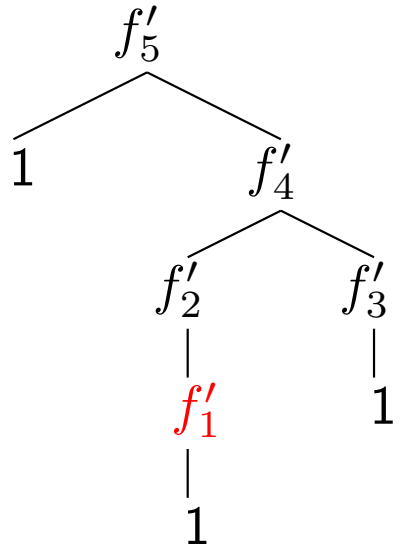


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

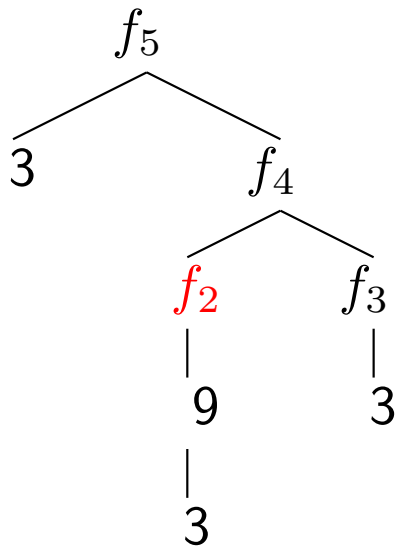


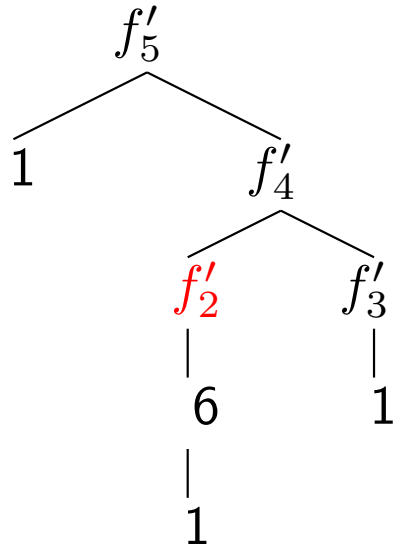


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

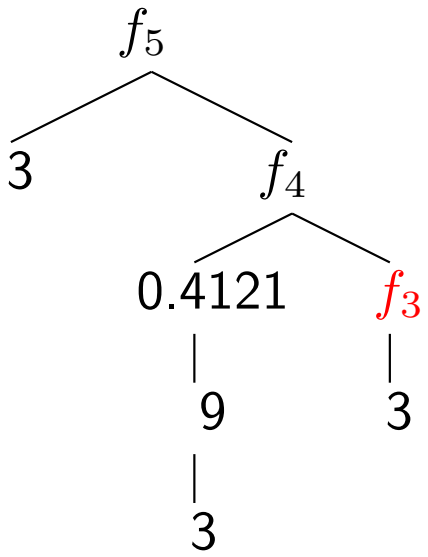


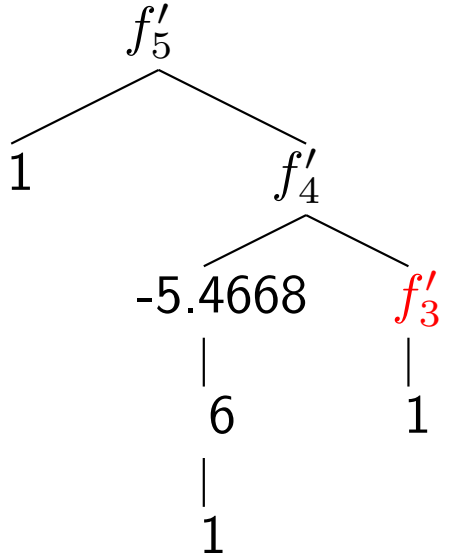


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

