INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT

eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 10 (2025)

# Accelerating Proton Therapy Dose Calculation: A Comprehensive Study of GPU-Based Pencil Beam and Monte Carlo Approaches for Clinical Adaptivity

**Dr. Aiden R. Moreau**

Global Institute for Computational Medicine, École Universitaire de Technologie, France

## Abstract

**Background**: The demand for fast, accurate dose calculation methods in proton therapy has spurred research into leveraging Graphics Processing Units (GPUs) for both pencil beam algorithms and full Monte Carlo simulations. This paper synthesizes methodological advances and evaluates the computational and clinical trade-offs of GPU-accelerated approaches, outlining pathways to integrate sub-second dose calculation into adaptive clinical workflows.

**Methods**: Building on foundational work in GPU programming and parallel computing, this study constructs a conceptual pipeline that integrates contemporary pencil beam models and GPU Monte Carlo techniques. The pipeline emphasizes runtime code generation, memory-conscious data structures, and validated physical models for proton transport and scattering. Methodological design choices are aligned with historical GPU acceleration practices and modern clinical constraints. Performance and accuracy trade-offs are analyzed qualitatively and quantitatively in relation to published benchmarks: fast pencil beam approximations (Da Silva et al., 2015; Fujimoto et al., 2011), GPU Monte Carlo platforms (Perl et al., 2012; Lee et al., 2022), and validated commissioning procedures (Azcona et al., 2023).

**Results**: The articulated GPU pipeline demonstrates that sub-second pencil beam computations are attainable for individual fields using a combination of double-Gaussian beam models, hierarchical memory access patterns, and optimized reduction strategies—consistent with previously reported sub-second results (Da Silva et al., 2015). When contrasted with GPU Monte Carlo methods, pencil beam techniques offer orders-of-magnitude speed advantages at the cost of limited modeling fidelity for heterogeneous media and complex scattering scenarios (Perl et al., 2012; Lee et al., 2022). GPU Monte Carlo implementations, while more computationally demanding, provide superior dosimetric fidelity that is advantageous for commissioning and adaptive replanning when clinical constraints permit longer computation windows (Paganetti et al., 2021; Azcona et al., 2023).

**Discussion**: We provide a deep interpretive analysis of algorithmic design, including the theoretical underpinnings of pencil beam decompositions and Monte Carlo sampling strategies, the implications of parallel hardware architectures on numerical stability and reproducibility, and the practical considerations of clinical integration—quality assurance, commissioning, and workflow compatibility. Limitations include the challenge of balancing speed, accuracy, and validation demands; potential numerical artifacts introduced by aggressive optimization; and the need for robust clinical validation across tissue heterogeneities (Schreuder et al., 2019; Goma et al., 2018). Future directions center on hybrid models that fuse fast pencil beam precomputations with targeted Monte Carlo refinement, automated commissioning pipelines, and exploitation of modern GPU programming paradigms and run-time code generation (Klöckner et al., 2012; Lulla, 2025).

**Conclusions**: GPU-accelerated dose calculation is mature enough to dramatically improve the feasibility of adaptive proton therapy. Through careful algorithmic design, validation, and clinical integration, facilities can realize near real-time planning for selected adaptive scenarios while retaining Monte Carlo accuracy where it matters most.

**Keywords:** proton therapy, GPU acceleration, pencil beam, Monte Carlo, adaptive radiotherapy, dose calculation, run-time code generation

## INTRODUCTION

Radiation therapy continues to evolve toward increasingly conformal and adaptive interventions, driven by technological advances in beam delivery systems, imaging, and computational resources. Proton therapy stands out among modalities for its physical dose characteristics—chiefly the Bragg peak—which permit superior sparing of normal tissues when compared to conventional photon therapy. However, the same physics that provide dosimetric advantages also increase the complexity of accurate dose calculation. Proton transport is sensitive to tissue heterogeneities, range uncertainties, and nuclear interactions, which complicates deterministic modeling and favors stochastic methods such as Monte Carlo for highest-fidelity dose prediction (Perl et al., 2012; Paganetti et al., 2021).

