Implementing AdamW into MAD-NG

Joshua Gray and Félix Carlier

Simulation Parameters

- Designed a very simple ring that included 4 duplicate cells each had:
  - 2 quadrupoles (foc and defoc -> ± 0.1)
  - 2 dipoles
  - 2 sextupoles (foc and defoc -> ± 1)
  - 6 BPMS (one after each magnet)

- The ring was designed to have a circumference of 40 m.

- Each quadrupole and sextupole were given a random error of about $\pm 10\%$ of their strength.

- The dipoles were left untouched.

Gradient Calculation

The aim is to calculate the gradient of the loss with respect to the parameters of the ring (the quadrupole and sextupole strengths). Using the chain rule:

$$\underbrace{\frac{\partial L}{\partial k_n}}_{1 \times \mathrm{NPARAM}} = \underbrace{\frac{\partial L}{\partial x_{\mathrm{BPM}}}}_{1 \times \mathrm{NBPM}} \underbrace{\frac{\partial x_{\mathrm{BPM}}}{\partial k_n}}_{\mathrm{NBPM} \times \mathrm{NPARAM}} + \underbrace{\frac{\partial L}{\partial y_{\mathrm{BPM}}}}_{1 \times \mathrm{NBPM}} \underbrace{\frac{\partial y_{\mathrm{BPM}}}{\partial k_n}}_{\mathrm{NBPM} \times \mathrm{NPARAM}}$$

Where k_n is the strength of the nth quadrupole or sextupole and NPARAM is the the number of quadrupole and sextupole strengths.

Loss Calculation

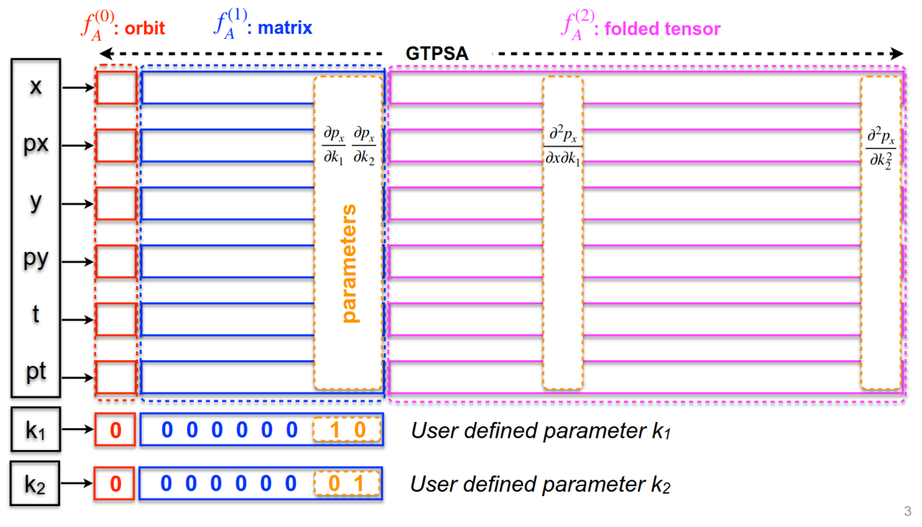
$$L = \frac{1}{2} \sum_{\text{BPMS}} \left[(x_{\text{err, bpm}} - x_{\text{noerr, bpm}})^2 + (y_{\text{err, bpm}} - y_{\text{noerr, bpm}})^2 \right]$$

At each BPM, we get the following derivative of the loss, which then forms a vector the size of the number of BPMs:

$$\frac{\partial L}{\partial x} = (x_{\text{err}} - x_{\text{noerr}})$$

$$\frac{\partial L}{\partial y} = (y_{\text{err}} - y_{\text{noerr}})$$

Derivatives of Coordinate with Parameters



$f_A^{(0)}$: orbit
$f_A^{(1)}$: matrix
$f_A^{(2)}$: folded tensor
GTPSA
x
px
y
py
t
pt
k_1
k_2
parameters
$\frac{\partial p_x}{\partial k_1}$ $\frac{\partial p_x}{\partial k_2}$
$\frac{\partial^2 p_x}{\partial x \partial k_1}$
$\frac{\partial^2 p_x}{\partial k_2^2}$
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
User defined parameter k_1
User defined parameter k_2
3

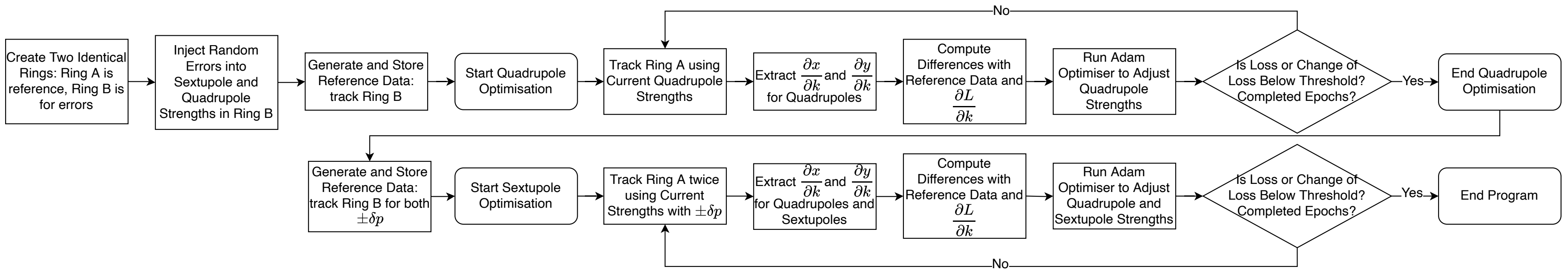
Courtesy of Laurent Deniau. Link to slides

