# Derivative-based Optimization for Applications in Physics

Andrea Walther
Institut für Mathematik
Humboldt-Universität zu Berlin and ZIB

Keynote Lecture
4th MODE Workshop on Differentiable Programming for Experimental Design
September 24, 2024

Berlin Mathematics Research Center
MATH+

# Outline

# Where are Derivatives Needed?

- Optimization:

unbounded: $\min f(x), \quad f: \mathbb{R}^n \to \mathbb{R}$

bounded: $\min f(x), \quad f: \mathbb{R}^n \to \mathbb{R}, \quad c(x) = 0,\ c: \mathbb{R}^n \to \mathbb{R}^m, \quad h(x) \leq 0,\ h: \mathbb{R}^n \to \mathbb{R}^l$

# Where are Derivatives Needed?

- Optimization:

  unbounded: $\min f(x), \quad f : \mathbb{R}^n \to \mathbb{R}$

  bounded: $\min f(x), \quad f : \mathbb{R}^n \to \mathbb{R}, \quad c(x) = 0, c : \mathbb{R}^n \to \mathbb{R}^m, \quad h(x) \leq 0, h : \mathbb{R}^n \to \mathbb{R}^l$

- Solution of nonlinear equation systems, i.e., $F(x) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^n$
  Newton method requires $F'(x) \in \mathbb{R}^{n \times n}$

# Where are Derivatives Needed?

- Optimization:

  unbounded: $\min f(x), \quad f:\mathbb{R}^n \to \mathbb{R}$
  bounded: $\min f(x), \quad f:\mathbb{R}^n \to \mathbb{R}, \quad c(x) = 0,\ c:\mathbb{R}^n \to \mathbb{R}^m, \quad h(x) \leq 0,\ h:\mathbb{R}^n \to \mathbb{R}^l$

- Solution of nonlinear equation systems, i.e., $F(x) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^n$
  Newton method requires $F'(x) \in \mathbb{R}^{n\times n}$
- Simulation of complex system
  - definition
  - integration of differential equations using implicit methods

# Where are Derivatives Needed?

- Optimization:

  unbounded: $\min f(x)$, $f : \mathbb{R}^n \to \mathbb{R}$
  bounded: $\min f(x)$, $f : \mathbb{R}^n \to \mathbb{R}$, $c(x) = 0, c : \mathbb{R}^n \to \mathbb{R}^m$, $h(x) \leq 0, h : \mathbb{R}^n \to \mathbb{R}^l$

- Solution of nonlinear equation systems, i.e., $F(x) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^n$
  Newton method requires $F'(x) \in \mathbb{R}^{n \times n}$
- Simulation of complex system
  - definition
  - integration of differential equations using implicit methods
- Sensitivity analysis
- Real-time control

# Where are Derivatives Needed?

- Optimization:

  unbounded: $\min f(x), \quad f : \mathbb{R}^n \to \mathbb{R}$
  bounded: $\min f(x), \quad f : \mathbb{R}^n \to \mathbb{R}, \quad c(x) = 0, c : \mathbb{R}^n \to \mathbb{R}^m, \quad h(x) \le 0, h : \mathbb{R}^n \to \mathbb{R}^l$

- Solution of nonlinear equation systems, i.e., $F(x) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^n$
  Newton method requires $F'(x) \in \mathbb{R}^{n \times n}$
- Simulation of complex system
  - definition
  - integration of differential equations using implicit methods
- Sensitivity analysis
- Real-time control
- ML, e.g., Stochastic Gradient Descent, Adam, . . .
  target functions quite often nonsmooth!

# Frequent Situation

# Computing Derivatives

**Given:**

Description of functional relation as

- formula $y = F(x)$ $\Rightarrow$ explicit expression $y' = F'(x)$
- computer program $\Rightarrow$ ?

# Computing Derivatives

**Given:**

Description of functional relation as

- formula $y = F(x)$ $\Rightarrow$ explicit expression $y' = F'(x)$
- computer program $\Rightarrow$ ?

**Task:**

Computation of derivatives taking

- requirements on exactness
- computational effort

into account



Berlin Mathematics Research Center

# Algorithmic Differentiation (AD)

aka Automatic Differentiation

= Differentiation of computer programs implementing $F : \mathbb{R}^n \mapsto \mathbb{R}^m$

# Algorithmic Differentiation (AD)

aka Automatic Differentiation

= Differentiation of computer programs implementing $F : \mathbb{R}^n \mapsto \mathbb{R}^m$

**Main Products:**

- Quantitative dependence information (local):
  - Weighted and directed partial derivatives
  - Error and condition number estimates . . .
  - Lipschitz constants, interval enclosures . . .

# Algorithmic Differentiation (AD)

aka Automatic Differentiation

= Differentiation of computer programs implementing $F : \mathbb{R}^n \mapsto \mathbb{R}^m$

**Main Products:**

- Quantitative dependence information (local):
  - Weighted and directed partial derivatives
  - Error and condition number estimates . . .
  - Lipschitz constants, interval enclosures . . .
- Qualitative dependence information (regional):
  - Sparsity structures, degrees of polynomials
  - Ranks, eigenvalue multiplicities . . .

# Algorithmic Differentiation (AD)

aka Automatic Differentiation

= Differentiation of computer programs implementing $F : \mathbb{R}^n \mapsto \mathbb{R}^m$

**Main Products:**

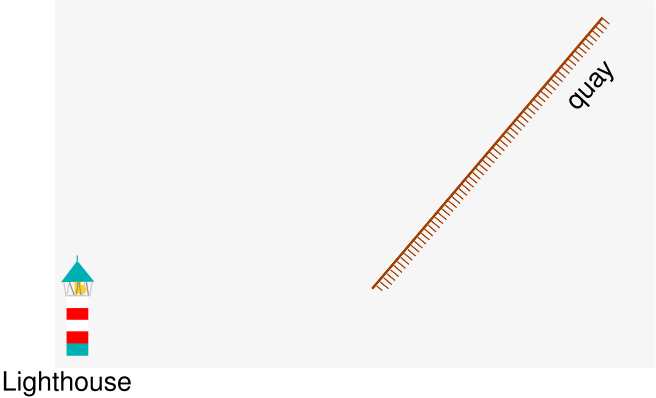
- Quantitative dependence information (local):
  - Weighted and directed partial derivatives
  - Error and condition number estimates . . .
  - Lipschitz constants, interval enclosures . . .
- Qualitative dependence information (regional):
  - Sparsity structures, degrees of polynomials
  - Ranks, eigenvalue multiplicities . . .

Assumption: $F$ differentiable in a neighbourhood of current argument $x$

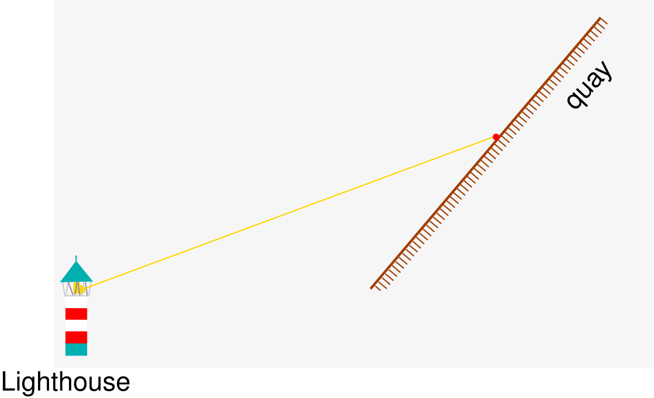


Berlin Mathematics Research Center

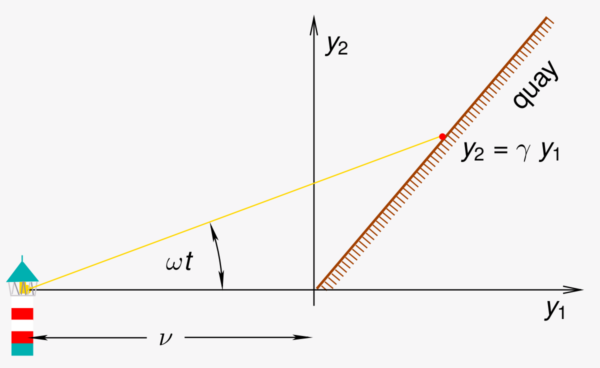
# The “Hello-World”-Example of AD



# The "Hello-World"-Example of AD

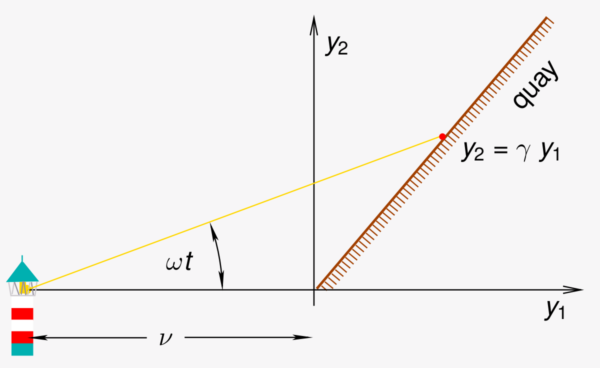


# The “Hello-World”-Example of AD




Lighthouse

# The “Hello-World”-Example of AD




Lighthouse

$$y_1 = \frac{\nu \tan(\omega\, t)}{\gamma - \tan(\omega\, t)} \quad \text{and} \quad y_2 = \frac{\gamma\, \nu \tan(\omega\, t)}{\gamma - \tan(\omega\, t)}$$

# Evaluation Procedure (Lighthouse)

$$y_1 = \frac{\nu \tan(\omega\, t)}{\gamma - \tan(\omega\, t)}$$

$$y_2 = \frac{\gamma\, \nu \tan(\omega\, t)}{\gamma - \tan(\omega\, t)}$$

$$\Longrightarrow$$

$$\begin{array}{lll}
v_{-3} & = x_1 = \nu & \\
v_{-2} & = x_2 = \gamma & \\
v_{-1} & = x_3 = \omega & \\
v_0 & = x_4 = t & \\
\hline
v_1 & = v_{-1} * v_0 & \equiv \varphi_1(v_{-1}, v_0) \\
v_2 & = \tan(v_1) & \equiv \varphi_2(v_1) \\
v_3 & = v_{-2} - v_2 & \equiv \varphi_3(v_{-2}, v_2) \\
v_4 & = v_{-3} * v_2 & \equiv \varphi_4(v_{-3}, v_2) \\
v_5 & = v_4 / v_3 & \equiv \varphi_5(v_4, v_3) \\
v_6 & = v_5 * v_{-2} & \equiv \varphi_6(v_5, v_{-2}) \\
\hline
y_1 & = v_5 & \\
y_2 & = v_6 &
\end{array}$$