Clinical adoption of Monte Carlo has historically been hampered by computational expense. This limitation has spurred research into surrogate and hybrid models—pencil beam algorithms and double-Gaussian representations among them—that trade some physical fidelity for computational speed, enabling practical clinical runtimes for tasks such as plan optimization and initial dose estimation (Soukup et al., 2005; da Silva et al., 2015). Simultaneously, the maturation of commodity parallel compute platforms—namely Graphics Processing Units (GPUs)—has renewed interest in bringing Monte Carlo into routine clinical use by dramatically reducing runtimes (Larsen & McAllister, 2001; Owens et al., 2007).

The parallel computing landscape has been shaped by studies that chart both the opportunities and constraints of GPUs for scientific computing (Asanović et al., 2006; Owens et al., 2007). Pioneering algorithms for matrix multiplication and general computational kernels demonstrated that GPUs could outperform traditional CPUs by leveraging massive parallelism, but also highlighted the need for algorithmic redesign to exploit memory hierarchies and thread-level parallelism (Larsen & McAllister, 2001; Brodtkorb et al., 2012). In proton therapy, these insights translate into algorithmic adaptations for particle propagation, collision modeling, and scoring that avoid serial bottlenecks and maximize data locality (Fujimoto et al., 2011; Da Silva et al., 2015).

Recent open-source implementations—both research and clinical—have further advanced GPU Monte Carlo into practical territory. Platforms like TOPAS and newer open-source GPU-native codes provide validated frameworks for research and clinical application, elucidating best practices for accuracy, QA, and commissioning (Perl et al., 2012; Lee et al., 2022). At the same time, carefully optimized pencil beam approaches implemented on GPUs have achieved sub-second runtimes for single-field dose calculations, making them attractive for rapid adaptive interventions (Da Silva et al., 2015).

Despite these advances, the field faces unresolved tensions: (1) how to integrate fast, approximate models with high-fidelity Monte Carlo in a clinically safe manner; (2) how to structure software for maintainability, portability, and clinical validation; and (3) how to design commissioning and QA processes that reflect the hybrid nature of modern pipelines (Azcona et al., 2023; Schreuder et al., 2019). Moreover, hardware improvements and programming models—scripting-based run-time code generation, improvements in GPU memory and interconnects, and heterogeneous computing—offer new opportunities but require careful translation into clinical-grade software engineering practices (Klöckner et al., 2012; Lulla, 2025).

This work acts as a comprehensive synthesis and prescriptive guide for designing a clinically oriented GPU-accelerated dose calculation pipeline for proton therapy. It draws upon foundational GPU programming literature and contemporary proton dose calculation research to produce a methodology that honors both the physics and the clinical constraints required for adoption. The aims of this manuscript are threefold: to articulate the algorithmic foundations and practical implementations for GPU-accelerated pencil beam and Monte Carlo dose calculation; to evaluate trade-offs in speed and accuracy with respect to clinical needs; and to propose integration pathways and validation strategies to enable safe deployment within adaptive proton therapy workflows.

## METHODOLOGY

The methodology section delineates a conceptualized, reproducible approach to developing and validating a GPU-accelerated proton dose engine that supports both pencil beam and Monte Carlo modalities. Emphasis is placed on software architecture, algorithmic kernels optimized for GPU execution, data structures for memory-efficient operation, and validation strategies that conform to clinical commissioning expectations.

### Software Design Philosophy

The software architecture adheres to separation-of-concerns principles: high-level orchestration, physics kernels, memory management, and I/O are modularized to permit independent optimization and testing. This mirrors recommendations from general-purpose GPU computation surveys and run-time code generation frameworks, which emphasize modularity for maintainability and performance portability (Owens et al., 2007; Klöckner et al., 2012). The system exposes both a scripting interface for rapid prototyping (Python bindings) and a compiled core for performance-critical components, enabling a workflow that balances developer productivity with runtime efficiency—a pattern underlined by scripting-based GPU run-time code generation approaches (Klöckner et al., 2012).

### Programming Models and Runtime Code Generation

Two complementary programming modes are advocated: a high-level control plane using a scripting language (e.g., Python) and a GPU-executable kernel plane employing either CUDA or OpenCL. The scripting plane performs job orchestration, patient data pre-processing (CT-to-stopping-power conversions), and post-processing of dose. The kernel plane handles the computationally intensive tasks: pencil beam convolution, particle transport steps, interaction sampling, and dose scoring. Run-time code generation allows kernel specializations based on machine capability and patient geometry (Klöckner et al., 2012). Specialized kernels can be generated on-the-fly to exploit GPU features like warp-level intrinsics and shared memory tiling without requiring multiple precompiled binaries, improving portability and maintainability.

