# Tomographic reconstruction from proton CT list-mode data using automatic differentiation

Aurélien Coussat  Jean Michel Létang  Simon Rit

CREATIS laboratory, Lyon, France

10/09/2026

# Outline

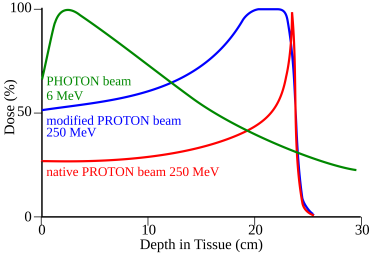

1. Introduction: sandwich time-of-flight proton computed tomography
2. First approach: calibration curve
3. Second approach: gradient descent reconstruction
4. Conclusion

# Outline

1. Introduction: sandwich time-of-flight proton computed tomography
2. First approach: calibration curve
3. Second approach: gradient descent reconstruction
4. Conclusion

# Introduction to proton computed tomography

- Proton therapy leverages the Bragg peak to target deep-seated tumors

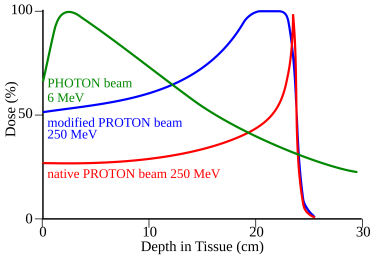



[1] Matthias Witt et al. In: *Zeitschrift für Medizinische Physik* 25.3 (2015)

[2] Ming Yang et al. In: *Physics in Medicine & Biology* 57.13 (2012)

# Introduction to proton computed tomography

- Proton therapy leverages the Bragg peak to target deep-seated tumors
- Treatment planning for proton therapy requires the RSP distribution within the patient's body

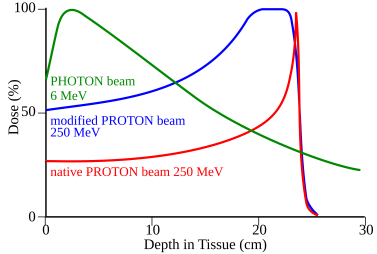


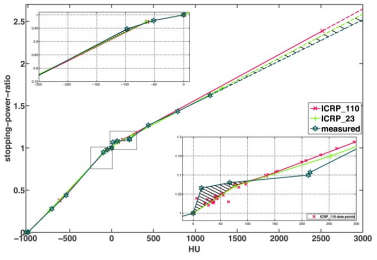

[1] Matthias Witt et al. In: *Zeitschrift für Medizinische Physik* 25.3 (2015)

[2] Ming Yang et al. In: *Physics in Medicine & Biology* 57.13 (2012)

# Introduction to proton computed tomography

- Proton therapy leverages the Bragg peak to target deep-seated tumors
- Treatment planning for proton therapy requires the RSP distribution within the patient's body
- This distribution is currently obtained by converting HUs obtained from an X-ray CT[1]



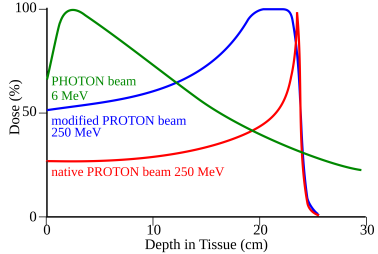



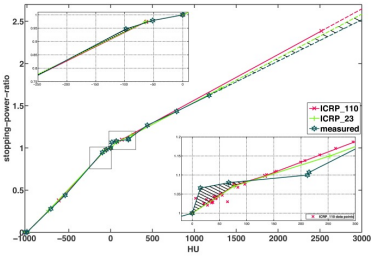

[1] Matthias Witt et al. In: *Zeitschrift für Medizinische Physik* 25.3 (2015)

[2] Ming Yang et al. In: *Physics in Medicine & Biology* 57.13 (2012)

# Introduction to proton computed tomography

- Proton therapy leverages the Bragg peak to target deep-seated tumors
- Treatment planning for proton therapy requires the RSP distribution within the patient's body
- This distribution is currently obtained by converting HUs obtained from an X-ray CT[1]
- The conversion introduces inaccuracies in the RSP estimation[2], resulting in increased safety margins



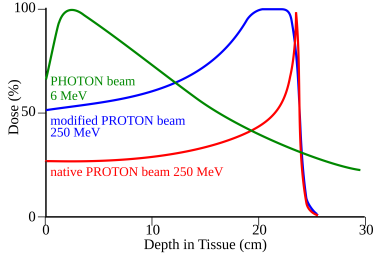



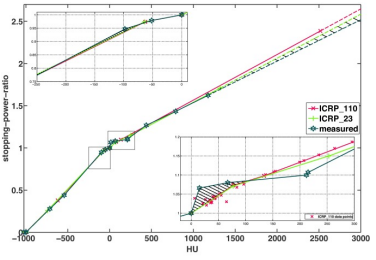

[1] Matthias Witt et al. In: *Zeitschrift für Medizinische Physik* 25.3 (2015)