# Function Evaluation in ML

Typical function evaluation (deep neural net):

Propagation of one data point:

$$
\begin{aligned}
x = x^{(1)} &\rightarrow \tilde{x}^{(1)} = W^{(1)} x^{(1)} + b^{(1)} && \rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \\
&\rightarrow \tilde{x}^{(2)} = W^{(2)} x^{(2)} + b^{(2)} && \rightarrow x^{(3)} = \rho(\tilde{x}^{(2)}) \\
&\rightarrow \cdots \\
&\rightarrow y = W^{(k)} x^{(k)} + b^{(k)}
\end{aligned}
$$



Berlin Mathematics Research Center

# Function Evaluation in ML

Typical function evaluation (deep neural net):

Propagation of one data point:

$$
\begin{aligned}
x = x^{(1)} &\rightarrow \tilde{x}^{(1)} = W^{(1)} x^{(1)} + b^{(1)} &&\rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \\
&\rightarrow \tilde{x}^{(2)} = W^{(2)} x^{(2)} + b^{(2)} &&\rightarrow x^{(3)} = \rho(\tilde{x}^{(2)}) \\
&\rightarrow \cdots \\
&\rightarrow y = W^{(k)} x^{(k)} + b^{(k)}
\end{aligned}
$$

Empirical risk, loss function, ...

$$
f(x_{1 \le i \le M}) = \frac{1}{M} \sum_{i=1}^{M} l(y_i(x_i), y_i^{NN})
$$

# Function Evaluation in ML

Typical function evaluation (deep neural net):

Propagation of one data point:

$$\begin{aligned} x = x^{(1)} &\rightarrow \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} &&\rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \\ &\rightarrow \tilde{x}^{(2)} = W^{(2)}x^{(2)} + b^{(2)} &&\rightarrow x^{(3)} = \rho(\tilde{x}^{(2)}) \\ &\rightarrow \cdots \\ &\rightarrow y = W^{(k)}x^{(k)} + b^{(k)} \end{aligned}$$

Empirical risk, loss function, ...

$$f(x_{1\leq i\leq M}) = \frac{1}{M}\sum_{i=1}^{M} l(y_i(x_i), y_i^{NN})$$

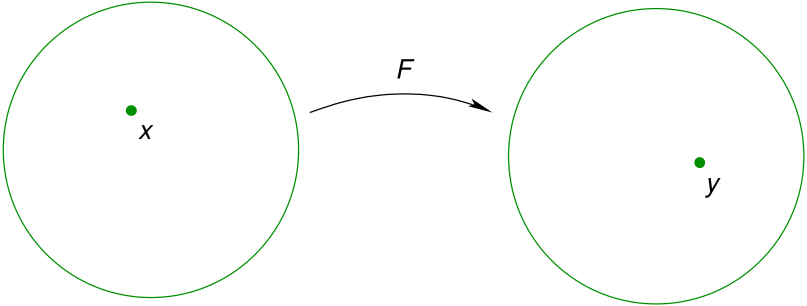
Stochastic gradient descent requires

$$\nabla_{W^1,b^1,\ldots,W^k,b^k} l(y_i(x_i), y_i^{NN})$$

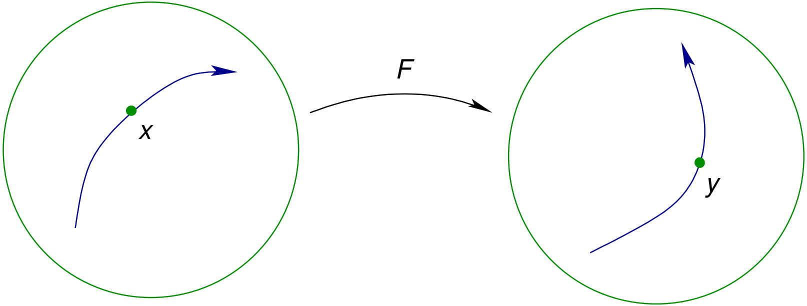


Berlin Mathematics Research Center

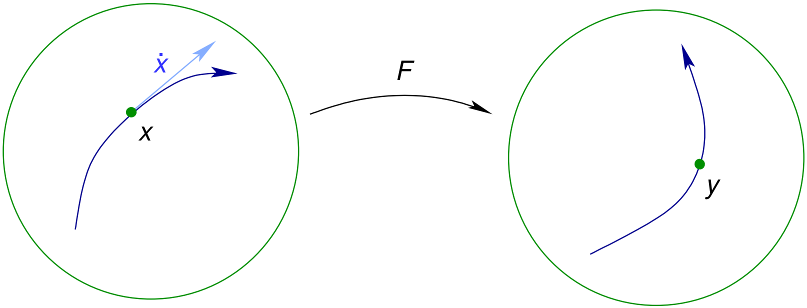
# Forward mode AD = Tangents/Sensitivities



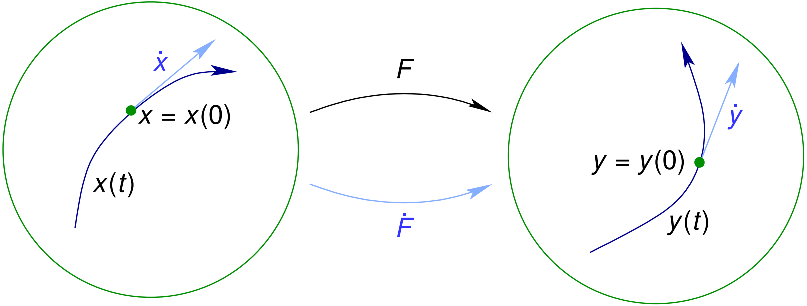
# Forward mode AD = Tangents/Sensitivities



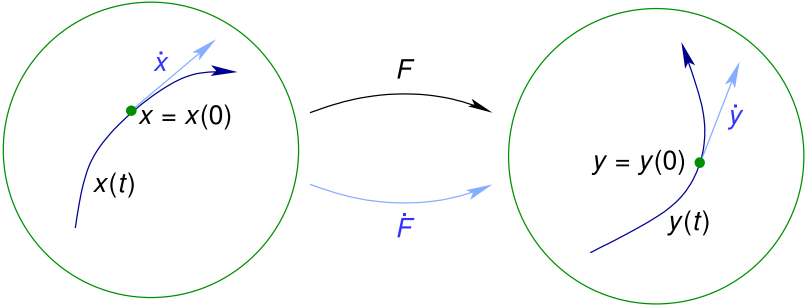
# Forward mode AD = Tangents/Sensitivities



# Forward mode AD = Tangents/Sensitivities



# Forward mode AD = Tangents/Sensitivities




$$\dot{y}(t) \;=\; \frac{\partial}{\partial t} F(x(t)) \;=\; F'(x(t))\,\dot{x}(t) \equiv \dot{F}(x, \dot{x})$$

# Forward Mode (Lighthouse)

$$
\begin{array}{lcl}
v_{-3} & = & x_1 = \nu \\
v_{-2} & = & x_2 = \gamma \\
v_{-1} & = & x_3 = \omega \\
v_0 & = & x_4 = t \\
\hline
v_1 & = & v_{-1} * v_0 \\
v_2 & = & \tan(v_1) \\
v_3 & = & v_{-2} - v_2 \\
v_4 & = & v_{-3} * v_2 \\
v_5 & = & v_4 / v_3 \\
v_6 & = & v_5 * v_{-2} \\
\hline
y_1 & = & v_5 \\
y_2 & = & v_6
\end{array}
$$

# Forward Mode (Lighthouse)

$$
\begin{array}{lll|lll}
v_{-3} & = & x_1 = \nu & \dot{v}_{-3} & = & \dot{x}_1 \\
v_{-2} & = & x_2 = \gamma & \dot{v}_{-2} & = & \dot{x}_2 \\
v_{-1} & = & x_3 = \omega & \dot{v}_{-1} & = & \dot{x}_3 \\
v_0 & = & x_4 = t & \dot{v}_0 & = & \dot{x}_4 \\
\hline
v_1 & = & v_{-1} * v_0 & & & \\
v_2 & = & \tan(v_1) & & & \\
v_3 & = & v_{-2} - v_2 & & & \\
v_4 & = & v_{-3} * v_2 & & & \\
v_5 & = & v_4 / v_3 & & & \\
v_6 & = & v_5 * v_{-2} & & & \\
\hline
y_1 & = & v_5 & & & \\
y_2 & = & v_6 & & &
\end{array}
$$

# Forward Mode (Lighthouse)

| | | | | | |
|---|---|---|---|---|---|
| $v_{-3}$ | $=$ | $x_1 = \nu$ | $\dot{v}_{-3}$ | $=$ | $\dot{x}_1$ |
| $v_{-2}$ | $=$ | $x_2 = \gamma$ | $\dot{v}_{-2}$ | $=$ | $\dot{x}_2$ |
| $v_{-1}$ | $=$ | $x_3 = \omega$ | $\dot{v}_{-1}$ | $=$ | $\dot{x}_3$ |
| $v_0$ | $=$ | $x_4 = t$ | $\dot{v}_0$ | $=$ | $\dot{x}_4$ |
| $v_1$ | $=$ | $v_{-1} * v_0$ | $\dot{v}_1$ | $=$ | $\dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0$ |
| $v_2$ | $=$ | $\tan(v_1)$ | | | |
| $v_3$ | $=$ | $v_{-2} - v_2$ | | | |
| $v_4$ | $=$ | $v_{-3} * v_2$ | | | |
| $v_5$ | $=$ | $v_4 / v_3$ | | | |
| $v_6$ | $=$ | $v_5 * v_{-2}$ | | | |
| $y_1$ | $=$ | $v_5$ | | | |
| $y_2$ | $=$ | $v_6$ | | | |