### Data Management and Memory Layout

Efficient GPU utilization requires attention to memory coalescence and minimizing divergent control flow. Data structures are organized to favor structure-of-arrays layouts for frequently accessed particle attributes (position, direction cosine, energy) to permit contiguous memory access across threads, thereby enhancing throughput (Owens et al., 2007). For Monte Carlo, particles are batched into work queues sized to the GPU's concurrency capability, with active particles occupying contiguous buffers. Dose scorers use spatially partitioned accumulation buffers with atomic accumulation strategies in device memory or per-thread-block accumulation followed by hierarchical reduction to minimize atomic contention—a strategy consistent with high-performance GPU reductions and prior GPU Monte Carlo implementations (Fujimoto et al., 2011; Lee et al., 2022).

### Physical Models

Pencil Beam Representation

For pencil beam calculations, the double-Gaussian beam model is selected for its balance of simplicity and improved lateral profile representation relative to single-Gaussian models (da Silva et al., 2015). The model decomposes a clinical field into discrete pencil beams, each described by an energy spectrum, central axis, and two Gaussian lateral components (core and halo) whose widths evolve with depth. Stopping-power integration is performed along central rays using calibrated CT-to-stopping-power mappings (Schneider et al., 1996; Goma et al., 2018). Heterogeneity corrections are applied using path-length scaling and lateral scaling heuristics that are standard in pencil beam algorithms, acknowledging their limitations in complex heterogeneities (Soukup et al., 2005).

### Monte Carlo Transport

The Monte Carlo engine simulates proton transport discretely, sampling interactions (elastic, inelastic, nuclear reactions) based on cross-section models appropriate for clinical energy ranges. The engine implements condensed-history steps for continuous electromagnetic slowing with discrete sampling of large-angle scatter and nuclear interactions following practices established in clinical Monte Carlo tools (Perl et al., 2012). Specialized variance reduction strategies—such as importance sampling, particle splitting near scoring regions, and Russian roulette for low-weight particles—are implemented judiciously to reduce variance while preserving physical fidelity. Outputs include volumetric dose tallies and standard uncertainty estimates.

### Parallelization Strategy

Both pencil beam convolution and Monte Carlo sampling are mapped to GPUs by distributing independent work units across thread blocks. For pencil beam, each pencil is handled by a thread block that evaluates depth-wise dose deposition and lateral spreading, employing shared memory for profile kernels and read-only caches for material properties. For Monte Carlo, each particle is treated as an independent track processed by a GPU thread, with careful handling of divergent branching through stackless kernel designs that group particles by similar states to minimize divergence (Owens et al., 2007). Inter-kernel load balancing is achieved by dynamic work queues on the device that collect newly generated secondary particles and assign them to idle threads, a pattern demonstrated in efficient GPU Monte Carlo codes (Lee et al., 2022).

### Validation, Commissioning, and Quality Assurance

Validation proceeds on multiple levels. Kernel-level unit tests verify energy deposition for simple geometries and analytic cases. System-level validation compares engine outputs against benchmark Monte Carlo results (e.g., TOPAS) and experimental measurements in water and heterogeneous phantoms (Perl et al., 2012; Schreuder et al., 2019). Commissioning follows established procedures for synchrotron-based systems and scanning beams, requiring beam model tuning against measured depth-dose and lateral profiles and confirming CT calibration for stopping powers (Azcona et al., 2023; Schneider et al., 1996). Monte Carlo-to-measurement deviations are quantified using gamma analysis and range difference metrics, with thresholds set according to clinical practice and regulatory expectations.

### Performance Metrics and Benchmarking

Benchmark workloads encompass single-field dose calculation for typical clinical field sizes, multi-field composite plans, and Monte Carlo-based high-fidelity simulations for select regions. Performance is measured in wall-clock time, throughput (particles/sec), dose map generation time, and statistical uncertainty for Monte Carlo tallies. Scaling tests examine GPU occupancy, memory throughput, and multi-GPU distribution strategies for larger workloads. Benchmarks are contextualized against published results demonstrating sub-second pencil beam calculations (Da Silva et al., 2015) and GPU Monte Carlo runtimes (Perl et al., 2012; Lee et al., 2022).

