A simple introduction to automatic differentiation (AD)

Atgeirr Flø Rasmussen

SINTEF Digital, Mathematics and Cybernetics

May 28, 2021

Availability of materials

The slides and programming examples are available on GitHub:

`https://github.com/atgeirr/SimpleAD`

What does AD provide

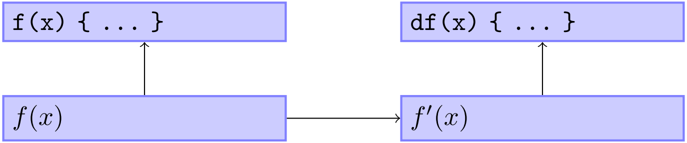
`f(x) { ... }`

`df(x) { ... }`

$f(x)$

$f'(x)$

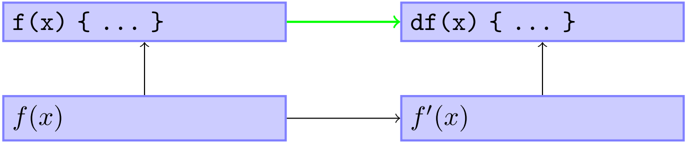
What does AD provide



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) may be slower than a *good* hand-made Jacobian implementation.

- We have also seen the opposite
- The more complex the equations, the better is AD

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, MRST, 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):

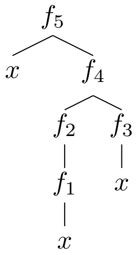
$$(u, \frac{df}{du})$$

Forward AD example (1)

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

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

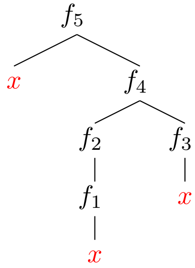
Rewritten:
$f(x) = f_5(x, f_4(f_2(f_1(x)), f_3(x)))$

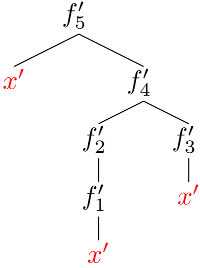


Forward AD example (2)

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

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

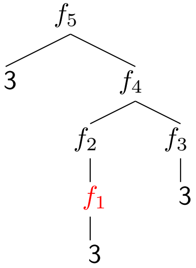


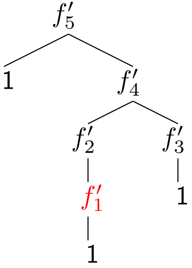


Forward AD example (2)

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

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

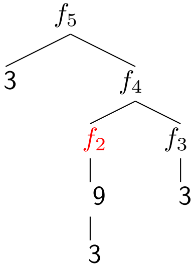


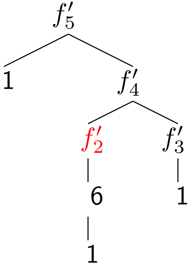


Forward AD example (2)

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

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

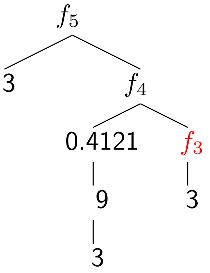


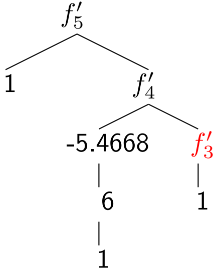


Forward AD example (2)

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

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

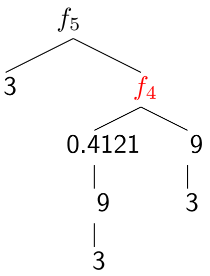


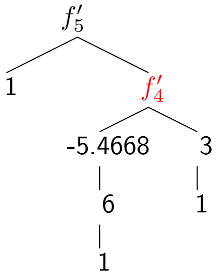


Forward AD example (2)

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

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

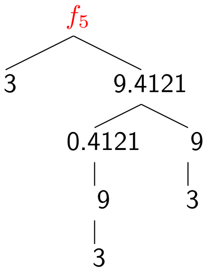


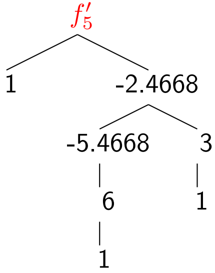


Forward AD example (2)

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

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

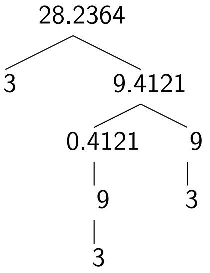


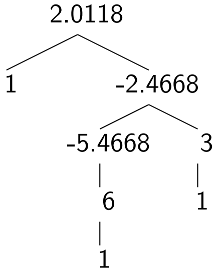


Forward AD example (2)

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.