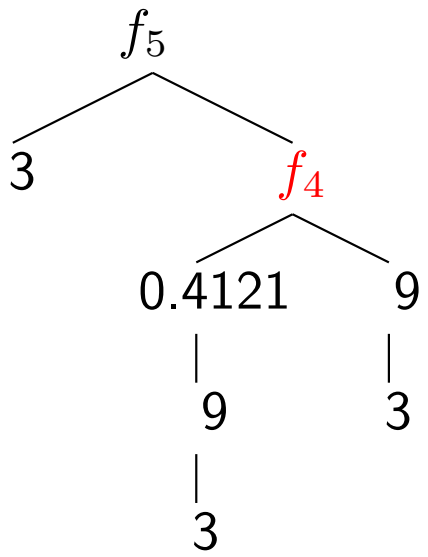


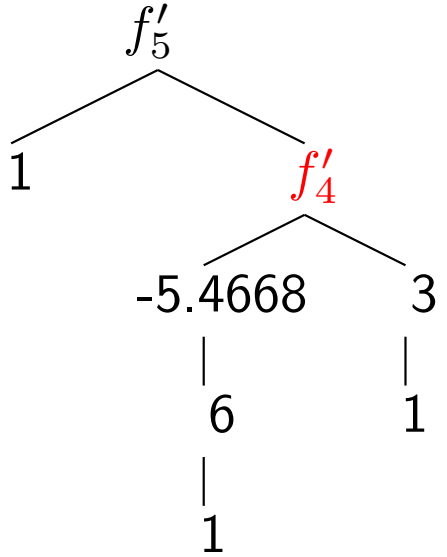


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

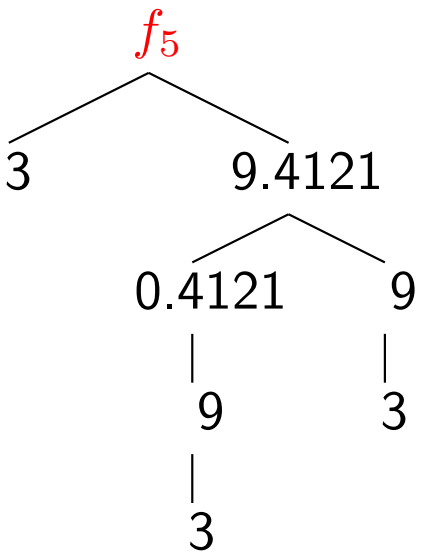


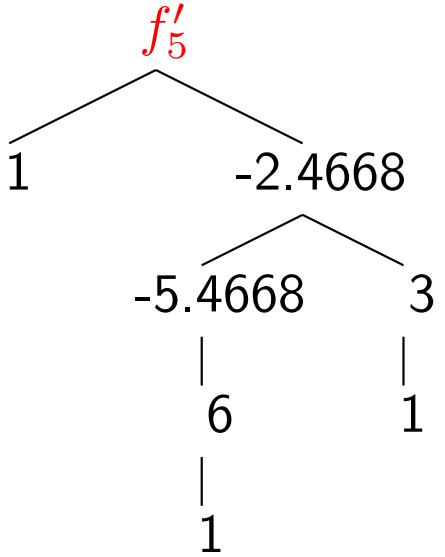


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$

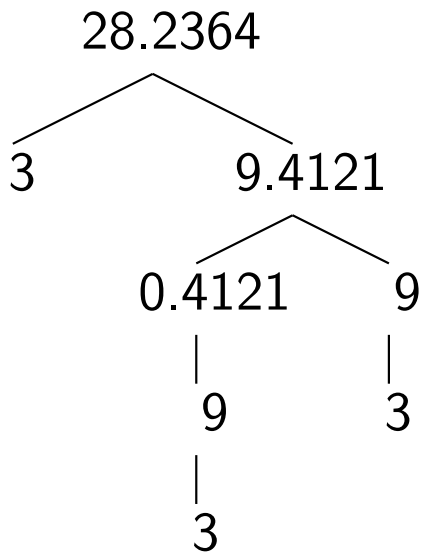


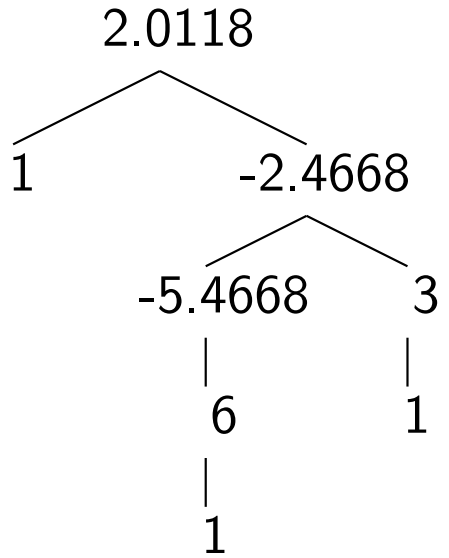


Forward AD example (2)