### Clinical Integration Considerations

Workflow integration encompasses DICOM RT plan ingestion, CT preprocessing, beam model selection, dose calculation, plan evaluation, and export for treatment delivery. Emphasis is placed on automated QA gates: verification against commissioning baselines, secondary dose calculation cross-checks using a different algorithm, and uncertainty propagation reporting to clinicians. An adaptive workflow is proposed wherein pencil beam calculations provide immediate dose feedback for decision-making, with targeted Monte Carlo recalculations for regions where precision is critical, such as near critical structures or in the presence of high-density implants (Paganetti et al., 2021).

### Ethical and Regulatory Considerations

Deployment of GPU-accelerated dose engines requires adherence to medical device regulations, robust documentation of software lifecycle processes, and traceable QA records. Validation and ongoing quality assurance practices follow radiation oncology standards to ensure patient safety. Open-source components should be treated with the same rigor as proprietary code, with strict versioning, reproducible build processes, and cryptographic checksums for released binaries.

## RESULTS

This section synthesizes expected outcomes and quantitative relationships grounded in the literature and the conceptual pipeline articulated above. Because this manuscript builds a methodological framework rather than reporting new experimental data, results are presented as an evaluative synthesis of performance, accuracy, and clinical viability drawn from published studies and from the theoretical implications of the described implementation strategies.

### Performance Achievements with Pencil Beam on GPUs

Published reports and implementations indicate that GPU-accelerated pencil beam methods can achieve sub-second to low-second runtimes for single-field dose calculations depending on field size, beam energy spread, and GPU generation (Da Silva et al., 2015; Fujimoto et al., 2011). These results are consistent with the following mechanistic understanding: pencil beam convolution reduces the three-dimensional dose integration problem to a set of independent depth-wise integrations convolved with lateral kernels, enabling massive parallelism across pencil beams and depths. Kernel-level optimizations—such as precomputation of lateral kernels, use of shared memory for kernel caching, and coalesced memory accesses for beam parameters—yield high occupancy and memory bandwidth utilization on modern GPUs, leading to substantial speedups relative to CPU implementations (Owens et al., 2007).

Quantitatively, the sub-second benchmark reported by Da Silva et al. (2015) for pencil beam calculations is achieved under representative clinical conditions using a double-Gaussian model and a carefully tuned GPU kernel pipeline (Da Silva et al., 2015). Extrapolating from kernel throughput and GPU memory bandwidth metrics, we expect single-field runtimes to increase roughly linearly with the number of pencil beams and the depth sampling resolution, while being sub-linear with respect to energy layers if per-layer batch processing and kernel reuse are employed. The practical implication is that clinical plans with moderate field complexity can be computed sufficiently fast to support near-real-time decision making when using pencil beam models on GPUs.

### Monte Carlo on GPUs: Throughput Versus Fidelity

GPU Monte Carlo platforms report significantly greater computational demands but produce higher-fidelity dosimetry, especially in heterogeneous media and near interfaces where deterministic models struggle (Perl et al., 2012; Lee et al., 2022). GPU Monte Carlo codes like MOQUI have demonstrated clinically relevant runtimes by exploiting data-parallel particle transport and optimized memory layouts, but still typically require longer runtimes than pencil beam methods—ranging from seconds to minutes depending on statistical precision requirements (Lee et al., 2022).

Statistical uncertainty in Monte Carlo tallies scales approximately with the inverse square root of the number of histories; hence, achieving a halving of uncertainty requires roughly a fourfold increase in histories and computation. GPUs ameliorate this cost by enabling much higher histories-per-second than CPUs, but the fundamental scaling law imposes diminishing returns. Variance reduction techniques and targeted refinement strategies (e.g., splitting in regions of interest) can effectively allocate computational effort to clinically relevant regions and reduce overall runtime for a desired uncertainty distribution (Perl et al., 2012).

### Hybrid Strategy Outcomes

By integrating pencil beam for initial rapid estimation and Monte Carlo for selective refinement, a hybrid workflow can reduce end-to-end adaptive planning time while preserving dosimetric fidelity where it matters most. For instance, a pipeline might use GPU pencil beam calculations to evaluate candidate adjustments in seconds and then trigger Monte Carlo recalculations for a narrowed set of fields or sub-volumes, limiting Monte Carlo computational expense while achieving clinically acceptable accuracy. Empirical case studies in the literature suggest that such hybrid tactics can deliver clinically usable results with modest computational resources while meeting QA thresholds established through commissioning studies (Da Silva et al., 2015; Azcona et al., 2023).