Berlin Mathematics Research Center

# Forward Mode (Lighthouse)

$$
\begin{array}{lcl|lcl}
v_{-3} & = & x_1 = \nu & \dot{v}_{-3} & = & \dot{x}_1 \\
v_{-2} & = & x_2 = \gamma & \dot{v}_{-2} & = & \dot{x}_2 \\
v_{-1} & = & x_3 = \omega & \dot{v}_{-1} & = & \dot{x}_3 \\
v_0 & = & x_4 = t & \dot{v}_0 & = & \dot{x}_4 \\
\hline
v_1 & = & v_{-1} * v_0 & \dot{v}_1 & = & \dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0 \\
v_2 & = & \tan(v_1) & \dot{v}_2 & = & \dot{v}_1 / \cos(v_1)^2 \\
v_3 & = & v_{-2} - v_2 & & & \\
v_4 & = & v_{-3} * v_2 & & & \\
v_5 & = & v_4 / v_3 & & & \\
v_6 & = & v_5 * v_{-2} & & & \\
\hline
y_1 & = & v_5 & & & \\
y_2 & = & v_6 & & &
\end{array}
$$

# Forward Mode (Lighthouse)

$$
\begin{array}{lcl|lcl}
v_{-3} & = & x_1 = \nu & \dot v_{-3} & = & \dot x_1 \\
v_{-2} & = & x_2 = \gamma & \dot v_{-2} & = & \dot x_2 \\
v_{-1} & = & x_3 = \omega & \dot v_{-1} & = & \dot x_3 \\
v_0 & = & x_4 = t & \dot v_0 & = & \dot x_4 \\
\hline
v_1 & = & v_{-1} * v_0 & \dot v_1 & = & \dot v_{-1} * v_0 + v_{-1} * \dot v_0 \\
v_2 & = & \tan(v_1) & \dot v_2 & = & \dot v_1 / \cos(v_1)^2 \\
v_3 & = & v_{-2} - v_2 & \dot v_3 & = & \dot v_{-2} - \dot v_2 \\
v_4 & = & v_{-3} * v_2 & & & \\
v_5 & = & v_4 / v_3 & & & \\
v_6 & = & v_5 * v_{-2} & & & \\
\hline
y_1 & = & v_5 & & & \\
y_2 & = & v_6 & & &
\end{array}
$$

# Forward Mode (Lighthouse)

$$
\begin{array}{lll|lll}
v_{-3} & = & x_1 = \nu & \dot{v}_{-3} & = & \dot{x}_1 \\
v_{-2} & = & x_2 = \gamma & \dot{v}_{-2} & = & \dot{x}_2 \\
v_{-1} & = & x_3 = \omega & \dot{v}_{-1} & = & \dot{x}_3 \\
v_0 & = & x_4 = t & \dot{v}_0 & = & \dot{x}_4 \\
\hline
v_1 & = & v_{-1} * v_0 & \dot{v}_1 & = & \dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0 \\
v_2 & = & \tan(v_1) & \dot{v}_2 & = & \dot{v}_1 / \cos(v_1)^2 \\
v_3 & = & v_{-2} - v_2 & \dot{v}_3 & = & \dot{v}_{-2} - \dot{v}_2 \\
v_4 & = & v_{-3} * v_2 & \dot{v}_4 & = & \dot{v}_{-3} * v_2 + v_{-3} * \dot{v}_2 \\
v_5 & = & v_4 / v_3 & & & \\
v_6 & = & v_5 * v_{-2} & & & \\
\hline
y_1 & = & v_5 & & & \\
y_2 & = & v_6 & & &
\end{array}
$$

# Forward Mode (Lighthouse)

| | | | | | |
|---|---|---|---|---|---|
| $v_{-3}$ | $=$ | $x_1 = \nu$ | $\dot{v}_{-3}$ | $=$ | $\dot{x}_1$ |
| $v_{-2}$ | $=$ | $x_2 = \gamma$ | $\dot{v}_{-2}$ | $=$ | $\dot{x}_2$ |
| $v_{-1}$ | $=$ | $x_3 = \omega$ | $\dot{v}_{-1}$ | $=$ | $\dot{x}_3$ |
| $v_0$ | $=$ | $x_4 = t$ | $\dot{v}_0$ | $=$ | $\dot{x}_4$ |
| $v_1$ | $=$ | $v_{-1} * v_0$ | $\dot{v}_1$ | $=$ | $\dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0$ |
| $v_2$ | $=$ | $\tan(v_1)$ | $\dot{v}_2$ | $=$ | $\dot{v}_1 / \cos(v_1)^2$ |
| $v_3$ | $=$ | $v_{-2} - v_2$ | $\dot{v}_3$ | $=$ | $\dot{v}_{-2} - \dot{v}_2$ |
| $v_4$ | $=$ | $v_{-3} * v_2$ | $\dot{v}_4$ | $=$ | $\dot{v}_{-3} * v_2 + v_{-3} * \dot{v}_2$ |
| $v_5$ | $=$ | $v_4 / v_3$ | $\dot{v}_5$ | $=$ | $(\dot{v}_4 - \dot{v}_3 * v_5) * (1 / v_3)$ |
| $v_6$ | $=$ | $v_5 * v_{-2}$ | | | |
| $y_1$ | $=$ | $v_5$ | | | |
| $y_2$ | $=$ | $v_6$ | | | |

Berlin Mathematics Research Center MATH+

# Forward Mode (Lighthouse)

$$
\begin{array}{lcl|lcl}
v_{-3} & = & x_1 = \nu & \dot{v}_{-3} & = & \dot{x}_1 \\
v_{-2} & = & x_2 = \gamma & \dot{v}_{-2} & = & \dot{x}_2 \\
v_{-1} & = & x_3 = \omega & \dot{v}_{-1} & = & \dot{x}_3 \\
v_0 & = & x_4 = t & \dot{v}_0 & = & \dot{x}_4 \\
\hline
v_1 & = & v_{-1} * v_0 & \dot{v}_1 & = & \dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0 \\
v_2 & = & \tan(v_1) & \dot{v}_2 & = & \dot{v}_1 / \cos(v_1)^2 \\
v_3 & = & v_{-2} - v_2 & \dot{v}_3 & = & \dot{v}_{-2} - \dot{v}_2 \\
v_4 & = & v_{-3} * v_2 & \dot{v}_4 & = & \dot{v}_{-3} * v_2 + v_{-3} * \dot{v}_2 \\
v_5 & = & v_4 / v_3 & \dot{v}_5 & = & (\dot{v}_4 - \dot{v}_3 * v_5) * (1 / v_3) \\
v_6 & = & v_5 * v_{-2} & \dot{v}_6 & = & \dot{v}_5 * v_{-2} + v_5 * \dot{v}_{-2} \\
\hline
y_1 & = & v_5 & & & \\
y_2 & = & v_6 & & &
\end{array}
$$

# Forward Mode (Lighthouse)

$$
\begin{array}{lcl|lcl}
v_{-3} & = & x_1 = \nu & \dot{v}_{-3} & = & \dot{x}_1 \\
v_{-2} & = & x_2 = \gamma & \dot{v}_{-2} & = & \dot{x}_2 \\
v_{-1} & = & x_3 = \omega & \dot{v}_{-1} & = & \dot{x}_3 \\
v_0 & = & x_4 = t & \dot{v}_0 & = & \dot{x}_4 \\
\hline
v_1 & = & v_{-1} * v_0 & \dot{v}_1 & = & \dot{v}_{-1} * v_0 + v_{-1} * \dot{v}_0 \\
v_2 & = & \tan(v_1) & \dot{v}_2 & = & \dot{v}_1 / \cos(v_1)^2 \\
v_3 & = & v_{-2} - v_2 & \dot{v}_3 & = & \dot{v}_{-2} - \dot{v}_2 \\
v_4 & = & v_{-3} * v_2 & \dot{v}_4 & = & \dot{v}_{-3} * v_2 + v_{-3} * \dot{v}_2 \\
v_5 & = & v_4 / v_3 & \dot{v}_5 & = & (\dot{v}_4 - \dot{v}_3 * v_5) * (1 / v_3) \\
v_6 & = & v_5 * v_{-2} & \dot{v}_6 & = & \dot{v}_5 * v_{-2} + v_5 * \dot{v}_{-2} \\
\hline
y_1 & = & v_5 & \dot{y}_1 & = & \dot{v}_5 \\
y_2 & = & v_6 & \dot{y}_2 & = & \dot{v}_6
\end{array}
$$

# Complexity (Forward Mode)

| tang | $c$ | $\pm$ | $*$ | $\psi$ |
|---|---|---|---|---|
| MOVES | 1 + 1 | 3 + 3 | 3 + 3 | 2 + 2 |
| ADDS | 0 | 1 + 1 | 0 + 1 | 0 + 0 |
| MULTS | 0 | 0 | 1 + 2 | 0 + 1 |
| NLOPS | 0 | 0 | 0 | 1 + 1 |

# Complexity (Forward Mode)

| tang | $c$ | $\pm$ | $*$ | $\psi$ |
|---|---|---|---|---|
| MOVES | $1+1$ | $3+3$ | $3+3$ | $2+2$ |
| ADDS | 0 | $1+1$ | $0+1$ | $0+0$ |
| MULTS | 0 | 0 | $1+2$ | $0+1$ |
| NLOPS | 0 | 0 | 0 | $1+1$ |

$\Longrightarrow$ $$\text{OPS}(F'(x)\dot{x}) \quad \leq \quad c\,\text{OPS}(F(x))$$

with $c \in [2, 5/2]$ platform dependent

# Forward Mode AD for ML

Typical function evaluation (deep neural net):

$$x = x^{(1)} \rightarrow \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} \quad \rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \quad \ldots \quad \rightarrow y = W^{(k)}x^{(k)} + b^{(k)}$$

Attention: Optimization variables $W$ and $b$ $\Rightarrow$ AD computes $\dot{W}$ and $\dot{b}$!

