High Performance Computing for Optimization of Radiation Therapy Treatment Plans

Felix Liu [1 2] Advisors: Albin Fredriksson [2] Artur Podobas [1] Stefano Markidis [1]

[1]KTH Royal Institute of Technology [2]RaySearch Laboratories

Motivation and Introduction

Mathematical optimization and simulation plays a key role in modern radiation therapy planning. Modern treatments require simulations and optimization to be performed for each patient case, which can be computationally demanding. The overarching goals of this project is:

- Investigate the possibilities to integrate ideas and methods from HPC to improve treatment planning efficiency.
- Explore possibilities to move different related computations to HPC hardware.
- Improve existing algorithms within mathematical optimization to achieve these goals.

Optimization in Radiation Therapy

$$\begin{aligned} \text{minimize}_x \quad & f(d(x)) && \text{(objective function)} \\ \text{subject to} \quad & c_i(d(x)) \leq 0, \quad 1,...,m && \text{(planning constraints)} \\ & x \in \chi && \text{(physical constraint)} \end{aligned}$$

Mathematical optimization in radiation therapy is used to find suitable *treatment plans* for each patient case. We have an inverse problem, where we want to find control parameters for the treatment machines that give desired dose characteristics, such as high uniform dose in the tumor, and low dose to surrounding risk organs.

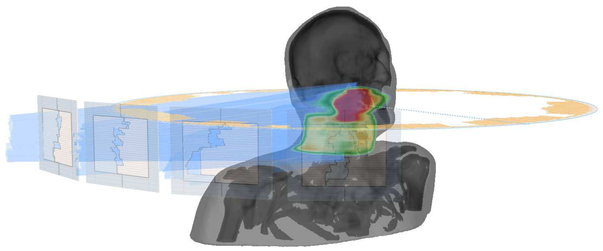


Figure 1. Illustration of patient being irradiated from multiple angles. The color in the phantom shows the accumulated dose, with red indicating higher dose.

Essentially, we shall consider three key computational components of treatment plan optimization

1. *Dose calculation to determine $d(x)$ (often modelled as a linear relationship $d(x) = Ax$).*
2. *Objective function evaluation $f(d(x))$, often weighted sum of separate objectives*
3. *Optimization solver for (non-linear) problems.*

Dose Summation on GPU

In spot scanning proton therapy, dose is a linear function of *spot weights*, which are variables in the optimization problem. I.e. $d(x) = Ax$, where $A \in \mathcal{R}^{n \times d}$, is a sparse matrix. Calculating $d(x)$ (SpMV) can be a significant bottleneck. ***Move to GPU?*** [5]

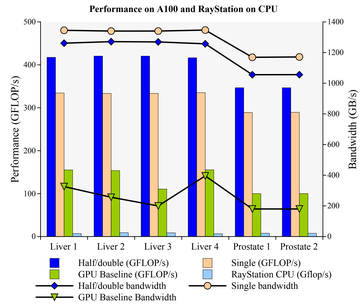


Figure 2. Performance comparison of dose summation implementations. Half/double is our implementation in mixed precision.

Distributed Objective Function Evaluation

The objective $f(x)$ is often a weighted sum of different goals $f(x) = \sum_i f_i(x)$ [2].

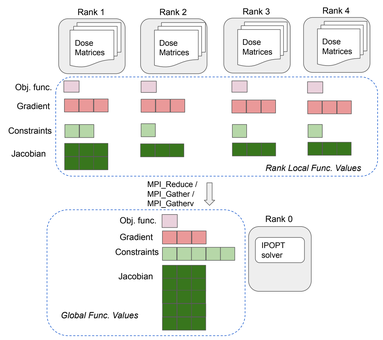
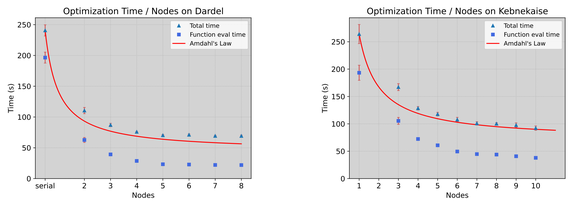


Figure 3. Illustration of distribution of work between MPI ranks.



(a) Parallel scaling on Dardel. (b) Parallel scaling on Kebnekaise.

Figure 4. Scaling tests on Dardel and Kebnekaise. The red line shows the Amdahl's law limit, due to the serial IPOPT optimizer.