Automatic Differentiation: OPM implementations

`class Evaluation`

- class implementing *forward AD*
- deals with *a single scalar value* at a time
- derivatives are *compile-time-size vectors*
- implemented with operator overloading
- discrete div, grad must be implemented “manually”

Dealing with off-diagonal Jacobian entries

- Create the non-zero structure of the Jacobian upfront
- Focus on one cell’s variables at a time
- Compute all terms depending on these
- Accumulate derivatives in correct Jacobian column
- Requires calculating fluxes twice
  - Value the same, but derivatives are different (w.r.t. each neighbour cell)

A simple (forward) AD example class

```cpp
class SimpleAd
{
private:
    double val_; // The value (corresponding to a regular double)
    double der_; // The derivative (of this variable)
 public:
    SimpleAd(double val, double der) : val_(val), der_(der) {}
    double value()       const { return val_; }
    double derivative() const { return der_; }
    SimpleAd operator+(const SimpleAd& rhs) const
    {
        // Derivative of sum is sum of derivatives.
        return { val_ + rhs.val_, der_ + rhs.der_ };
    }
    SimpleAd operator*(const SimpleAd& rhs) const
    {
        // Derivative of product follows well-known product rule.
        return { val_ * rhs.val_, der_ * rhs.val_ + val_ * rhs.der_ };
    }
};
```

Exercises (part 1: making do without AD)

These exercises assume the file newtonexample-exercise.cpp is available.

1. The example file contains a simple Newton's method that requires both a function and its derivative to be passed. Read and understand the function `newtonUpdate()`. *Compile and run the example, verifying that it produces the expected result.*
2. Changing the function, you must also change the derivative function. *Uncomment "part 2" and fix the derivative until the example again compiles and runs successfully.*

Exercises (part 2: using AD)

These exercises assume the file adexample-exercise.cpp is available.

1. The example file contains a simple Newton's method that uses AD for its implementation. Read and understand `newtonUpdate()`. *Compile and run the example, verifying that it produces the expected result.*
2. Functions such as sine, cosine and exponential require special treatment. *Using the provided sin() function as an example, implement a cos() function.*
3. There are still operators missing for more general expressions. *Uncomment "part 2" and add missing features until the example again compiles and runs successfully.*
4. An AD class must handle expressions containing raw doubles, either by a) including appropriate operators, or b) implicit conversions to the AD type. *What approach has been used for SimpleAD in the example file? How would the alternative approach have been implemented?*

Thank you for listening!

Bonus: 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:

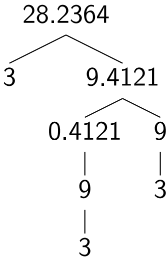
$$\mathrm{adj}(u) = \mathrm{adj}(f_i)\frac{\partial f_i}{\partial u}.$$

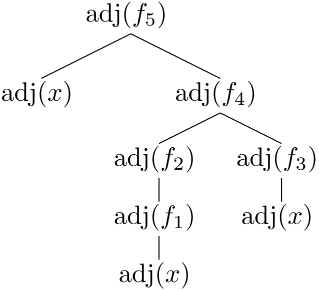
Using $\mathrm{adj}(u)$ to mean the adjoint $\frac{df}{du}$.
(So $\mathrm{adj}(x)$ is our goal.)