### Accuracy and Clinical Acceptability

Accuracy comparisons between pencil beam and Monte Carlo methods emphasize scenarios where pencil beam models are most challenged—heterogeneous tissues, high-density implants, and small, highly modulated fields. Empirical studies have demonstrated measurable deviations in range and dose distributions for pencil beam algorithms in the presence of pronounced heterogeneities, while Monte Carlo retains accuracy under these conditions (Schreuder et al., 2019; Goma et al., 2018). Consequently, clinical deployment must carefully assess acceptable error bounds and define safeguard procedures (e.g., automatic Monte Carlo confirmation when heterogeneity metrics exceed thresholds) to prevent clinically significant discrepancies.

### Commissioning and Validation Results

Commissioning procedures for GPU-accelerated engines align with established practices requiring measurement-based tuning of beam models and cross-validation with trusted Monte Carlo benchmarks (Azcona et al., 2023). Practical results from prior commissioning efforts indicate that GPU-enabled engines can match measurement-based commissioning metrics—such as depth-dose curves and lateral profiles—to within clinically accepted tolerances after careful parameterization and iterative model calibration (Azcona et al., 2023; Schneider et al., 1996).

### Scalability and Multi-GPU Considerations

For large workloads or high-precision Monte Carlo simulations, multi-GPU scaling is feasible but introduces communication and load-balancing considerations. Partitioning strategies—either spatial partitioning of the dose scorer or particle-based distribution—affect scalability and memory duplication overheads. Empirical evidence suggests near-linear scaling for well-partitioned problems with minimal inter-GPU communication; however, as the number of GPUs increases, efficiencies diminish due to communication overheads and potential duplication of large data structures (Owens et al., 2007). Practical multi-GPU deployment must balance these trade-offs and may favor single-node multi-GPU systems with high-speed interconnects for clinical environments.

### Software Maintainability and Reproducibility

Adoption in clinical contexts requires software that is maintainable and reproducible. Run-time code generation facilitates portability by producing optimized kernels tailored to the target GPU architecture, and scripting interfaces simplify integration into clinical pipelines (Klöckner et al., 2012). However, run-time generation also requires robust reproducibility strategies—deterministic build seeds and explicit kernel caching mechanisms—to ensure consistent behavior across deployments. Traceable version control and binary release practices are thus essential to meet clinical regulatory requirements.

## DISCUSSION

This section interprets the methodological and synthesized results in depth, articulating theoretical implications, counter-arguments, limitations, and a forward-looking research agenda.

### Interpretation of Performance-Accuracy Trade-offs

The juxtaposition between pencil beam and Monte Carlo approaches reveals a quintessential trade-off in computational radiotherapy: speed versus physical fidelity. Pencil beam models achieve remarkable speed because they compress the many-body problem of particle interactions into parameterized kernels and deterministic integrals. This compression is justified in relatively homogeneous media or when clinical tolerances permit small deviations, enabling workflows—such as online adaptive replanning—where immediacy is paramount (Soukup et al., 2005; da Silva et al., 2015).

Conversely, Monte Carlo's fidelity stems from simulating the underlying stochastic physics. This fidelity is crucial where clinical decisions hinge upon fine spatial accuracy, such as dose near critical structures or in regions where tissue heterogeneity causes significant scattering or range shifts. GPU acceleration substantially reduces Monte Carlo runtimes, but the fundamental statistical nature of Monte Carlo imposes a computational floor for uncertainty that cannot be circumvented without introducing biased variance reduction. Hence, Monte Carlo remains the gold standard for commissioning and final verification while pencil beam methods offer a pragmatic tool for immediate decision-making.

### Theoretical Implications for Algorithm Design

The architecture of GPUs—massive parallelism combined with complex memory hierarchies—requires algorithmic designs that differ from traditional CPU-centric thinking (Asanović et al., 2006; Owens et al., 2007). Algorithms must be recast to maximize throughput via data parallelism and to minimize thread divergence. For pencil beam convolution, this manifests as decomposing the domain into homogeneous processing elements (pencil beams) that map cleanly to thread blocks. For Monte Carlo, it motivates stackless kernel designs and grouping by particle state to reduce divergence. The implication for future algorithmic research is clear: radiotherapy algorithms should be formulated with hardware-aware primitives—reductions, prefix-sums, and tiled memory patterns—as fundamental building blocks rather than as afterthoughts.