[2] Ming Yang et al. In: *Physics in Medicine & Biology* 57.13 (2012)

# Introduction to proton computed tomography

- Proton therapy leverages the Bragg peak to target deep-seated tumors
- Treatment planning for proton therapy requires the RSP distribution within the patient's body
- This distribution is currently obtained by converting HUs obtained from an X-ray CT[1]
- The conversion introduces inaccuracies in the RSP estimation[2], resulting in increased safety margins
- PCT: use protons instead of X-rays!



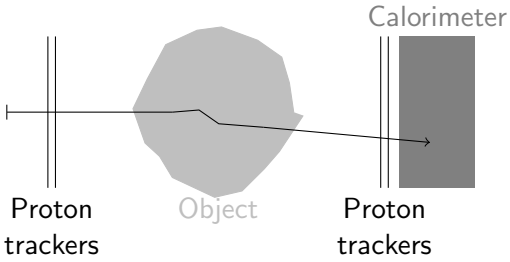




[1] Matthias Witt et al. In: *Zeitschrift für Medizinische Physik* 25.3 (2015)

[2] Ming Yang et al. In: *Physics in Medicine & Biology* 57.13 (2012)

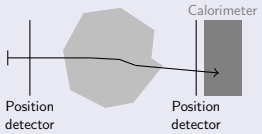
# Traditional pCT scanner

- In the object, protons are deflected
  - Multiple Coulomb scattering
  - Nuclear interactions
- Up- and downstream detectors track the protons' position and direction to estimate their most likely path (MLP)
- The calorimeter measures the residual energy to estimate the integral of the *stopping power* along the MLP
- Several projections are acquired



# Energy-loss, TOF and sandwich TOF pCT

## Energy-loss pCT

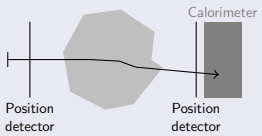



- Energy is detected using a calorimeter
- Challenging to have high acquisition rate

[3]

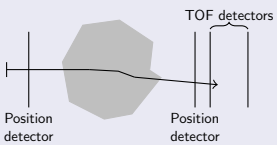
# Energy-loss, TOF and sandwich TOF pCT

## Energy-loss pCT




- Energy is detected using a calorimeter
- Challenging to have high acquisition rate

## TOF pCT[3]



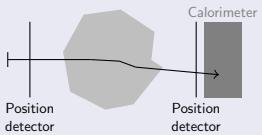

- Energy is estimated using TOF between two downstream detectors
- Impractical set-up

[3] William A. Worstell et al. In: *Medical Imaging 2019: Physics of Medical Imaging*. Vol. 10948. 2019, Felix Ulrich-Pur et al. In: *Physics in Medicine & Biology* 67.9 (2022), Nils Krah et al. In: *Physics in Medicine & Biology* 67.16 (2022)

[4] F. Ulrich-Pur et al. In: *Journal of Instrumentation* 18.02 (2023)

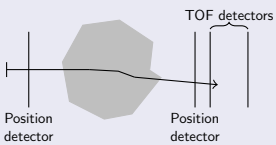
# Energy-loss, TOF and sandwich TOF pCT

## Energy-loss pCT



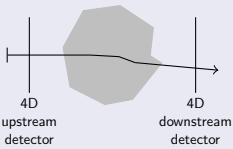

- Energy is detected using a calorimeter
- Challenging to have high acquisition rate

## TOF pCT[3]




- Energy is estimated using TOF between two downstream detectors
- Impractical set-up

## STOF pCT[4]



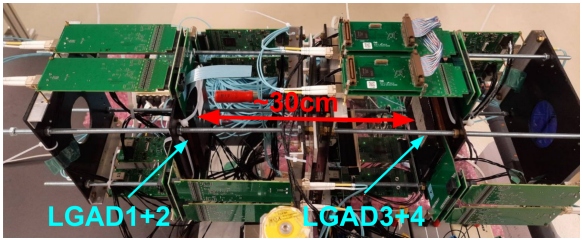

- WEPL is directly estimated from the TOF between the detectors
- Can be implemented with two 4D LGADs

[3] William A. Worstell et al. In: *Medical Imaging 2019: Physics of Medical Imaging*. Vol. 10948. 2019, Felix Ulrich-Pur et al. In: *Physics in Medicine & Biology* 67.9 (2022), Nils Krah et al. In: *Physics in Medicine & Biology* 67.16 (2022)

[4] F. Ulrich-Pur et al. In: *Journal of Instrumentation* 18.02 (2023)