Reverse AD example

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

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

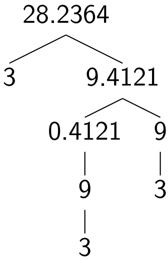


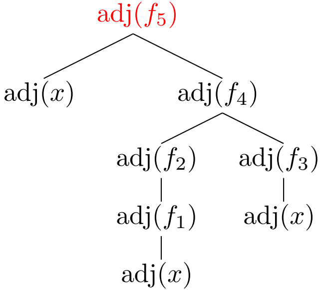


Reverse AD example

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

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

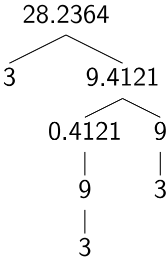


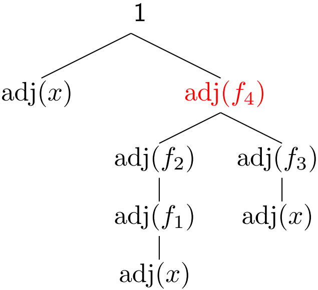


Reverse AD example

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

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

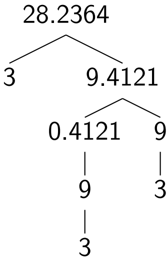


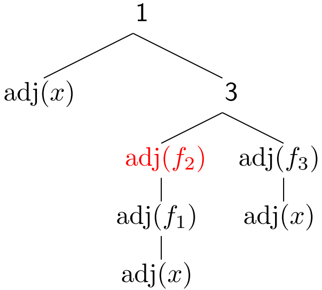


Reverse AD example

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

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

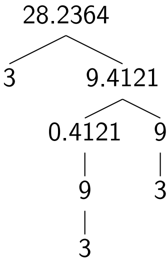


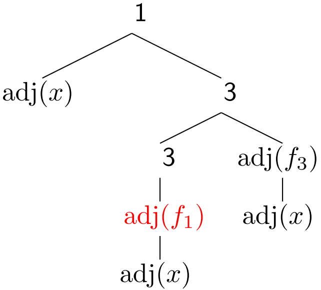


Reverse AD example

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

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

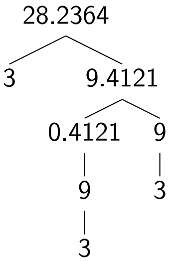


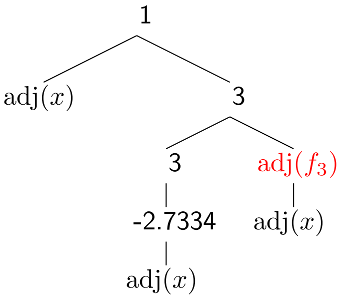


Reverse AD example

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

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

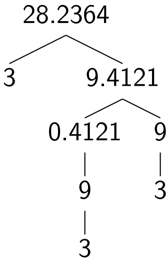


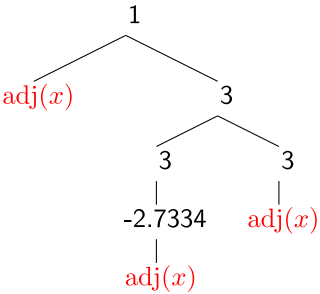


Reverse AD example

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

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

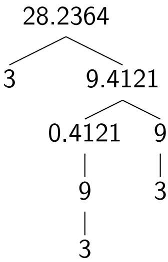


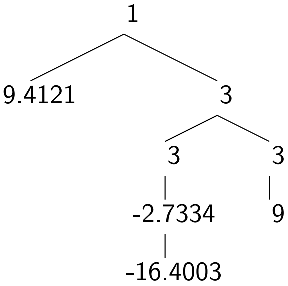


Reverse AD example

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

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

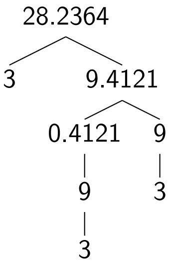


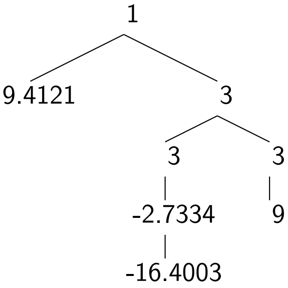


Reverse AD example

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

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





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