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & f_1'(u) &= 2uu' \\
f_2(u) &= \sin(u) & f_2'(u) &= \cos(u)u' \\
f_3(u) &= 3u & f_3'(u) &= 3u' \\
f_4(u,v) &= u + v & f_4'(u,v) &= u' + v' \\
f_5(u,v) &= u \cdot v & f_5'(u,v) &= u'v + uv'
\end{aligned}
$$





Properties of forward AD

- Easy to implement with operator overloading
- Storage required (scalar): $2\times$ normal (value, derivative).
- Storage required ($f : R^m \to R^n$): $(n+1)\times$ normal (value, derivative vector), unless sparse.

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

What next?

Reverse Mode AD

Recall: Reverse Mode

Carry derivatives with respect to dependent variables (adjoints):

$$(u, \frac{df}{du})$$

We will use the chain rule again, but in the opposite direction:

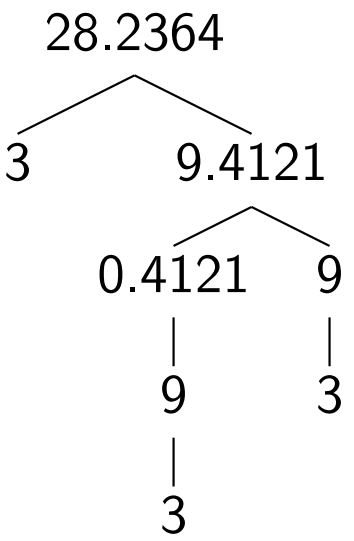
$$\mathrm{adj}(u) = \mathrm{adj}(f_i)\frac{\partial f_i}{\partial u}.$$

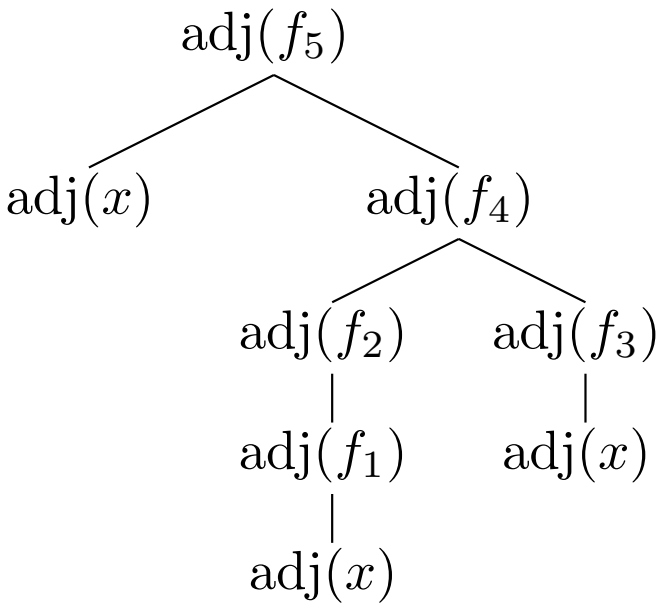
Using $\mathrm{adj}(u)$ to mean the adjoint $\frac{df}{du}$.
(So $\mathrm{adj}(x)$ is our goal.)

Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

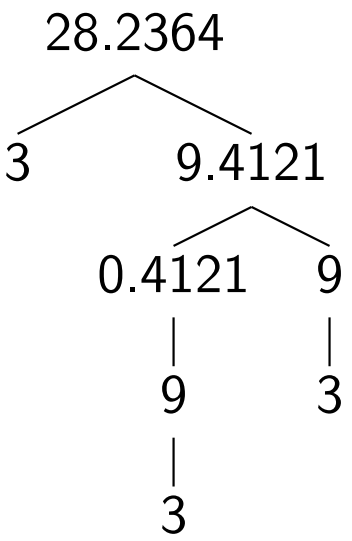


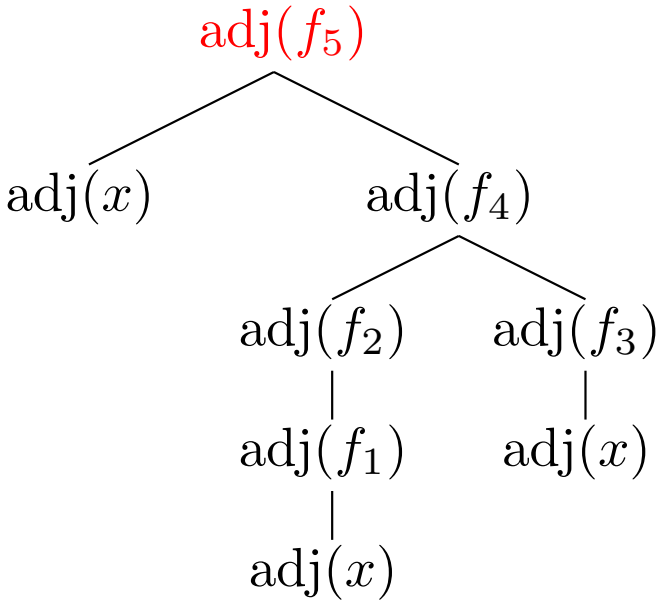


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

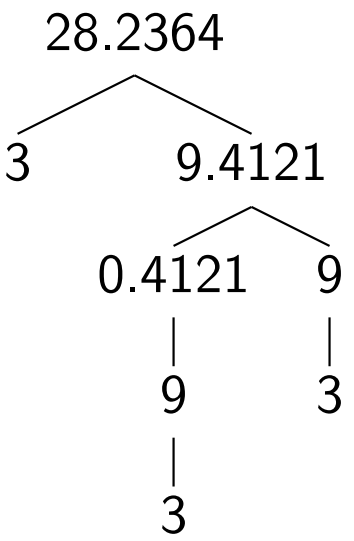


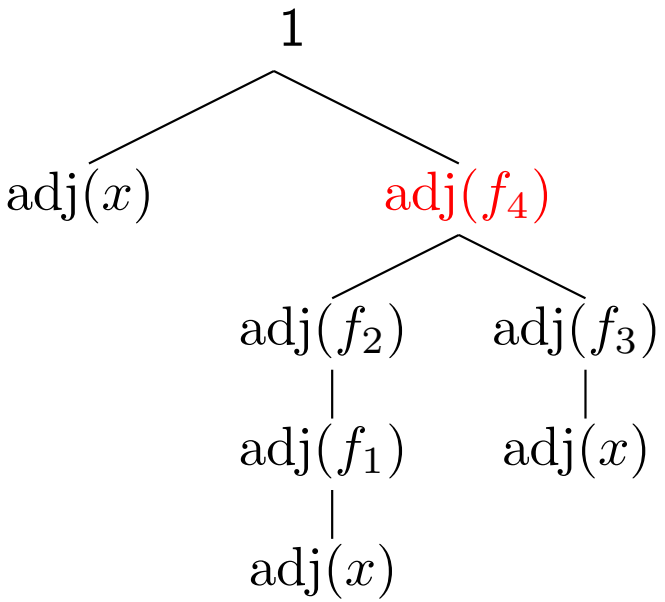


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

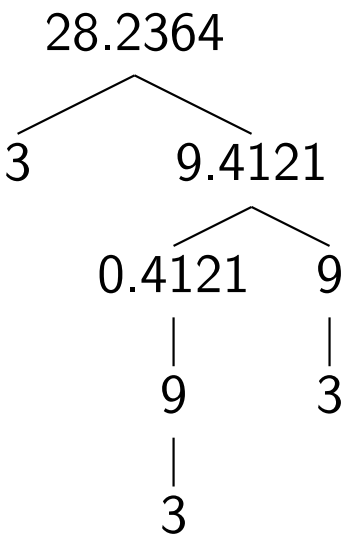


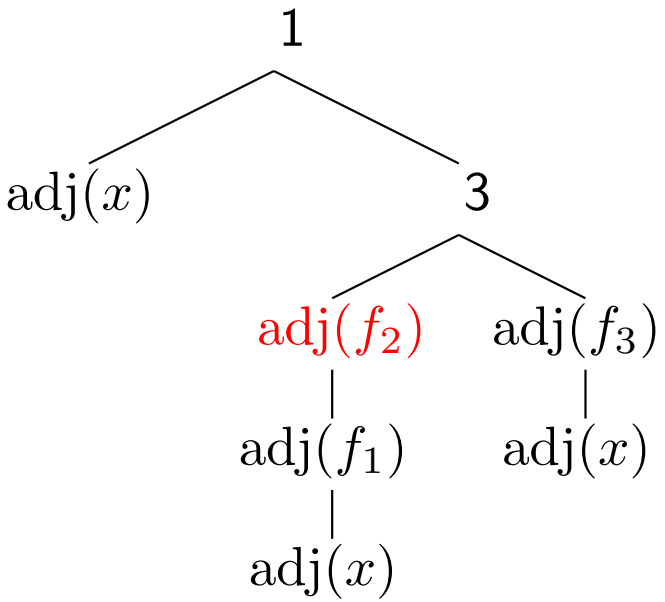


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

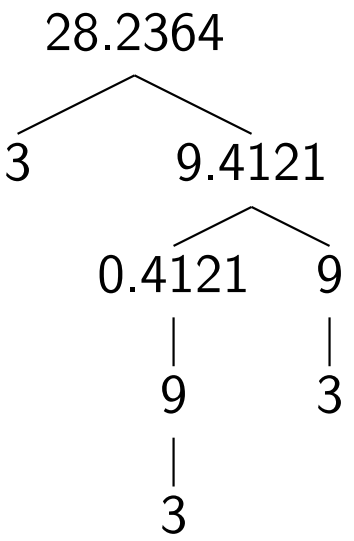


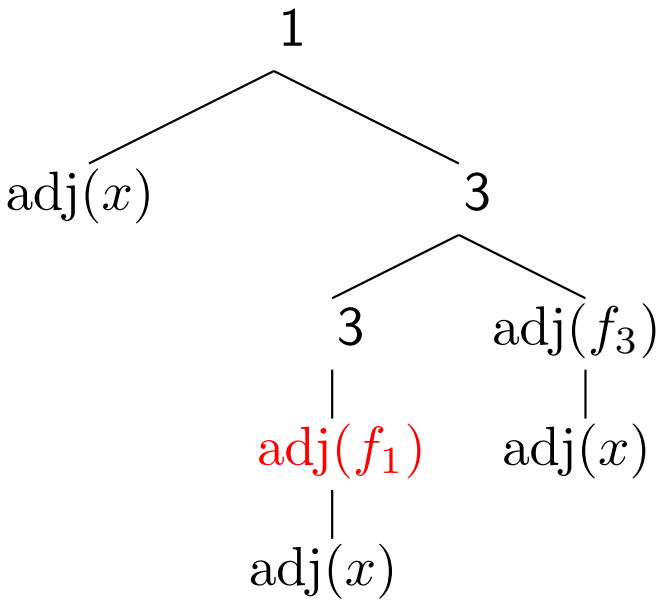


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