# Forward Mode AD for ML

Typical function evaluation (deep neural net):

$$x = x^{(1)} \rightarrow \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} \quad \rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \quad \ldots \quad \rightarrow y = W^{(k)}x^{(k)} + b^{(k)}$$

Attention: Optimization variables $W$ and $b$ $\Rightarrow$ AD computes $\dot{W}$ and $\dot{b}$!

$$\begin{aligned} x = x^{(1)} \rightarrow \tilde{x}^{(1)} &= W^{(1)}x^{(1)} + b^{(1)} \quad &\rightarrow x^{(2)} &= \rho(\tilde{x}^{(1)}) \\ \dot{\tilde{x}}^{(1)} &= \dot{W}^{(1)}x^{(1)} + \dot{b}^{(1)} \quad &\rightarrow \dot{x}^{(2)} &= \rho'(\tilde{x}^{(1)})\dot{\tilde{x}}^{(1)} \end{aligned}$$

# Forward Mode AD for ML

Typical function evaluation (deep neural net):

$$x = x^{(1)} \to \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} \quad \to x^{(2)} = \rho(\tilde{x}^{(1)}) \quad \dots \quad \to y = W^{(k)}x^{(k)} + b^{(k)}$$

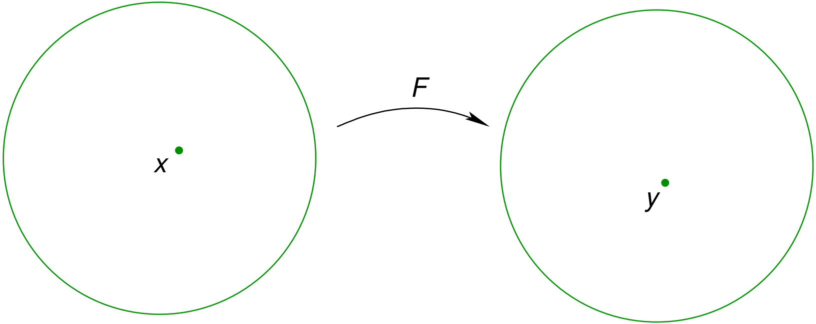
Attention: Optimization variables $W$ and $b$ $\Rightarrow$ AD computes $\dot{W}$ and $\dot{b}$!

$$\begin{aligned}
x = x^{(1)} \to & \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} & \to x^{(2)} = \rho(\tilde{x}^{(1)}) \\
& \dot{\tilde{x}}^{(1)} = \dot{W}^{(1)}x^{(1)} + \dot{b}^{(1)} & \to \dot{x}^{(2)} = \rho'(\tilde{x}^{(1)})\dot{\tilde{x}}^{(1)} \\
\to & \tilde{x}^{(2)} = W^{(2)}x^{(2)} + b^{(2)} & \to x^{(3)} = \rho(\tilde{x}^{(2)}) \\
& \dot{\tilde{x}}^{(2)} = \dot{W}^{(2)}x^{(2)} + W^{(2)}\dot{x}^{(2)} + \dot{b}^{(2)} & \to \dot{x}^{(3)} = \rho'(\tilde{x}^{(3)})\dot{\tilde{x}}^{(3)} \\
\to & \dots \\
\to & y = W^{(k)}x^{(k)} + b^{(k)} \\
\to & \dot{y} = \dot{W}^{(k)}x^{(k)} + W^{(k)}\dot{x}^{(k)} + \dot{b}^{(k)}
\end{aligned}$$



Berlin Mathematics Research Center

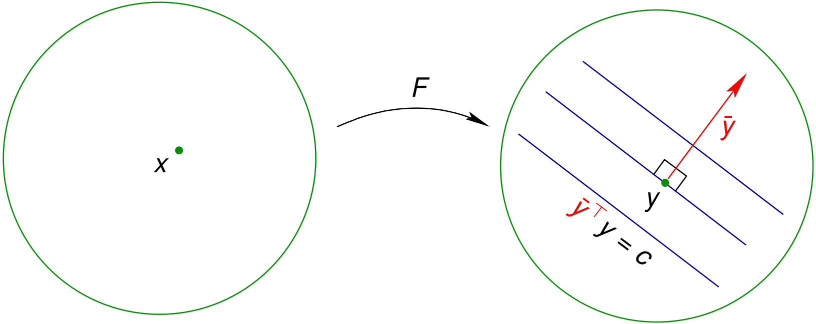
# Reverse Mode AD = Discrete Adjoints



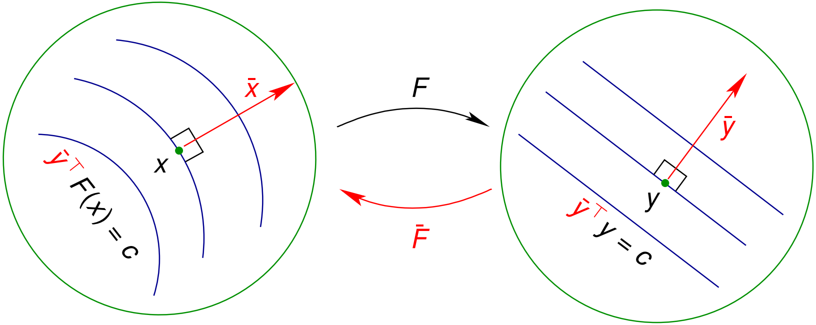



Berlin Mathematics Research Center

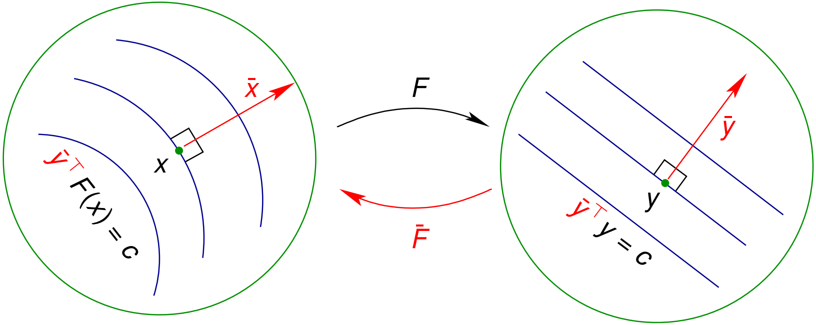
# Reverse Mode AD = Discrete Adjoints



# Reverse Mode AD = Discrete Adjoints



# Reverse Mode AD = Discrete Adjoints




$$\bar{x} \equiv \bar{y}^\top F'(x) = \nabla_x \langle \bar{y}^\top F(x) \rangle \equiv \bar{F}(x, \bar{y})$$

# Reverse Mode (Lighthouse)

$$v_{-3} = x_1; \quad v_{-2} = x_2; \quad v_{-1} = x_3; \quad v_0 = x_4;$$
$$v_1 = v_{-1} * v_0;$$
$$v_2 = \tan(v_1);$$
$$v_3 = v_{-2} - v_2;$$
$$v_4 = v_{-3} * v_2;$$
$$v_5 = v_4 / v_3;$$
$$v_6 = v_5 * v_{-2};$$
$$y_1 = v_5; \quad y_2 = v_6;$$

---

$$\bar{v}_5 = \bar{y}_1; \quad \bar{v}_6 = \bar{y}_2;$$
$$\bar{v}_5 \mathrel{+}= \bar{v}_6 * v_{-2}; \quad \bar{v}_{-2} \mathrel{+}= \bar{v}_6 * v_5;$$
$$\bar{v}_4 \mathrel{+}= \bar{v}_5 / v_3; \quad \bar{v}_3 \mathrel{-}= \bar{v}_5 * v_5 / v_3;$$
$$\bar{v}_{-3} \mathrel{+}= \bar{v}_4 * v_2; \quad \bar{v}_2 \mathrel{+}= \bar{v}_4 * v_{-3};$$
$$\bar{v}_{-2} \mathrel{+}= \bar{v}_3; \; \bar{v}_2 \mathrel{-}= \bar{v}_3;$$
$$\bar{v}_1 \mathrel{+}= \bar{v}_2 / \cos^2(v_1);$$
$$\bar{v}_{-1} \mathrel{+}= \bar{v}_1 * v_0; \; \bar{v}_0 \mathrel{+}= \bar{v}_1 * v_{-1};$$
$$\bar{x}_4 = \bar{v}_0; \quad \bar{x}_3 = \bar{v}_{-1}; \quad \bar{x}_2 = \bar{v}_{-2}; \quad \bar{x}_1 = \bar{v}_{-3};$$

Berlin Mathematics Research Center

# Complexity (Reverse Mode)

| grad | $c$ | $\pm$ | $*$ | $\psi$ |
|---|---|---|---|---|
| MOVES | 1 + 1 | 3 + 6 | 3 + 8 | 2 + 5 |
| ADDS | 0 | 1 + 2 | 0 + 2 | 0 + 1 |
| MULTS | 0 | 0 | 1 + 2 | 0 + 1 |
| NLOPS | 0 | 0 | 0 | 1 + 1 |



Berlin Mathematics Research Center

# Complexity (Reverse Mode)

| grad | $c$ | $\pm$ | $*$ | $\psi$ |
|---|---|---|---|---|
| MOVES | $1+1$ | $3+6$ | $3+8$ | $2+5$ |
| ADDS | $0$ | $1+2$ | $0+2$ | $0+1$ |
| MULTS | $0$ | $0$ | $1+2$ | $0+1$ |
| NLOPS | $0$ | $0$ | $0$ | $1+1$ |

$$\text{OPS}(\bar{y}^\top F'(x)) \leq c\,\text{OPS}(F(x)), \qquad \text{MEM}(\bar{y}^\top F'(x)) \sim \text{OPS}(F(x))$$

with $c \in [3,4]$ platform dependent

# Complexity (Reverse Mode)

| grad | $c$ | $\pm$ | $*$ | $\psi$ |
|---|---|---|---|---|
| MOVES | $1+1$ | $3+6$ | $3+8$ | $2+5$ |
| ADDS | 0 | $1+2$ | $0+2$ | $0+1$ |
| MULTS | 0 | 0 | $1+2$ | $0+1$ |
| NLOPS | 0 | 0 | 0 | $1+1$ |

$$\text{OPS}(\bar{y}^\top F'(x)) \leq c\,\text{OPS}(F(x)), \qquad \text{MEM}(\bar{y}^\top F'(x)) \sim \text{OPS}(F(x))$$

with $c \in [3,4]$ platform dependent

**Remarks:**

- Cost for gradient calculation independent of $n$
- Memory requirement may cause problem! $\Rightarrow$ Checkpointing

# Reverse Mode AD for ML

Typical function evaluation (deep neural net):

$$
\begin{aligned}
x = x^{(1)} &\rightarrow \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} &&\rightarrow x^{(2)} = \rho(\tilde{x}^{(1)})\\
&\rightarrow \tilde{x}^{(2)} = W^{(2)}x^{(2)} + b^{(2)} &&\rightarrow x^{(3)} = \rho(\tilde{x}^{(2)})\\
&\rightarrow \cdots\\
&\rightarrow y = W^{(k)}x^{(k)} + b^{(k)}
\end{aligned}
$$