# A few words about hardware

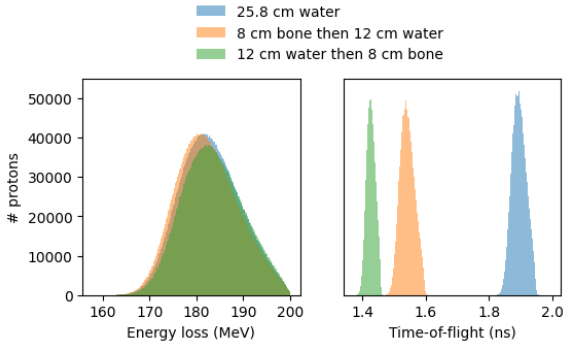
- Detectors implemented with LGADs
- Time resolution down to 30 ps to 50 ps possible
- Spatial resolution of less than 100 µm possible
- High particle rates (HADES: $10^8$ p/s/cm$^2$)
- Used in beam monitoring systems in high-energy physics and medical applications[5]




Courtesy of Felix Ulrich-Pur.

[5] Anna Vignati et al. In: *Medical Physics* 50.9 (2023)

# Difficulty of sandwich time-of-flight pCT

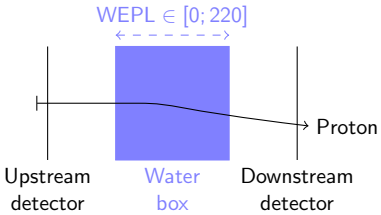



- Protons traversing different materials with equivalent WEPLs result in
  - equal energy loss
  - different TOFs
- The conversion between TOF and WEPL is non-bijective
- The forward problem is non-linear and non-trivial to invert

# Outline

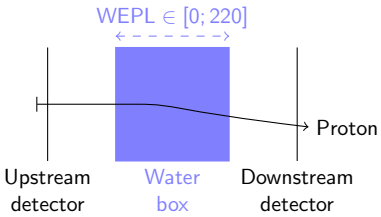
1. Introduction: sandwich time-of-flight proton computed tomography
2. First approach: calibration curve
3. Second approach: gradient descent reconstruction
4. Conclusion

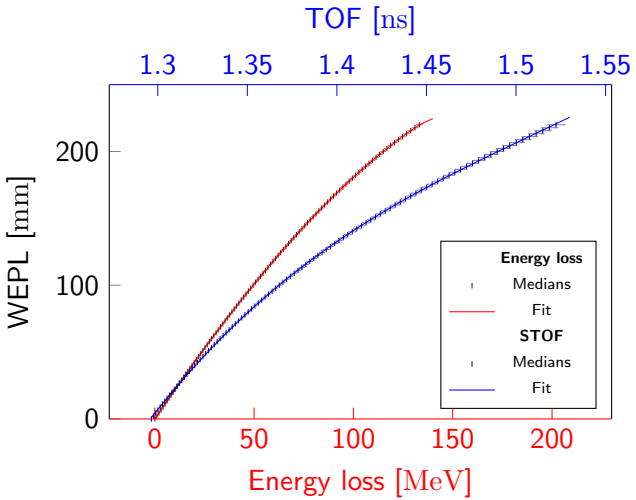
# First attempt using a 1D calibration curve




https://github.com/RTKConsortium/PCT/blob/main/applications/pctweplfit/pctweplfit.py Aurélien Coussat et al. In: *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*. May 2025

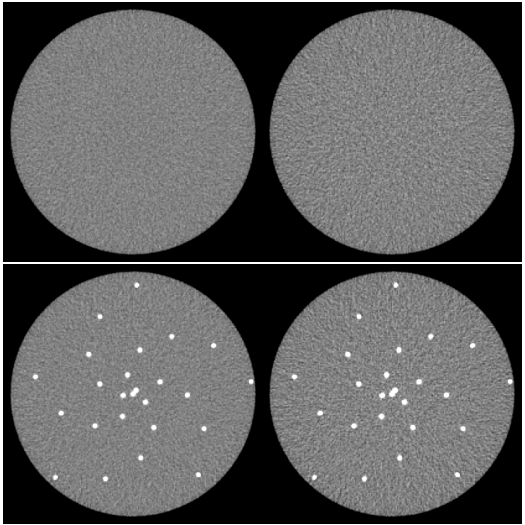
# First attempt using a 1D calibration curve