- MAD-NG allows parameters to be included in the coordinates
- We track through the thick maps of the accelerator over several turns
- At each BPM, the derivative is retrieved from the coordinates

AdamW Algorithm

- The AdamW algorithm was taken from the PyTorch documentation and implemented in MAD-NG.
- The only inputs to the algorithm are the gradients of the loss with respect to the parameters and the parameters themselves.
- The algorithm is implemented in a single function, which is called for each iteration/epoch.

Method



Create Two Identical Rings: Ring A is reference, Ring B is for errors
Inject Random Errors into Sextupole and Quadrupole Strengths in Ring B
Generate and Store Reference Data: track Ring B
Start Quadrupole Optimisation
Track Ring A using Current Quadrupole Strengths
Extract $\frac{\partial x}{\partial k}$ and $\frac{\partial y}{\partial k}$ for Quadrupoles
Compute Differences with Reference Data and $\frac{\partial L}{\partial k}$
Run Adam Optimiser to Adjust Quadrupole Strengths
Is Loss or Change of Loss Below Threshold? Completed Epochs?
No
Yes
End Quadrupole Optimisation
Generate and Store Reference Data: track Ring B for both $\pm\delta p$
Start Sextupole Optimisation
Track Ring A twice using Current Strengths with $\pm\delta p$
Extract $\frac{\partial x}{\partial k}$ and $\frac{\partial y}{\partial k}$ for Quadrupoles and Sextupoles
Compute Differences with Reference Data and $\frac{\partial L}{\partial k}$
Run Adam Optimiser to Adjust Quadrupole and Sextupole Strengths
Is Loss or Change of Loss Below Threshold? Completed Epochs?
Yes
No
End Program

Flowchart of the entire process of the algorithm.

Initial coordinates of the form $(x, p_x, y, p_y, t, \delta p)$:

$$\begin{array}{lllllll}
\text{On momentum:} & (1.5\times10^{-3}, & -1\times10^{-4}, & -1\times10^{-3}, & 1\times10^{-4}, & 0, & 0) \\
\text{Off momentum+ :} & (1.5\times10^{-3}, & -1\times10^{-4}, & -1\times10^{-3}, & 1\times10^{-4}, & 0, & 3\times10^{-4}) \\
\text{Off momentum- :} & (1.5\times10^{-3}, & -1\times10^{-4}, & -1\times10^{-3}, & 1\times10^{-4}, & 0, & -3\times10^{-4})
\end{array}$$

Initial Parameters

The following were the input parameters to the quadrupoles and sextupoles:

Magnet Strengths	**1**	**2**	**3**	**4**
Focusing Quadrupole	0.0972	0.103	0.105	0.0932
Defocusing Quadrupole	-0.0987	-0.0936	-0.105	-0.0928
Focusing Sextupole	1.08	0.918	0.994	0.903
Defocusing Sextupole	-1.03	-1.10	-0.909	-0.941

After Quadrupole Optimisation

The following were the relative differences from the initial parameters after fitting just the quadrupoles to a track with the on momentum coordinates:

Relative Errors	**1**	**2**	**3**	**4**
Focusing Quadrupole	2.25×10^{-4}	3.05×10^{-4}	2.88×10^{-4}	6.81×10^{-4}
Defocusing Quadrupole	7.20×10^{-5}	3.49×10^{-4}	5.00×10^{-4}	3.03×10^{-4}

So after this optimisation, an error below 0.1 % was achieved in 2500 iterations and about 40 seconds.

After Off Momentum Optimisation

The following were the relative differences from the initial parameters after fitting both the quadrupoles and sextupoles to two tracks with the off momentum coordinates plus or minus 0.3 %:

Relative Errors	**1**	**2**	**3**	**4**
Focusing Quadrupole	2.30×10^{-6}	3.41×10^{-5}	4.66×10^{-7}	1.02×10^{-5}
Defocusing Quadrupole	3.37×10^{-5}	6.02×10^{-6}	7.80×10^{-6}	2.47×10^{-5}
Focusing Sextupole	4.28×10^{-3}	1.36×10^{-3}	3.29×10^{-3}	1.84×10^{-3}
Defocusing Sextupole	3.15×10^{-3}	3.32×10^{-3}	2.62×10^{-3}	1.94×10^{-3}

So after this optimisation, an error below 0.5 % was achieved in 30,000 iterations and less than 10 minutes.

Summary of Results

- In simulation, it was possible to show a proof of concept of the AdamW algorithm in MAD-NG.
- Propagation of uncertainties and using as weights in the loss function are still to be implemented.
- For this example to achieve such small errors, requires a high precision on the BPM positions - potentially a significant drawback.

What do you think of the current implementation? Are there any improvements you suggest, anything important that I have missed?

Thank you for listening!

Any questions?