# Reverse Mode AD for ML

Typical function evaluation (deep neural net):

$$
\begin{aligned}
x = x^{(1)} &\rightarrow \tilde{x}^{(1)} = W^{(1)} x^{(1)} + b^{(1)} \quad \rightarrow x^{(2)} = \rho(\tilde{x}^{(1)}) \\
&\rightarrow \tilde{x}^{(2)} = W^{(2)} x^{(2)} + b^{(2)} \quad \rightarrow x^{(3)} = \rho(\tilde{x}^{(2)}) \\
&\rightarrow \cdots \\
&\rightarrow y = W^{(k)} x^{(k)} + b^{(k)}
\end{aligned}
$$

With $\bar{y} = 1$ one obtains

$$
\bar{W}^{(k)} = [x^{(k)}], \qquad \bar{x}^{(k)} = W^{(k)}, \qquad \bar{b}^{(k)} = \mathbb{1} \qquad , \ldots
$$



Berlin Mathematics Research Center

# Reverse Mode AD for ML

Typical function evaluation (deep neural net):

$$
\begin{aligned}
x = x^{(1)} &\to \tilde{x}^{(1)} = W^{(1)} x^{(1)} + b^{(1)} && \to x^{(2)} = \rho(\tilde{x}^{(1)}) \\
&\to \tilde{x}^{(2)} = W^{(2)} x^{(2)} + b^{(2)} && \to x^{(3)} = \rho(\tilde{x}^{(2)}) \\
&\to \cdots \\
&\to y = W^{(k)} x^{(k)} + b^{(k)}
\end{aligned}
$$

With $\bar{y} = 1$ one obtains

$$
\begin{aligned}
& & \bar{W}^{(k)} &= [x^{(k)}], & \bar{x}^{(k)} &= W^{(k)}, & \bar{b}^{(k)} &= \mathbb{1} \qquad , \ldots \\
\bar{\tilde{x}}^{(2)} &= \rho'(x^{(2)}) * \bar{x}^{(3)}, & \bar{W}^{(2)} &= x^{(2)} * \bar{\tilde{x}}^{(2)}, & \bar{x}^{(2)} &= W^{(2)} * \bar{\tilde{x}}^{(2)}, & \bar{b}^{(2)} &= \bar{\tilde{x}}^{(2)} \\
\bar{\tilde{x}}^{(1)} &= \rho'(x^{(1)}) * \bar{x}^{(2)}, & \bar{W}^{(1)} &= x^{(1)} * \bar{\tilde{x}}^{(1)}, & \bar{x}^{(1)} &= W^{(1)} * \bar{\tilde{x}}^{(1)}, & \bar{b}^{(1)} &= \bar{\tilde{x}}^{(1)}
\end{aligned}
$$

# Reverse Mode AD for ML

Typical function evaluation (deep neural net):

$$
\begin{aligned}
x = x^{(1)} &\to \tilde{x}^{(1)} = W^{(1)}x^{(1)} + b^{(1)} \quad \to x^{(2)} = \rho(\tilde{x}^{(1)}) \\
&\to \tilde{x}^{(2)} = W^{(2)}x^{(2)} + b^{(2)} \quad \to x^{(3)} = \rho(\tilde{x}^{(2)}) \\
&\to \cdots \\
&\to y = W^{(k)}x^{(k)} + b^{(k)}
\end{aligned}
$$

With $\bar{y} = 1$ one obtains

$$
\begin{aligned}
& \bar{W}^{(k)} = [x^{(k)}], \quad \bar{x}^{(k)} = W^{(k)}, \quad \bar{b}^{(k)} = \mathbb{1} \quad , \ldots \\
\bar{\tilde{x}}^{(2)} = \rho'(x^{(2)}) * \bar{x}^{(3)}, \quad & \bar{W}^{(2)} = x^{(2)} * \bar{\tilde{x}}^{(2)}, \quad \bar{x}^{(2)} = W^{(2)} * \bar{\tilde{x}}^{(2)}, \quad \bar{b}^{(2)} = \bar{\tilde{x}}^{(2)} \\
\bar{\tilde{x}}^{(1)} = \rho'(x^{(1)}) * \bar{x}^{(2)}, \quad & \bar{W}^{(1)} = x^{(1)} * \bar{\tilde{x}}^{(1)}, \quad \bar{x}^{(1)} = W^{(1)} * \bar{\tilde{x}}^{(1)}, \quad \bar{b}^{(1)} = \bar{\tilde{x}}^{(1)}
\end{aligned}
$$

very simple to implement!

# Historical Development of AD

| | | | | | |
|---|---|---|---|---|---|
| J. Nolan | 1953 | $\rightarrow$ | J. M. Thames et al. | 1975 | $\rightarrow$ |
| L. M. Beda et al. | 1959 | $\rightarrow$ | D. D. Warner | 1975 | $\rightarrow$ |
| A. Gibbons | 1960 | $\rightarrow$ | | | |
| J. W. Hanson et al. | 1962 | $\rightarrow$ | J. Joss | 1980 | $\rightarrow$ |
| R. E. Wengert | 1964 | $\rightarrow$ | | | |
| R. D. Wilkins | 1964 | $\rightarrow$ | | | |
| G. Wanner | 1965 | $\rightarrow$ | L. B. Rall | 1980 | $\rightarrow$ |
| R. Bellman et al. | 1965 | $\rightarrow$ | | | |
| Y. F. Chang | 1967 | $\rightarrow$ | R. Kalaba et al. | 1983 | $\rightarrow$ |
| | | | | | |
| D. Barton et al. | 1971 | $\rightarrow$ | | | |
| | | | | | |
| R. E. Pugh | 1972 | $\rightarrow$ | | | |
| | | | L. C. W. Dixon et al. | 1986 | $\rightarrow$ |
| | | | ... | | |

# Historical Development of AD

| | | | | | |
|---|---|---|---|---|---|
| J. Nolan | 1953 | $\rightarrow$ | J. M. Thames et al. | 1975 | $\rightarrow$ |
| L. M. Beda et al. | 1959 | $\rightarrow$ | D. D. Warner | 1975 | $\rightarrow$ |
| A. Gibbons | 1960 | $\rightarrow$ | W. Miller | 1975 | $\leftarrow$ |
| J. W. Hanson et al. | 1962 | $\rightarrow$ | J. Joss | 1980 | $\rightarrow$ |
| R. E. Wengert | 1964 | $\rightarrow$ | G. Kedem | 1980 | $\leftarrow$ |
| R. D. Wilkins | 1964 | $\rightarrow$ | B. Speelpenning | 1980 | $\leftarrow$ |
| G. Wanner | 1965 | $\rightarrow$ | L. B. Rall | 1980 | $\rightarrow$ |
| R. Bellman et al. | 1965 | $\rightarrow$ | W. Baur, V. Strassen | 1983 | $\leftarrow$ |
| Y. F. Chang | 1967 | $\rightarrow$ | R. Kalaba et al. | 1983 | $\rightarrow$ |
| S. Linnainma | 1970 | $\leftarrow$ | M. Iri et al. | 1984 | $\leftarrow$ |
| D. Barton et al. | 1971 | $\rightarrow$ | K. W. Kim et al. | 1984 | $\leftarrow$ |
| G. M. Ostrowski | 1971 | $\leftarrow$ | J. W. Sawyer | 1984 | $\leftarrow$ |
| R. E. Pugh | 1972 | $\rightarrow$ | | | |
| W. Stacey | 1973 | $\leftarrow$ | L. C. W. Dixon et al. | 1986 | $\rightarrow$ |
| P. Werbos | 1974 | $\leftarrow$ | ... | | |

# Historical Development of AD

| | | | | | |
|---|---|---|---|---|---|
| J. Nolan | 1953 | $\rightarrow$ | J. M. Thames et al. | 1975 | $\rightarrow$ |
| L. M. Beda et al. | 1959 | $\rightarrow$ | D. D. Warner | 1975 | $\rightarrow$ |
| A. Gibbons | 1960 | $\rightarrow$ | W. Miller | 1975 | $\leftarrow$ |
| J. W. Hanson et al. | 1962 | $\rightarrow$ | J. Joss | 1980 | $\rightarrow$ |
| R. E. Wengert | 1964 | $\rightarrow$ | G. Kedem | 1980 | $\leftarrow$ |
| R. D. Wilkins | 1964 | $\rightarrow$ | B. Speelpenning | 1980 | $\leftarrow$ |
| G. Wanner | 1965 | $\rightarrow$ | L. B. Rall | 1980 | $\rightarrow$ |
| R. Bellman et al. | 1965 | $\rightarrow$ | W. Baur, V. Strassen | 1983 | $\leftarrow$ |
| Y. F. Chang | 1967 | $\rightarrow$ | R. Kalaba et al. | 1983 | $\rightarrow$ |
| S. Linnainma | 1970 | $\leftarrow$ | M. Iri et al. | 1984 | $\leftarrow$ |
| D. Barton et al. | 1971 | $\rightarrow$ | K. W. Kim et al. | 1984 | $\leftarrow$ |
| G. M. Ostrowski | 1971 | $\leftarrow$ | J. W. Sawyer | 1984 | $\leftarrow$ |
| R. E. Pugh | 1972 | $\rightarrow$ | E. M. Oblow et al. | 1985 | $\leftrightarrow$ |
| W. Stacey | 1973 | $\leftarrow$ | L. C. W. Dixon et al. | 1986 | $\rightarrow$ |
| P. Werbos | 1974 | $\leftarrow$ | . . . | | |

# Historical Development of AD

| | | | | | |
|---|---|---|---|---|---|
| J. Nolan | 1953 | → | J. M. Thames et al. | 1975 | → |
| L. M. Beda et al. | 1959 | → | D. D. Warner | 1975 | → |
| A. Gibbons | 1960 | → | W. Miller | 1975 | ← |
| J. W. Hanson et al. | 1962 | → | J. Joss | 1980 | → |
| R. E. Wengert | 1964 | → | G. Kedem | 1980 | ← |
| R. D. Wilkins | 1964 | → | B. Speelpenning | 1980 | ← |
| G. Wanner | 1965 | → | L. B. Rall | 1980 | → |
| R. Bellman et al. | 1965 | → | W. Baur, V. Strassen | 1983 | ← |
| Y. F. Chang | 1967 | → | R. Kalaba et al. | 1983 | → |
| S. Linnainma | 1970 | ← | M. Iri et al. | 1984 | ← |
| D. Barton et al. | 1971 | → | K. W. Kim et al. | 1984 | ← |
| G. M. Ostrowski | 1971 | ← | J. W. Sawyer | 1984 | ← |
| R. E. Pugh | 1972 | → | E. M. Oblow et al. | 1985 | ↔ |
| W. Stacey | 1973 | ← | L. C. W. Dixon et al. | 1986 | → |
| **P. Werbos** | **1974** | ← | . . . | | |