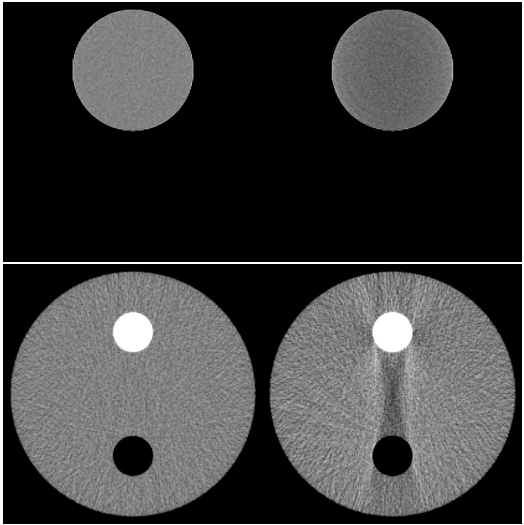


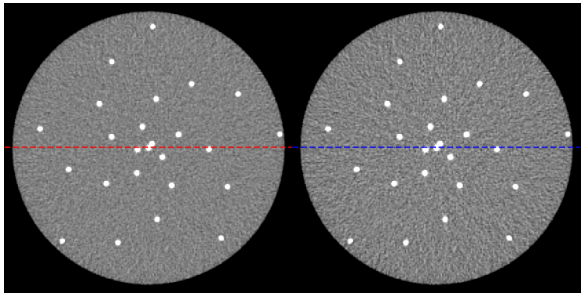

https://github.com/RTKConsortium/PCT/blob/main/applications/pctweplfit/pctweplfit.py Aurélien Coussat et al. In: *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*. May 2025

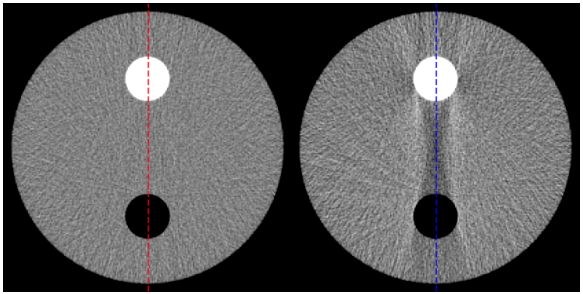
# Tomographic reconstructions (energy loss/sandwich time-of-flight)

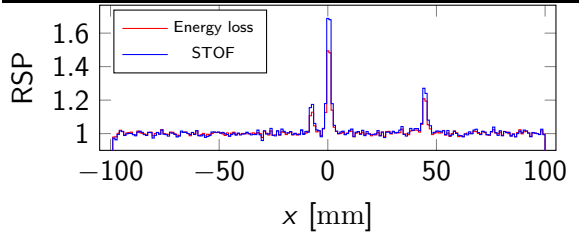


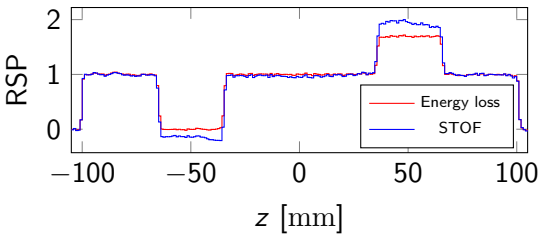


# Profiles






RSP
1.6
1.4
1.2
1
Energy loss
STOF
−100
−50
0
50
100
x [mm]


RSP
2
1
0
Energy loss
STOF
−100
−50
0
50
100
z [mm]

# Advantages and limitations

- Advantages:
  - Trivial to implement
  - Minimal computational cost
- Limitations:
  - Can only accurately reconstruct water
  - Assumes that the phantom is centered at the isocenter

# Advantages and limitations

- Advantages:
  - Trivial to implement
  - Minimal computational cost
- Limitations:
  - Can only accurately reconstruct water
  - Assumes that the phantom is centered at the isocenter

→Need for a more advanced reconstruction technique!

# Outline

1. Introduction: sandwich time-of-flight proton computed tomography
2. First approach: calibration curve
3. Second approach: gradient descent reconstruction
4. Conclusion

# Outline of the method

1. Start with an estimate of the RSP distribution

# Outline of the method

1. Start with an estimate of the RSP distribution
2. Calculate what would be the measured TOFs assuming this estimate *(forward model)*

# Outline of the method

1. Start with an estimate of the RSP distribution
2. Calculate what would be the measured TOFs assuming this estimate *(forward model)*
3. Measure a distance between calculated TOFs and measured TOFs *(cost function)*

# Outline of the method

1. Start with an estimate of the RSP distribution
2. Calculate what would be the measured TOFs assuming this estimate *(forward model)*
3. Measure a distance between calculated TOFs and measured TOFs *(cost function)*
4. Use this metric to update the estimate accordingly *(gradient descent)*

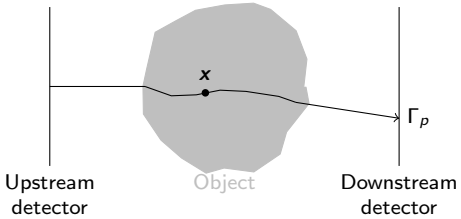
# Outline of the method

1. Start with an estimate of the RSP distribution
2. Calculate what would be the measured TOFs assuming this estimate *(forward model)*
3. Measure a distance between calculated TOFs and measured TOFs *(cost function)*
4. Use this metric to update the estimate accordingly *(gradient descent)*
5. Start over from 1