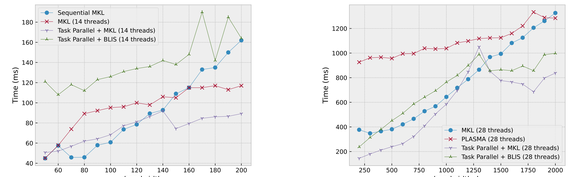
Interior Point Methods (IPM) for Optimization

One of the main computational kernels in IPM is solving a linear system arising from Newton's method on (perturbed) first-order optimality conditions. We investigate two avenues:

1. Task-based parallel Cholesky Factorization for banded matrices
2. Solving the Newton systems using iterative linear algebra (active research topic in the optimization community)

Task-based parallel Cholesky for banded matrices

We propose a task-based parallel method for Cholesky factorization using OpenMP tasks, and the standard LAPACK layout for banded matrices [4].



(a) Smaller matrix bandwidths (b) Larger matrix bandwidths

Figure 5. Performance comparison of banded Cholesky factorization implementations on Kebnekaise.

Iterative Solvers for Newton Systems in IPM

System to solve is on *doubly augmented* form [1]:

$$\begin{pmatrix} Q + 2B^T D^{-1} B & B^T \\ B & D \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda_A \end{pmatrix} = \begin{pmatrix} r_1 + 2B^T D^{-1} r_2 \\ r_2 \end{pmatrix}.$$

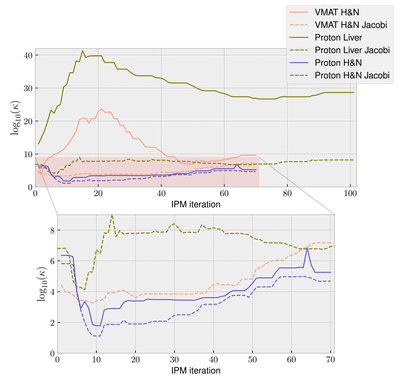
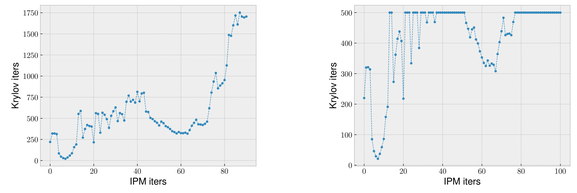


Figure 6. Condition Numbers of KKT matrices.

Challenge: *Ill-conditioning of linear systems. (sometimes extreme)*

Prototype Implementation

Solve KKT-system using Jacobi preconditioned conjugate gradients. We implemented a prototype interior point method and evaluated its performance on real-world optimization problems from radiation therapy planning [3].



(a) CG iterations required in each optimizer step. (b) CG iterations required in each optimizer step, with a maximum of 500.

Figure 7. CG iterations until convergence in each step in the interior point optimizer.

Our results indicated that our prototype interior point method can find sufficiently accurate solutions in reasonable time. ***Promising for GPU acceleration to extract more performance!***

Future Research Directions

- Porting of the interior point method prototype to GPU
- Exploring other preconditioners.
- Other optimization algorithms usable? (First-order method, unconstrained methods, etc.)

Conclusions

- Modern treatment planning (in clinics) makes use of GPU acceleration for many calculations already.
- Constrained optimization algorithm remain challenging to port.
- Krylov iterative solvers show promise for problems from radiation therapy, and may provide an avenue forward.

References

[1] Anders Forsgren, Philip E Gill, and Joshua D Griffin. Iterative solution of augmented systems arising in interior methods. *SIAM Journal on Optimization*, 18(2):666–690, 2007.

[2] Felix Liu, Måns I Andersson, Albin Fredriksson, and Stefano Markidis. Distributed objective function evaluation for optimization of radiation therapy treatment plans. In *International Conference on Parallel Processing and Applied Mathematics*, pages 383–395. Springer, 2022.

[3] Felix Liu, Albin Fredriksson, and Stefano Markidis. Krylov solvers for interior point methods with applications in radiation therapy. *arXiv preprint arXiv:2308.00637*, 2023.

[4] Felix Liu, Albin Fredriksson, and Stefano Markidis. Parallel cholesky factorization for banded matrices using openmp tasks. *arXiv preprint arXiv:2305.04635*, 2023.

[5] Felix Liu, Niclas Jansson, Artur Podobas, Albin Fredriksson, and Stefano Markidis. Accelerating radiation therapy dose calculation with nvidia gpus. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 449–458. IEEE, 2021.

felixliu@kth.se