Rumelhart at al. (1986) made backpropagation famous for neural nets



Berlin Mathematics Research Center MATH+

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
- $$\text{Forward mode:} \quad \text{OPS}(F'(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [2, 5/2]$$



Berlin Mathematics Research Center MATH+

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
- Forward mode: $\quad \mathrm{OPS}(F'(x)\dot{x}) \quad \leq \quad c\,\mathrm{OPS}(F), \quad c \in [2, 5/2]$

  = discrete analogon to sensitivity equation



Berlin Mathematics Research Center

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
- $$\begin{array}{llcll} \text{Forward mode:} & \mathrm{OPS}(F'(x)\dot{x}) & \leq & c\,\mathrm{OPS}(F), & c \in [2, 5/2] \\ \text{Reverse mode:} & \mathrm{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\mathrm{OPS}(F), & c \in [3, 4] \\ & \mathrm{MEM}(\bar{y}^\top F'(x)) & \sim & \mathrm{OPS}(F), & \end{array}$$

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
-

$$\begin{array}{llll} \text{Forward mode:} & \mathrm{OPS}(F'(x)\dot{x}) & \leq & c\,\mathrm{OPS}(F), \quad c \in [2, 5/2] \\ \text{Reverse mode:} & \mathrm{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\mathrm{OPS}(F), \quad c \in [3, 4] \\ & \mathrm{MEM}(\bar{y}^\top F'(x)) & \sim & \mathrm{OPS}(F), \end{array}$$

= discrete analogon to adjoint equation



Berlin Mathematics Research Center MATH+

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
- $$\begin{array}{llcll} \text{Forward mode:} & \mathrm{OPS}(F'(x)\dot{x}) & \leq & c\,\mathrm{OPS}(F), & c \in [2, 5/2] \\ \text{Reverse mode:} & \mathrm{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\mathrm{OPS}(F), & c \in [3, 4] \\ & \mathrm{MEM}(\bar{y}^\top F'(x)) & \sim & \mathrm{OPS}(F), & \end{array}$$

$\Longrightarrow$ Gradients are cheap $\sim$ Function costs!!

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
-

$$\begin{array}{llcll} \text{Forward mode:} & \mathrm{OPS}(F'(x)\dot{x}) & \leq & c\,\mathrm{OPS}(F), & c \in [2, 5/2] \\ \text{Reverse mode:} & \mathrm{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\mathrm{OPS}(F), & c \in [3, 4] \\ & \mathrm{MEM}(\bar{y}^\top F'(x)) & \sim & \mathrm{OPS}(F), & \end{array}$$

$\Longrightarrow$ Gradients are cheap $\sim$ Function costs!!

- Combination: $\mathrm{OPS}(\bar{y}^\top F''(x)\dot{x}) \leq c\,\mathrm{OPS}(F), \quad c \in [7, 10]$
- Consistent derivative information!

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
-

$$\begin{array}{llcll} \text{Forward mode:} & \text{OPS}(F'(x)\dot{x}) & \leq & c\,\text{OPS}(F), & c \in [2, 5/2] \\ \text{Reverse mode:} & \text{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\text{OPS}(F), & c \in [3, 4] \\ & \text{MEM}(\bar{y}^\top F'(x)) & \sim & \text{OPS}(F), & \end{array}$$

$\Longrightarrow$ Gradients are cheap $\sim$ Function costs!!

- Combination: $\text{OPS}(\bar{y}^\top F''(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [7, 10]$
- Consistent derivative information!
- AD-tools: ADOL-C, CoDiPack, Tapenade, INTLAB, ADiMat, ...
- General purpose tools: FEniCS, SU2, PyTorch, TensorFlow, ...

# Overview AD Theory and Tools

- Differentiation of computer programmes with working accuracy (Griewank, Kulshreshtha, Walther 2012)
-

$$\begin{array}{lllll} \text{Forward mode:} & \text{OPS}(F'(x)\dot{x}) & \leq & c\,\text{OPS}(F), & c \in [2, 5/2] \\ \text{Reverse mode:} & \text{OPS}(\bar{y}^\top F'(x)) & \leq & c\,\text{OPS}(F), & c \in [3, 4] \\ & \text{MEM}(\bar{y}^\top F'(x)) & \sim & \text{OPS}(F), & \end{array}$$

$$\Longrightarrow \quad \text{Gradients are cheap} \sim \text{Function costs!!}$$

- Combination: $\text{OPS}(\bar{y}^\top F''(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [7, 10]$
- Consistent derivative information!
- AD-tools: ADOL-C, CoDiPack, Tapenade, INTLAB, ADiMat, ...
- General purpose tools: FEniCS, SU2, PyTorch, TensorFlow, ...

(Griewank, Walther 2008), (Naumann 2012), www.autodiff.org



Berlin Mathematics Research Center

# Automatic Differentiation by OverLoading in C++

- ADOL-C version 2.7, available at COIN-OR since 2009, open source (GPL or ECL)
- based on operator overloading, trace as internal representation

# Automatic Differentiation by OverLoading in C++

- ADOL-C version 2.7, available at COIN-OR since 2009, open source (GPL or ECL)
- based on operator overloading, trace as internal representation
- general-purpose AD tool with focus on functionalities
- interfaces to ColPack (Purdue University) and Ipopt (COIN-OR)

# Automatic Differentiation by OverLoading in C++

- ADOL-C version 2.7, available at COIN-OR since 2009, open source (GPL or ECL)
- based on operator overloading, trace as internal representation
- general-purpose AD tool with focus on functionalities
- interfaces to ColPack (Purdue University) and Ipopt (COIN-OR)
- current developments
  - Julia interface ADOLC.jl
  - exploitation of fixed-point structure for second-order derivatives
  - generalized derivatives for nonsmooth functions

# Piezoelectricity

Fundamental properties:

- Transformation of **mechanical energy** into **electrical energy**
- Transformation of **electrical energy** into **mechanical energy**

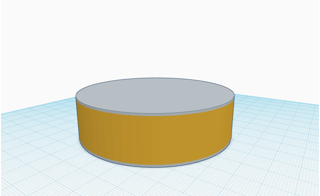
# Piezoelectricity

Fundamental properties:

- Transformation of **mechanical energy** into **electrical energy**
- Transformation of **electrical energy** into **mechanical energy**

Used in wide range of applications:
Pressure Sensors, Ultrasonic Cleaning, Ultrasound Imaging, Piezoelectric Speakers, Electronic Toothbrushes, Instrument Pickups, Microphones, Piezoelectric Igniters, Electricity Generation, Tennis Racquets, . . .

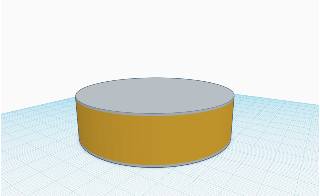
# The Considered Setting



https://www.piceramic.de/de/produkte/piezokeramische-bauelemente/scheiben-staebe-und-zylinder/piezoelektrische-scheiben-1206710/

- Piezoceramics come in many shapes and sizes
  here: disk shaped ceramics (very popular, cheap(er) simulation)

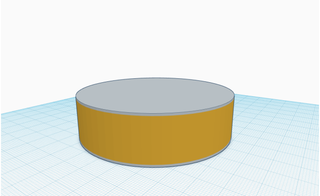
# The Considered Setting



https://www.piceramic.de/de/produkte/piezokeramische-bauelemente/scheiben-staebe-und-zylinder/piezoelektrische-scheiben-1206710/

- Piezoceramics come in many shapes and sizes
  here: disk shaped ceramics (very popular, cheap(er) simulation)
- Thanks to Benjamin Jurgelucks and Veronika Schulze!
  Cooperation with Measurement Engineering Group, Prof. Henning, Univ Paderborn

# The Considered Setting



https://www.piceramic.de/de/produkte/piezokeramische-bauelemente/scheiben-staebe-und-zylinder/piezoelektrische-scheiben-1206710/

- Piezoceramics come in many shapes and sizes
  here: disk shaped ceramics (very popular, cheap(er) simulation)
- Thanks to Benjamin Jurgelucks and Veronika Schulze!
  Cooperation with Measurement Engineering Group, Prof. Henning, Univ Paderborn
- Here: Consider only small loads $\Rightarrow$ Disregard thermal effects
  Nonlinear effects $\Rightarrow$ DFG research group NEPTUN

# Inverse Problem - State of the Art

- Sensitivity too small for some parameter
  (using conventional methods or data provided by manufacturer)
  Up to 20% error not uncommon

# Inverse Problem - State of the Art

- Sensitivity too small for some parameter
  (using conventional methods or data provided by manufacturer)
  Up to 20% error not uncommon

- Alternative approaches:
  - Use multiple differently shaped piezoceramics
    $\Rightarrow$ Leads to inconsistent datasets
  - Use additional measuremens of surface displacement
    $\Rightarrow$ Very expensive and still low sensitivity
  - Low sensitivity parameters are excluded from parameter identification

# Inverse Problem - State of the Art

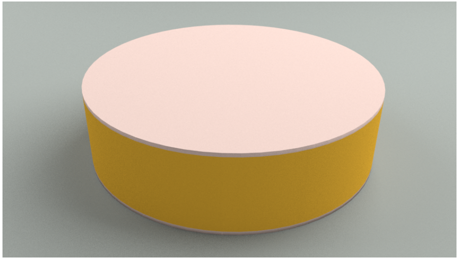
- Sensitivity too small for some parameter
  (using conventional methods or data provided by manufacturer)
  Up to 20% error not uncommon

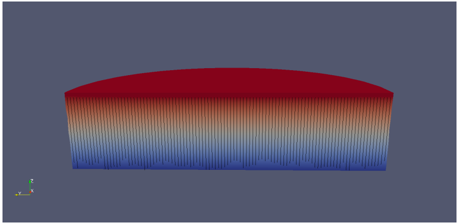
- Alternative approaches:
  - Use multiple differently shaped piezoceramics
    $\Rightarrow$ Leads to inconsistent datasets
  - Use additional measuremens of surface displacement
    $\Rightarrow$ Very expensive and still low sensitivity
  - Low sensitivity parameters are excluded from parameter identification