# Mathematical model of STOF pCT

- TOF of proton $p$:

$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})} \tag{1}$$



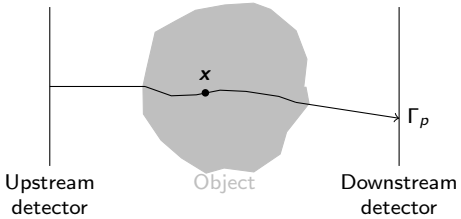
# Mathematical model of STOF pCT

- TOF of proton $p$:

$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})} \tag{1}$$

- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p(\boldsymbol{x}) = c\sqrt{1 - \frac{m_p c^2}{\left(m_p c^2 + E_p(\boldsymbol{x})\right)^2}} \tag{2}$$



# Mathematical model of STOF pCT

- TOF of proton $p$:

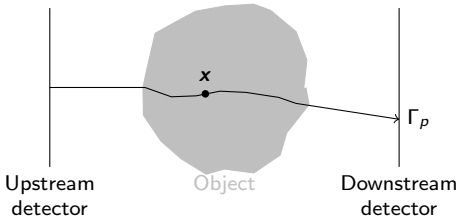
$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})} \tag{1}$$

- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p(\boldsymbol{x}) = c\sqrt{1 - \frac{m_p c^2}{\left(m_p c^2 + E_p(\boldsymbol{x})\right)^2}} \tag{2}$$

- Energy of the proton $p$ at point $\boldsymbol{x} = \Gamma_p(s)$:

$$E_p\left(\Gamma_p(s)\right) = E_{\text{in}} - \int_0^s \underbrace{\rho(\Gamma_p(s'))}_{\text{RSP}} \underbrace{S_{\text{water}}\left(E_p\left(\Gamma_p(s')\right)\right)}_{\text{SP of water}} \mathrm{d}s' \tag{3}$$



# Forward model

## Continuous version

- TOF of proton $p$:

$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})}$$

- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p(\boldsymbol{x}) = c\sqrt{1 - \frac{m_p c^2}{(m_p c^2 + E_p(\boldsymbol{x}))^2}}$$

- Energy of the proton $p$ at point $\boldsymbol{x} = \Gamma_p(s)$:

$$E_p(\Gamma_p(s)) = E_{\text{in}} - \int_{-\infty}^{s} \rho(\Gamma_p(s')) \times S_{\text{water}}(E_p(\Gamma_p(s')))\mathrm{d}s'$$

# Forward model

## Continuous version

- TOF of proton $p$:

$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})}$$

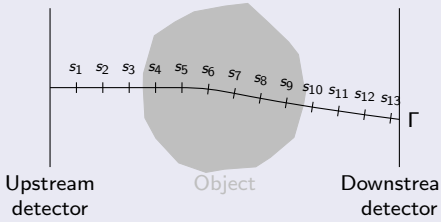
- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p(\boldsymbol{x}) = c\sqrt{1 - \frac{m_p c^2}{\left(m_p c^2 + E_p(\boldsymbol{x})\right)^2}}$$

- Energy of the proton $p$ at point $\boldsymbol{x} = \Gamma_p(s)$:

$$E_p(\Gamma_p(s)) = E_{\text{in}} - \int_{-\infty}^{s} \rho(\Gamma_p(s')) \times S_{\text{water}}(E_p(\Gamma_p(s')))\mathrm{d}s'$$

## Discrete version



# Forward model

## Continuous version

- TOF of proton $p$:

$$\tau_p = \int_{\Gamma_p} \frac{\mathrm{d}\boldsymbol{x}}{v_p(\boldsymbol{x})}$$

- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p(\boldsymbol{x}) = c\sqrt{1 - \frac{m_p c^2}{\left(m_p c^2 + E_p(\boldsymbol{x})\right)^2}}$$

- Energy of the proton $p$ at point $\boldsymbol{x} = \Gamma_p(s)$:

$$E_p(\Gamma_p(s)) = E_{\text{in}} - \int_{-\infty}^{s} \rho(\Gamma_p(s')) \times S_{\text{water}}(E_p(\Gamma_p(s')))\mathrm{d}s'$$

## Discrete version

- TOF of proton $p$:

$$\tau_p \approx \sum_{k=1}^{K} \frac{\Delta_p^{s_k}}{v_p^{s_k}} \tag{4}$$

- Velocity of the proton $p$ at point $\boldsymbol{x}$:

$$v_p^{s_k} = c\sqrt{1 - \frac{m_p c^2}{\left(m_p c^2 + E_p^{s_k}\right)^2}} \tag{5}$$