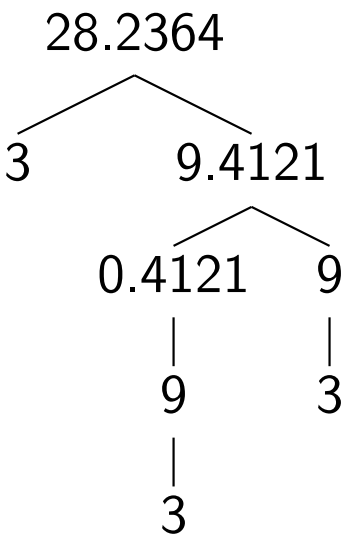


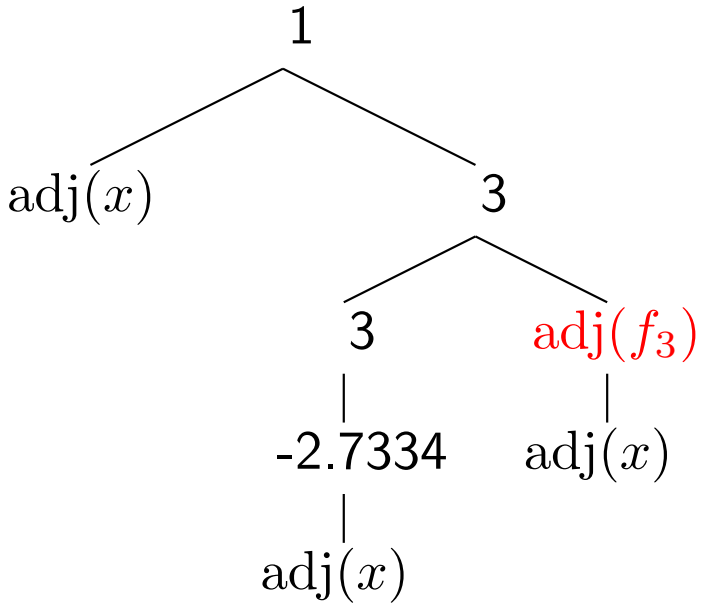


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 && \mathrm{adj}(u) = \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) && \mathrm{adj}(u) = \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u && \mathrm{adj}(u) = \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v && \mathrm{adj}(u) = \mathrm{adj}(f_4), && \mathrm{adj}(v) = \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v && \mathrm{adj}(u) = \mathrm{adj}(f_5) \cdot v, && \mathrm{adj}(v) = \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