**Goal:** Identify all parameters using a single piezoceramic and impedance measurements

# AD-enabled Optimization of the Electrodes

Fully covering electrodes

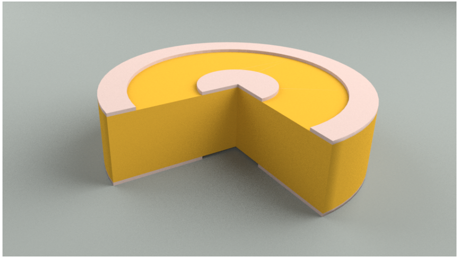


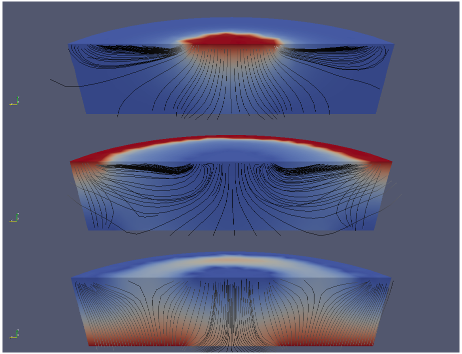


Thanks to B. Jurgelucks

# AD-enabled Optimization of the Electrodes

Triple-ring electrodes





Thanks to B. Jurgelucks



Berlin Mathematics Research Center
MATH+

# Real Measurements

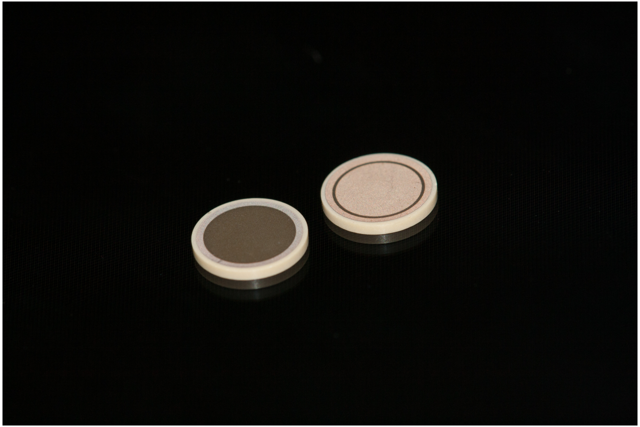
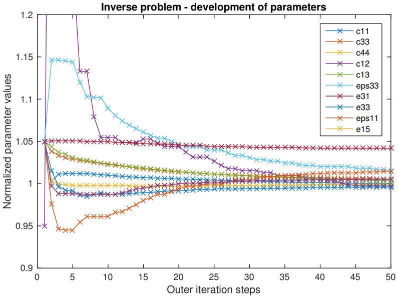
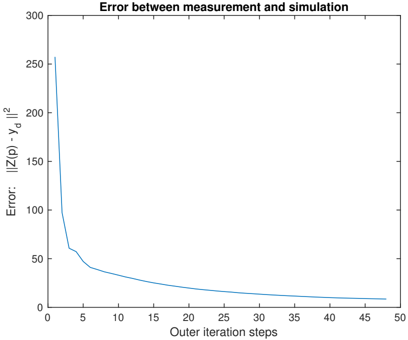


Photo by S. Olfert

# Appropriate Version of AD-enabled Gauss-Newton Method



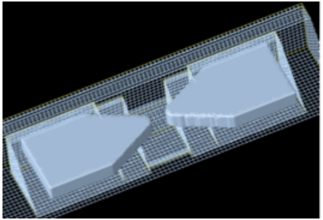

- None of the 10 (!) parameters diverges
- To the best of our knowledge this has not been possible with only one piezoceramic

# Optical Nano-Structures

**State of the art:** Nano-structures are used to confine light

Simple example: Bow-tie antenna

- metallic nano structure
- two triangles and gap
- Size: 100 nm ($< \lambda_{\text{light}}$!)
- intensity enhancement in gap

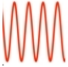


# Possible Configurations

Simple setting:

- different structure

- “simple” excitation



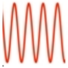
- pure metal
- extremely short dephasing

# Possible Configurations

Simple setting:

- different structure

- “simple” excitation



- pure metal
- extremely short dephasing

Advanced setting:

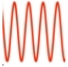
- fixed structure

# Possible Configurations

Simple setting:

- different structure

- “simple” excitation

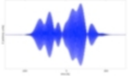


- pure metal
- extremely short dephasing

Advanced setting:

- fixed structure

- sophisticated excitation



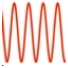
Berlin Mathematics Research Center MATH+

# Possible Configurations

Simple setting:

- different structure

- “simple” excitation

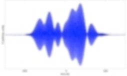


- pure metal
- extremely short dephasing

Advanced setting:

- fixed structure

- sophisticated excitation



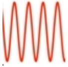
- add resonances in semiconductors

# Possible Configurations

Simple setting:

- different structure

- “simple” excitation

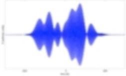


- pure metal
- extremely short dephasing

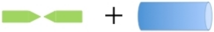
Advanced setting:

- fixed structure

- sophisticated excitation



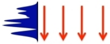
- add resonances in semiconductors
- longer dephasing



# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:

 $\longleftarrow$ adaptable light puls $E(t)$

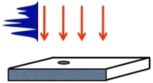


Berlin Mathematics Research Center MATH+

# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:



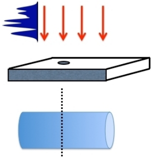
$\longleftarrow$ adaptable light puls $E(t)$

$\longleftarrow$ metal aperture with air hole

# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:



⟵ adaptable light puls $E(t)$

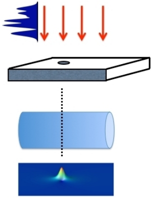
⟵ metal aperture with air hole

⟵ quantum wire

# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:



⟵ adaptable light puls $E(t)$

⟵ metal aperture with air hole

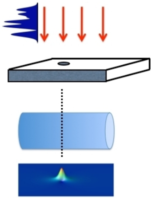
⟵ quantum wire

⟵ electron distribution?

# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:



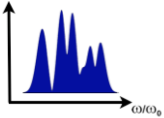
⟵ adaptable light puls $E(t)$

⟵ metal aperture with air hole

⟵ quantum wire

⟵ electron distribution?

Light puls:

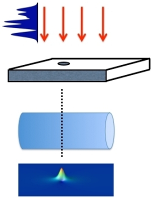



$$\text{with } E(t) = \sum A_i \exp\left(-\left(\frac{t-t_i}{\delta t_i}\right)^2\right)\cos(\omega_i t + \phi_i)$$

# Test Case: Quantum Wire

Cooperation with T. Meier, M. Reichelt, Dep. Physik, Uni Paderborn

Generic configuration:



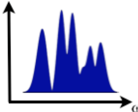
$\longleftarrow$ adaptable light puls $E(t)$

$\longleftarrow$ metal aperture with air hole

$\longleftarrow$ quantum wire

$\longleftarrow$ electron distribution?

Light puls:




with $E(t) = \sum A_i \exp\left(-\left(\frac{t-t_i}{\delta t_i}\right)^2\right) \cos(\omega_i t + \phi_i)$

Parameter: $A_i, \phi_i, \omega_i, t_i$ $\Rightarrow$ up to 120!

# Mathematical Formulation

**State equation:**

$$
\begin{aligned}
\frac{\partial}{\partial t} p &= \frac{i}{\hbar}(\epsilon_0 - \epsilon_1)p + \frac{i}{\hbar}\mathbf{E}(t)\cdot\mathbf{d}\,(n_0 - n_1) \\
\frac{\partial}{\partial t} n_0 &= \frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t)\cdot\mathbf{d}\,p^*] \\
\frac{\partial}{\partial t} n_1 &= -\frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t)\cdot\mathbf{d}\,p^*] \\
1 &= n_1 + n_0
\end{aligned}
$$

$\Rightarrow$ Three complex-valued coupled differential equations $p$, $n_0$ and $n_1$ distributed in space.

# Mathematical Formulation

**State equation:**

$$\begin{aligned}
\frac{\partial}{\partial t} p &= \frac{i}{\hbar}(\epsilon_0 - \epsilon_1)p + \frac{i}{\hbar}\mathbf{E}(t)\cdot\mathbf{d}\,(n_0 - n_1) \\
\frac{\partial}{\partial t} n_0 &= \frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t)\cdot\mathbf{d}\,p^*] \\
\frac{\partial}{\partial t} n_1 &= -\frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t)\cdot\mathbf{d}\,p^*] \\
1 &= n_1 + n_0
\end{aligned}$$

Target:
Maximize energy at given time and given place with constant energy



Berlin Mathematics Research Center

# Mathematical Formulation

**State equation:**

$$\begin{aligned}
\frac{\partial}{\partial t} p &= \frac{i}{\hbar}(\epsilon_0 - \epsilon_1)p + \frac{i}{\hbar}\mathbf{E}(t) \cdot \mathbf{d}\,(n_0 - n_1) \\
\frac{\partial}{\partial t} n_0 &= \frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t) \cdot \mathbf{d}\,p^*] \\
\frac{\partial}{\partial t} n_1 &= -\frac{2}{\hbar}\mathrm{Im}\,[\mathbf{E}(t) \cdot \mathbf{d}\,p^*] \\
1 &= n_1 + n_0
\end{aligned}$$

Target:
Maximize energy at given time and given place with constant energy

= Maximize emitted radiation

$$I_{rad} = \left|\omega^2 P(\omega)\right|^2 = \left|\omega^2\, 2\,\mathrm{Re}(d\,p)\right|^2$$

# Function Evaluation of $I_{rad}(f(g(x)))$

```
x[] ← (phase[], amplitude[], width[], point[])

for time=0 to Tfinal do
    if (time >= Tobs && time < Tobs+dt)
        eval_time_step1(x,int_tar)
    else
        eval_time_step2(x,int_tar)
    end if
end for

eval_target(int_tar,fitness)
```



Berlin Mathematics Research Center MATH+

# Function Evaluation of $I_{rad}(f(g(x)))$

```
x[] ← (phase[], amplitude[], width[], point[])

for time=0 to Tfinal do
    if (time >= Tobs && time < Tobs+dt)
        eval_time_step1(x,int_tar)
    else
        eval_time_step2(x,int_tar)
    end if
end for

eval_target(int_tar,fitness)
```

scenarios:
\# independents $\in \{20, 60, 120\}$ $\Rightarrow$ Reverse mode!