- Energy of the proton $p$ at point $\boldsymbol{x} = \Gamma_p(s_k)$:

$$E_p^{s_k} = E_p^{s_{k-1}} - \Delta_p^{s_k} \tilde{\rho}_{\boldsymbol{\rho}}\left(\Gamma_p(s_{k-1})\right) S_{\text{water}}(E_p^{s_{k-1}}) \tag{6}$$

# Optimization scheme

- Let
  - $\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) \in \mathbb{R}^P$ the TOFs produced by the forward model with RSP map $\boldsymbol{\rho} \in \mathbb{R}^3$
  - $\boldsymbol{\tau} \in \mathbb{R}^P$ the $P$ TOF measurements

[6] John Duchi, Elad Hazan, and Yoram Singer. In: *Journal of Machine Learning research* 12.7 (2011)

[7] Diederik P. Kingma and Jimmy Ba. In: *CoRR* abs/1412.6980 (2014)

# Optimization scheme

- Let
  - $\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) \in \mathbb{R}^P$ the TOFs produced by the forward model with RSP map $\boldsymbol{\rho} \in \mathbb{R}^3$
  - $\boldsymbol{\tau} \in \mathbb{R}^P$ the $P$ TOF measurements
- The cost function can be defined as

$$\mathcal{L}(\boldsymbol{\rho}) = \|\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) - \boldsymbol{\tau}\|_2^2 \tag{7}$$

[6] John Duchi, Elad Hazan, and Yoram Singer. In: *Journal of Machine Learning research* 12.7 (2011)

[7] Diederik P. Kingma and Jimmy Ba. In: *CoRR* abs/1412.6980 (2014)

# Optimization scheme

- Let
  - $\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) \in \mathbb{R}^P$ the TOFs produced by the forward model with RSP map $\boldsymbol{\rho} \in \mathbb{R}^3$
  - $\boldsymbol{\tau} \in \mathbb{R}^P$ the $P$ TOF measurements
- The cost function can be defined as

$$\mathcal{L}(\boldsymbol{\rho}) = \|\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) - \boldsymbol{\tau}\|_2^2 \tag{7}$$

- The solution to the reconstruction problem is $\rho^*$ such that

$$\rho^* = \arg\min_{\rho} \mathcal{L}(\boldsymbol{\rho}) \tag{8}$$

[6] John Duchi, Elad Hazan, and Yoram Singer. In: *Journal of Machine Learning research* 12.7 (2011)

[7] Diederik P. Kingma and Jimmy Ba. In: *CoRR* abs/1412.6980 (2014)

# Optimization scheme

- Let
  - $\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) \in \mathbb{R}^P$ the TOFs produced by the forward model with RSP map $\boldsymbol{\rho} \in \mathbb{R}^3$
  - $\boldsymbol{\tau} \in \mathbb{R}^P$ the $P$ TOF measurements
- The cost function can be defined as

$$\mathcal{L}(\boldsymbol{\rho}) = \|\hat{\boldsymbol{\tau}}(\boldsymbol{\rho}) - \boldsymbol{\tau}\|_2^2 \tag{7}$$

- The solution to the reconstruction problem is $\rho^*$ such that

$$\rho^* = \arg\min_{\rho} \mathcal{L}(\boldsymbol{\rho}) \tag{8}$$

- Can be iteratively solved using a gradient-descent algorithm (SGD, Adagrad[6], Adam[7])

[6] John Duchi, Elad Hazan, and Yoram Singer. In: *Journal of Machine Learning research* 12.7 (2011)

[7] Diederik P. Kingma and Jimmy Ba. In: *CoRR* abs/1412.6980 (2014)

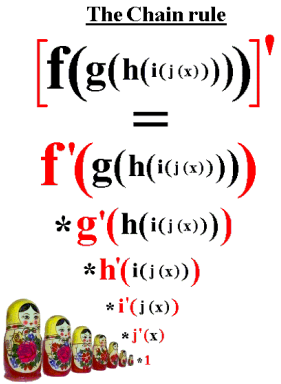
# PyTorch's automatic differentiation engine

- PyTorch is a BSD-licensed C++ deep learning library with Python bindings
- PyTorch provides an automatic differenciation engine named `autograd` (mainly for its backpropagation algorithm)



# Automatic differentiation and the chain rule

Automatic differentiation works by repeatedly applying the chain rule $\left(\frac{\partial x}{\partial z} = \frac{\partial x}{\partial y}\frac{\partial y}{\partial z}\right)$:



# Automatic differentiation and the chain rule

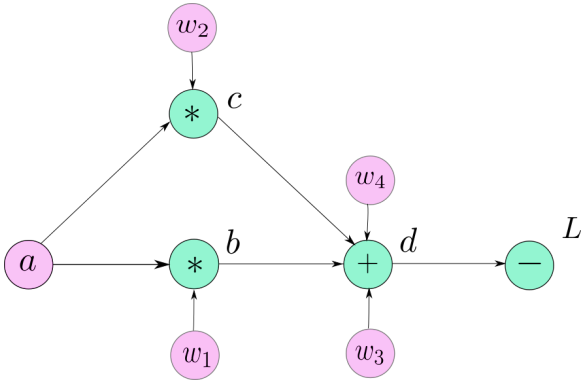
Automatic differentiation works by repeatedly applying the chain rule $\left(\frac{\partial x}{\partial z} = \frac{\partial x}{\partial y}\frac{\partial y}{\partial z}\right)$:




- $\frac{\partial \mathcal{L}}{\partial \rho_i} = \left(\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\tau}}}\right)^\top \frac{\partial \hat{\boldsymbol{\tau}}}{\partial \rho_i}$
- $\frac{\partial \hat{\tau}_p}{\partial \rho_i} = -\sum_{k=1}^{K} \frac{\Delta_p^{s_k}}{(v_p^{s_k})^2} \frac{\partial v_p^{s_k}}{\partial \rho_i}$
- $\frac{\partial v_p^{s_k}}{\partial \rho_i} = \frac{m_p c^2}{\left(E_p^{s_k} + m_p c\right)^3 \sqrt{\frac{-m_p c^2}{(E_p^{s_k} + m_p c)^2} + 1}} \frac{\partial E_p^{s_k}}{\partial \rho_i}$
- ...

# Calculating the derivative with automatic differenciation

- PyTorch's `autograd` builds a graph representing the computation
- The gradients are computed using a `backward` pass that accumulates gradients using the chain rule

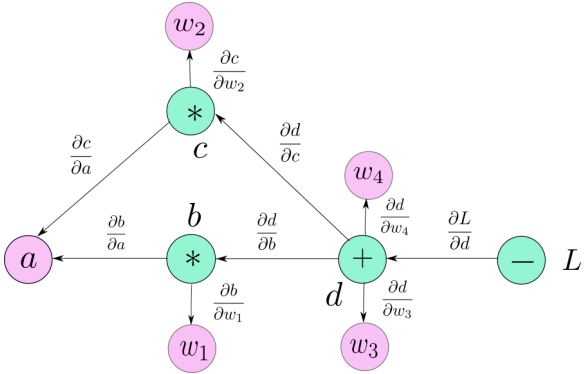



https://www.digitalocean.com/community/tutorials/pytorch-101-understanding-graphs-and-automatic-differentiation

```
import torch
a = 5.
w = torch.tensor([1., 2., 3., 4.],
↪  requires_grad=True)
b = a * w[0]
c = a * w[1]
d = b + c + w[2] + w[3]
L = -d
print(L)  # tensor(-22.,
↪  grad_fn=<NegBackward0>)
L.backward()
print(w.grad)  # tensor([-5., -5., -1.,
↪  -1.])
```

# Calculating the derivative with automatic differenciation

- PyTorch's autograd builds a graph representing the computation
- The gradients are computed using a backward pass that accumulates gradients using the chain rule

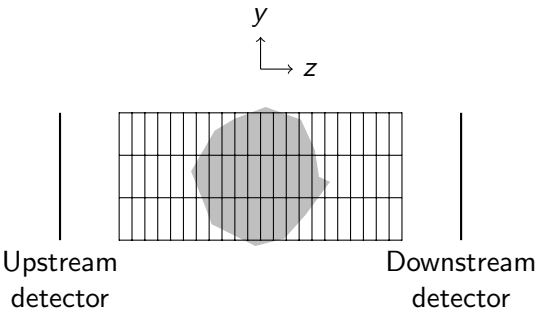



https://www.digitalocean.com/community/tutorials/pytorch-101-understanding-graphs-and-automatic-differentiation

```
import torch
a = 5.
w = torch.tensor([1., 2., 3., 4.],
↪  requires_grad=True)
b = a * w[0]
c = a * w[1]
d = b + c + w[2] + w[3]
L = -d
print(L)  # tensor(-22.,
↪  grad_fn=<NegBackward0>)
L.backward()
print(w.grad)  # tensor([-5., -5., -1.,
↪  -1.])
```

# Parameters

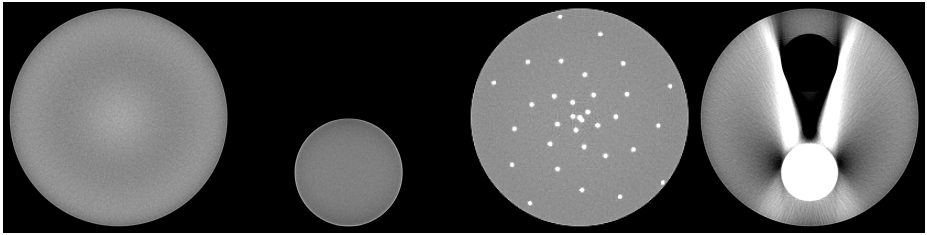
- Optimizer: Adagrad[8]
- Learning rate: 0.3
- Proton step size: 1 mm
- Grid size: $210 \times 3 \times 210$
- Voxel size: $1\,\text{mm} \times 70\,\text{mm} \times 1\,\text{mm}$
- Number of projections: 360