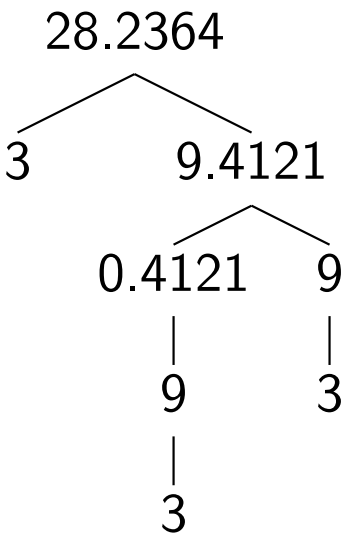


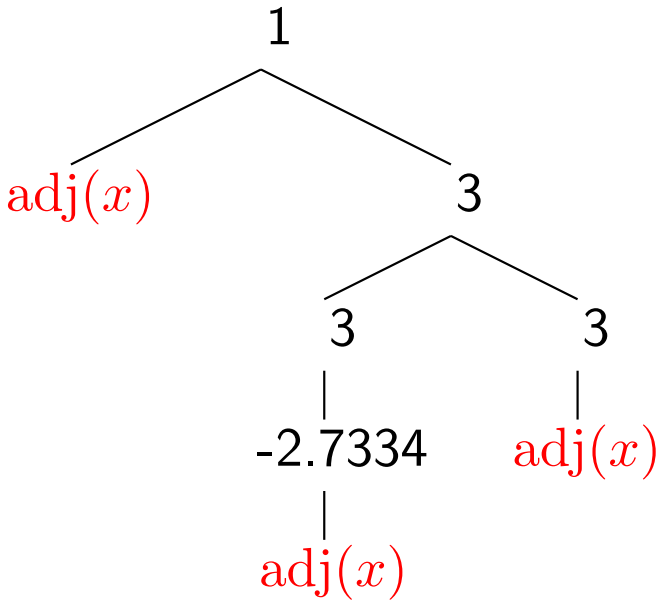


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

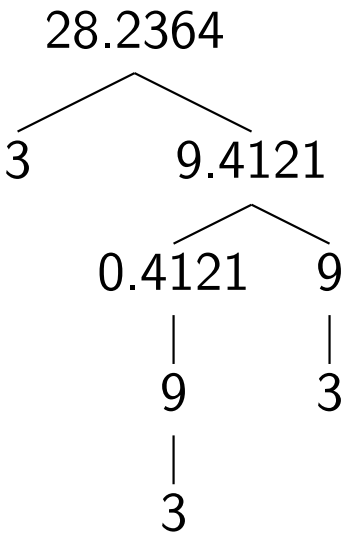


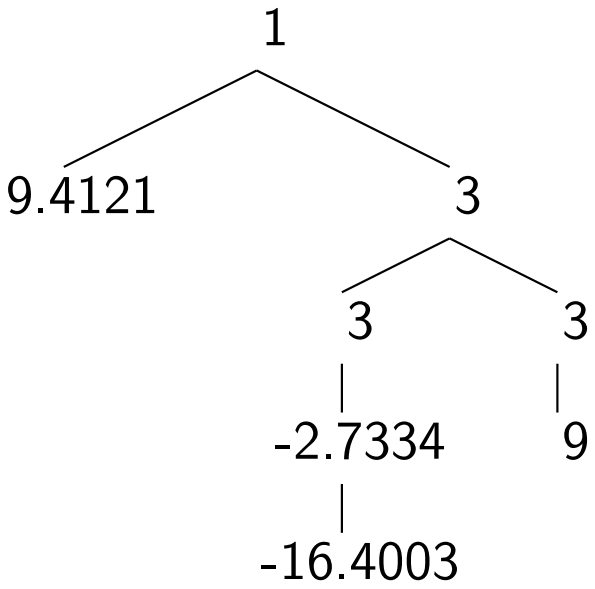


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$

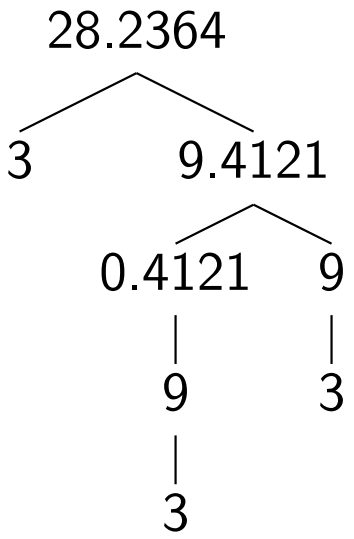


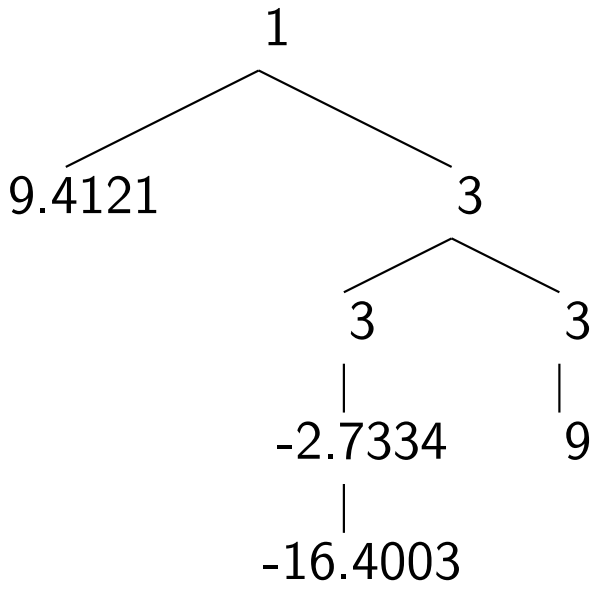


Reverse AD example

Example function: $f(x) = x(sin(x^2) + 3x)$. Computing $f(3), f'(3)$.
Sequence of elementary functions:

$$
\begin{aligned}
f_1(u) &= u^2 & \mathrm{adj}(u) &= \mathrm{adj}(f_1) \cdot 2u \\
f_2(u) &= \sin(u) & \mathrm{adj}(u) &= \mathrm{adj}(f_2) \cdot \cos(u) \\
f_3(u) &= 3u & \mathrm{adj}(u) &= \mathrm{adj}(f_3) \cdot 3 \\
f_4(u,v) &= u + v & \mathrm{adj}(u) &= \mathrm{adj}(f_4), & \mathrm{adj}(v) &= \mathrm{adj}(f_4) \\
f_5(u,v) &= u \cdot v & \mathrm{adj}(u) &= \mathrm{adj}(f_5) \cdot v, & \mathrm{adj}(v) &= \mathrm{adj}(f_5) \cdot u
\end{aligned}
$$





Must sum contributions:
$f'(3) = 9.4121 - 16.4003 + 9 = 2.0118.$

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

What next?