# Function Evaluation of $I_{rad}(f(g(x)))$

```
x[] ← (phase[], amplitude[], width[], point[])

for time=0 to Tfinal do
    if (time >= Tobs && time < Tobs+dt)
        eval_time_step1(x,int_tar)
    else
        eval_time_step2(x,int_tar)
    end if
end for

eval_target(int_tar,fitness)
```

scenarios:

# independents $\in \{20, 60, 120\}$ $\Rightarrow$ Reverse mode!

# time steps $\in \{16000, 32000, 160000\}$ $\Rightarrow$ Checkpointing!

# Quantum Wire: Optimization

**So far:** Genetic algorithms

**Now:** L-BFGS and efficient gradient computation

- ADOL-C coupled with hand-coded adjoints
- Checkpointing (160 000 time steps!!)

$\Rightarrow$ TIME(gradient)/TIME(target function) $<$ 7 despite of checkpointing!

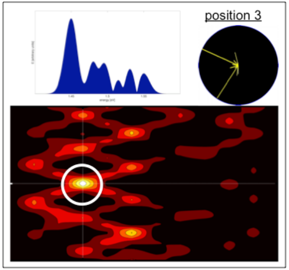
# Quantum Wire: Optimization

**So far:** Genetic algorithms

**Now:** L-BFGS and efficient gradient computation

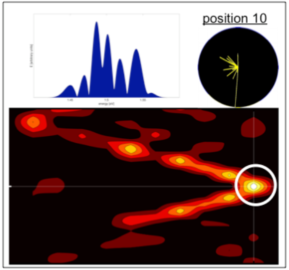
- ADOL-C coupled with hand-coded adjoints
- Checkpointing (160 000 time steps!!)

$\Rightarrow$ TIME(gradient)/TIME(target function) $<$ 7 despite of checkpointing!




excite
- at **same** position
- at **same** time
- with **same** energy

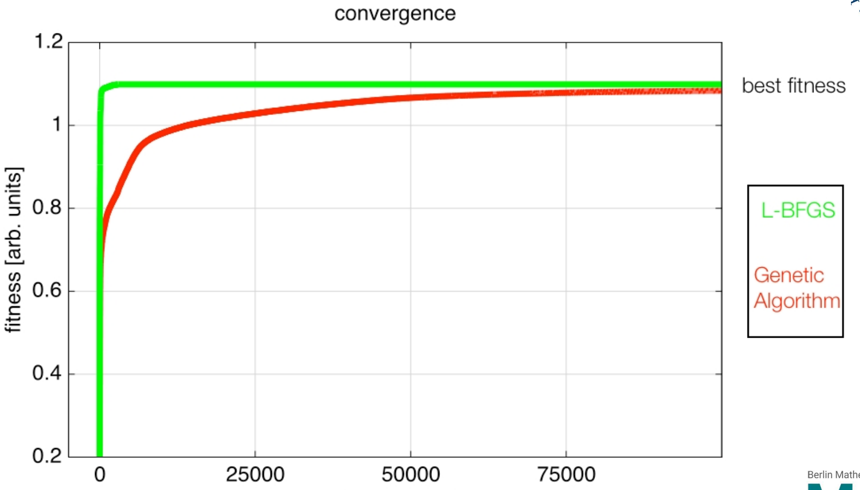
optimize
- for **same** $t_{opt}$
- **different** positions






Berlin Mathematics Research Center MATH+

# Quantum Wire: Comparison



# Photonic Nano-Resonators

Joint work with F. Binkowski, J. Kullig, F. Betz, L. Zschiedrich, J. Wiersig, S. Burger

Applications:

- probing single molecules with ultrahigh sensitivity
- designing nanoantennas with a tailored directivity
- large spontaneous emission rate or realizing efficient single-photon sources

# Photonic Nano-Resonators

Joint work with F. Binkowski, J. Kullig, F. Betz, L. Zschiedrich, J. Wiersig, S. Burger

Applications:

- probing single molecules with ultrahigh sensitivity
- designing nanoantennas with a tailored directivity
- large spontaneous emission rate or realizing efficient single-photon sources

Furter aspects:

- computed by solving the source-free Maxwell's equations

# Photonic Nano-Resonators

Joint work with F. Binkowski, J. Kullig, F. Betz, L. Zschiedrich, J. Wiersig, S. Burger

Applications:

- probing single molecules with ultrahigh sensitivity
- designing nanoantennas with a tailored directivity
- large spontaneous emission rate or realizing efficient single-photon sources

Furter aspects:

- computed by solving the source-free Maxwell's equations
- sensitivities of resonances are of interest for
  - better understanding of the underlying physical effects
  - an efficient optimization of corresponding photonic devices

# Behaviour of Sensitivities at EPs

Exceptional points: spectral degeneracies
(eigenfrequencies and eigenmodes coalesce)

# Behaviour of Sensitivities at EPs

Exceptional points: spectral degeneracies
(eigenfrequencies and eigenmodes coalesce)

- parametric fine-tuning is needed to achieve such non-Hermitian degeneracies
  $\Rightarrow$ exceptional points (EPs)

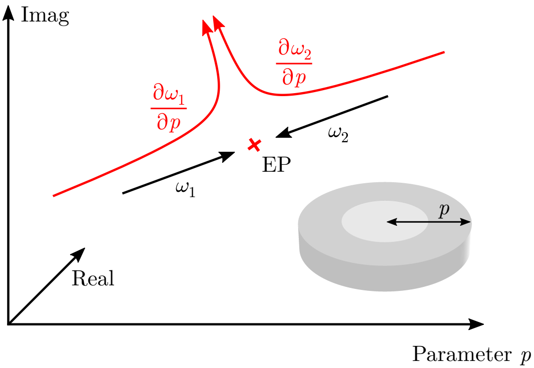
# Behaviour of Sensitivities at EPs

Exceptional points: spectral degeneracies
(eigenfrequencies and eigenmodes coalesce)

- parametric fine-tuning is needed to achieve such non-Hermitian degeneracies
  $\Rightarrow$ exceptional points (EPs)
- EPs have been connected to many interesting effects including
  - ultra-sensitive sensors
  - control of light transport
  - electromagnetically induced transparency
  - optical amplifiers,...

# Behaviour of Sensitivities at EPs

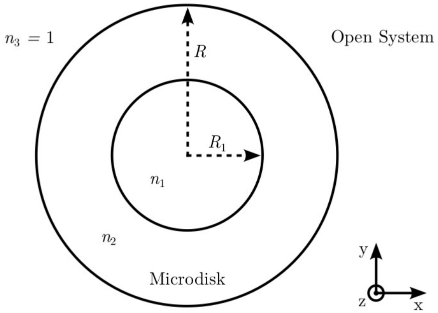
Exceptional points: spectral degeneracies
(eigenfrequencies and eigenmodes coalesce)

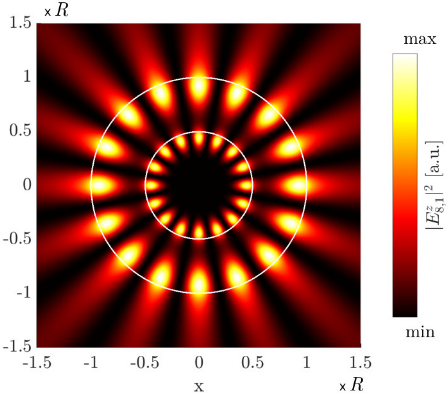


# Calculation of EPs and Their Sensitivities

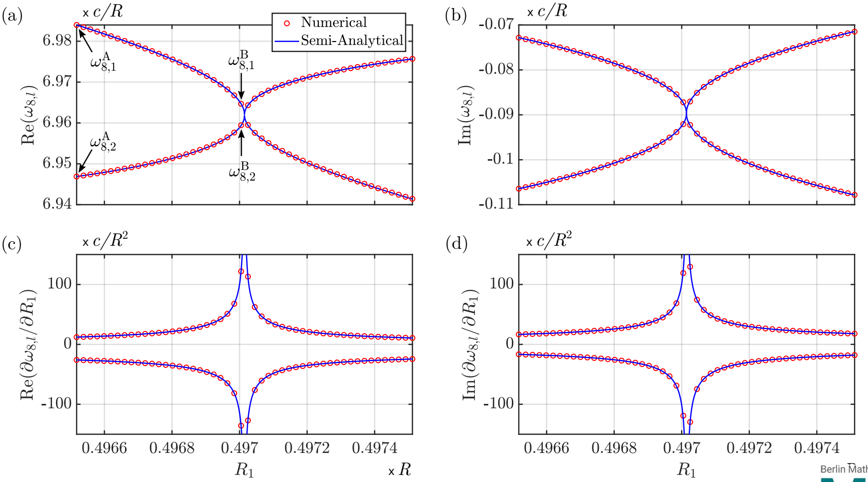
- RPExpand
  Software for Riesz projection expansion of resonance phenomena
  by F. Betz, F. Binkowski, S. Burger
- JCMSuite
  part of JCMWave (commercial software)
  based on FEM combined with efficient contour-integral method
  implements own AD approach

# The Considered Setting



# Eigenfrequencies and Their Sensitivities



# Conclusions

- Basics of Algorithmic Differentiation
  - Efficient evaluation of derivatives with working accuracy
  - Theory for basic modes complete, advanced AD?
  - Various tools/implementations available

# Conclusions

- Basics of Algorithmic Differentiation
  - Efficient evaluation of derivatives with working accuracy
  - Theory for basic modes complete, advanced AD?
  - Various tools/implementations available
- Structure exploitation indispensable

# Conclusions

- Basics of Algorithmic Differentiation
  - Efficient evaluation of derivatives with working accuracy
  - Theory for basic modes complete, advanced AD?
  - Various tools/implementations available
- Structure exploitation indispensable
- AD for applications in physics:
  - parameter identification for piezoceramics[1]
  - optimized excitation of nano antenna[2]
  - sensitivities of exceptional points of nano-resonators[3]

[1] e.g., L. Claes et al.: Inverse procedure for measuring piezoelectric material parameters using a single multi-electrode sample. J. Sensors and Sensor Systems 12 (1) (2023)

[2] e.g., M. Reichelt, A. Walther, T. Meier: Tailoring the high-harmonic emission in two-level systems and semiconductors by pulse shaping. JOSA B 29 (2) (2012)

[3] F. Binkowski, J. Kullig, F. Betz, L. Zschiedrich, A. Walther, J. Wiersig, S. Burger: Computing eigenfrequency sensitivities near exceptional points,Phys. Rev. Research 6 (2) (2024)