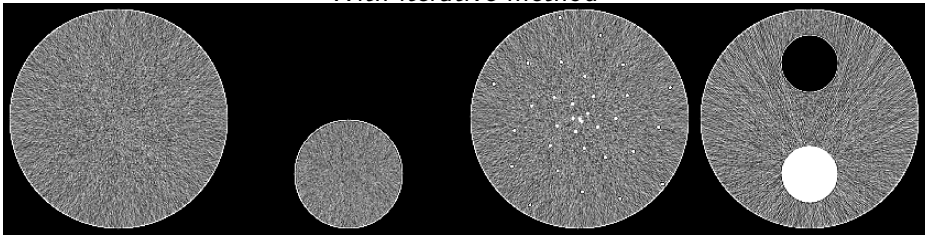

[8] John Duchi, Elad Hazan, and Yoram Singer. In: *Journal of Machine Learning research* 12.7 (2011)

# Reconstructions

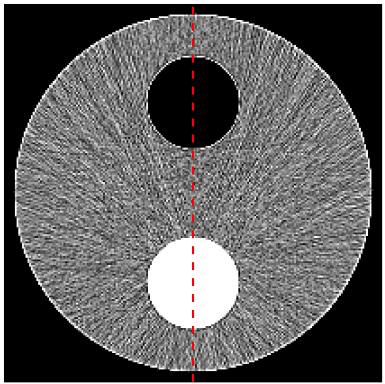
Using calibration curve

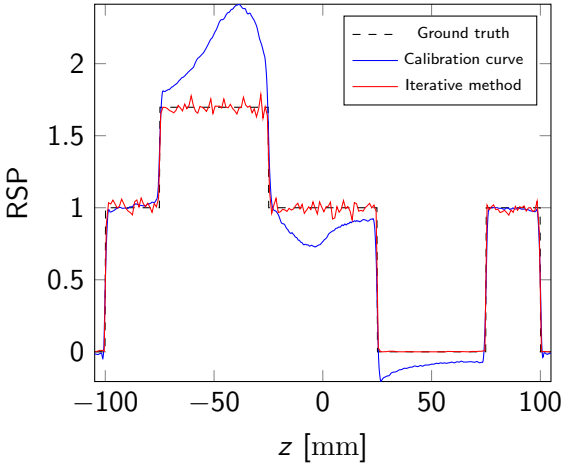


With iterative method



# Profile: bone phantom





# Advantages and limitations of this method

- Advantages:
  - Can reconstruct any kind of (reasonable) phantom
  - Can incorporate a priori information about the object
    - Object support
    - Total variation
    - ...

# Advantages and limitations of this method

- Advantages:
  - Can reconstruct any kind of (reasonable) phantom
  - Can incorporate a priori information about the object
    - Object support
    - Total variation
    - ...
- Limitations:
  - Computational cost are very high
  - Requires to manually tweak some hyperparameters
    - Learning rate
    - Step size along MLPs
    - Number of iterations
    - Optimizer
    - ...

# Outline

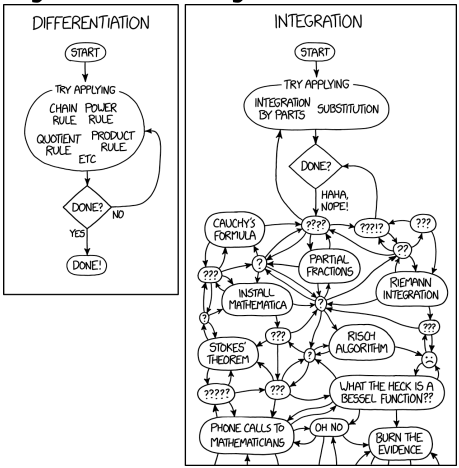
1. Introduction: sandwich time-of-flight proton computed tomography
2. First approach: calibration curve
3. Second approach: gradient descent reconstruction
4. Conclusion

# Conclusion

- The calibration curve approach is simple but provides rough and inaccurate sandwich time-of-flight pCT reconstruction
- Iterative reconstruction of the RSP distribution is possible, at a high computational cost
- Differentiable programming is a powerful tool to optimize complex functions
- The corresponding code is being contributed to Proton Computed Tomography[9]

[9] https://github.com/RTKConsortium/PCT

# Thank you for your attention!




https://xkcd.com/2117/