Automatic Differentiation: OPM implementation

AutoDiffBlock

- class implementing *forward AD* in OPM
- deals with *vectors of data* at a time
- derivatives are *sparse matrices*
- implemented with operator overloading
- based on Eigen library for basic types and operands
- helper library provides discrete operators

Automatic Differentiation: library example

```
/// Elementwise operator *
AutoDiffBlock operator*(const AutoDiffBlock& rhs) const
{
    ... // Omitted some code dealing with constants
    int num_blocks = numBlocks();
    std::vector<M> jac(num_blocks);
    assert(numBlocks() == rhs.numBlocks());
    typedef Eigen::DiagonalMatrix<Scalar, Eigen::Dynamic> D;
    D D1 = val_.matrix().asDiagonal();
    D D2 = rhs.val_.matrix().asDiagonal();
    for (int block = 0; block < num_blocks; ++block) {
        assert(jac_[block].rows() == rhs.jac_[block].rows());
        assert(jac_[block].cols() == rhs.jac_[block].cols());
        if( jac_[block].nonZeros() == 0 && rhs.jac_[block].nonZeros() == 0 ) {
            jac[block] = M( D2.rows(), jac_[block].cols() );
        }
        else if( jac_[block].nonZeros() == 0 )
            jac[block] = D1*rhs.jac_[block];
        else if ( rhs.jac_[block].nonZeros() == 0 ) {
            jac[block] = D2*jac_[block];
        }
        else {
            jac[block] = D2*jac_[block] + D1*rhs.jac_[block];
        }
    }
    return function(val_ * rhs.val_, std::move(jac));
}
```

Automatic Differentiation: usage example

Black-oil Mass Balance Equations (w/dissolved gas)

$$\begin{cases} \partial_t(\phi b_w s_w) + \operatorname{div}(b_w v_w) = q_w \\ \partial_t(\phi b_o s_o) + \operatorname{div}(b_o v_o) = q_o \\ \partial_t(\phi(b_g s_g + R_s b_o s_o)) + \operatorname{div}(b_g v_g + R_s b_o v_o) = q_g \end{cases}$$

Automatic Differentiation: usage example

Black-oil Mass Balance Equations (w/dissolved gas)

$$\begin{cases} \partial_t(\phi b_w s_w) + \operatorname{div}(b_w v_w) = q_w \\ \partial_t(\phi b_o s_o) + \operatorname{div}(b_o v_o) = q_o \\ \partial_t(\phi(b_g s_g + R_s b_o s_o)) + \operatorname{div}(b_g v_g + R_s b_o v_o) = q_g \end{cases}$$

OPM Implementation (almost: omit dissolved gas for space)

p_o, s_w, s_g, R_s and R_v can be primary variables (depending on state in each cell)

```
...
accum[phase] =   pv_mult * b[phase] * sat[phase]
               - pv_mult0 * b0[phase] * sat0[phase];
...
dp = ops_.ngrad * pressure[phase]
tdp = transmissibilities * dp;
UpwindSelector<double> upwind(grid_, ops_, tdp.value());
mflux[phase] = upwind.select(b[phase] * mob[phase]) * tdp;
...
material_balance[phase] =
    (pv/dt) * accum[phase] + ops_.div * mflux[phase];
```

Residuals in `material_balance.value()` and the associate Jacobians in `material_balance.derivative()`.

Overview of talk

Why automatic differentiation?

Forward Mode AD

Reverse Mode AD

OPM implementation

What next?

Current issues

- Assembly performance (setting up linear systems) is worse than competition.
- Vectorized notation is great for some, perhaps not for all?

Outlook and questions

- Investigate using backward mode (some or all parts).
- Is it possible to avoid vectorization penalty (few math ops per memory access)?
- Can we gain by using expression templates?
- Look at using other implementations? ADETL?
- Alternative techniques?

Thank you for listening!