### Reproducibility and Numerical Stability

Aggressive optimizations and parallel reductions raise concerns about numerical stability and bitwise reproducibility. Floating-point reductions in parallel can produce non-deterministic sums depending on reduction order; while clinically irrelevant in many contexts, such differences can complicate regression testing and regulatory traceability. Two complementary strategies mitigate this issue: (1) use of numerically stable accumulation techniques (e.g., Kahan summation variants adapted for parallelism) and (2) deterministic scheduling of reductions through controlled work partitioning and fixed reduction trees. While these strategies may slightly increase runtime, they provide the reproducibility needed for clinical acceptance and for consistent commissioning baselines.

## Validation Strategies and Clinical Safety

Any clinical deployment must satisfy rigorous validation and quality assurance practices. This requirement implies a hierarchical validation strategy: kernel unit tests, integration tests against analytical benchmarks, phantom studies with measured dosimetry, and clinical end-to-end tests using retrospective cases. Additionally, commissioning must include CT calibration procedures and verification against external dosimetry for multiple beam energies and configurations (Schneider et al., 1996; Azcona et al., 2023). A particularly important clinical safeguard is the automatic detection of cases where pencil beam approximations may fail (e.g., presence of high-density implants, steep heterogeneity gradients), triggering mandatory Monte Carlo verification before clinical approval.

## Counter-Arguments and Nuanced Considerations

Some practitioners argue that given the availability of high-performance clusters and cloud resources, the emphasis on pencil beam speed is less critical—Monte Carlo can be used more liberally when compute resources permit. While this is a compelling viewpoint for research and commissioning, it underestimates practical clinical constraints: latency requirements for online adaptation, data transfer delays, and regulatory considerations concerning cloud-based patient data handling. Moreover, not all centers possess scalable cloud resources or the regulatory framework to offload protected health information. Therefore, fast local methods retain practical value for many centers.

Another counter-argument concerns the risk of over-reliance on GPU hardware that may become obsolete or unsupported. This risk underscores the importance of architectural portability—using abstraction layers and run-time code generation to target multiple backends and to permit fallback to CPU implementations if necessary. Software engineering rigor and modular design help insulate clinical workflows from hardware obsolescence.

## Limitations

The present paper synthesizes methodologies and literature but does not present novel experimental data. Thus, the conclusions are inferential and rely on the published performance and validation data available in the literature. Another limitation is that hardware-specific optimizations may not generalize across GPU generations; performance claims should be contextualized to particular hardware classes and driver ecosystems. Additionally, clinical workflows vary widely across institutions, so the proposed pipeline may need considerable adaptation to local standards, imaging systems, and regulatory environments.

## Future Directions

Several research avenues promise to further bridge the speed-accuracy divide. Hybrid methods that dynamically combine pencil beam precomputations with localized Monte Carlo refinement could provide the best of both worlds for adaptive workflows. Machine learning models trained to predict regions where Monte Carlo corrections are necessary may further optimize computational allocation. Advances in heterogeneous computing—combining GPU, CPU, and specialized accelerators—could enable new partitioning strategies that exploit each device's strengths. Continued work on run-time code generation and portable high-level abstractions will ease maintenance burdens and accelerate clinical translation (Klöckner et al., 2012; Lulla, 2025).

### Operational Recommendations

For centers aiming to adopt GPU-accelerated dose calculation, a staged approach is recommended: begin with GPU-accelerated pencil beam for low-risk, fast-turnaround tasks while implementing robust QA and commissioning; parallelly pilot GPU Monte Carlo workflows for commissioning and targeted verification; and finally, integrate hybrid workflows with automated verification triggers to ensure clinical safety. Investment in software engineering practices—automated testing, continuous integration, and reproducible builds—is as essential as computational hardware to ensure reliable clinical operation.

## CONCLUSION

GPU acceleration has transformed the computational landscape for proton therapy dose calculation, enabling both sub-second pencil beam computations and practical GPU Monte Carlo simulations. Each approach serves different clinical needs: pencil beam methods provide the speed required for rapid clinical decision-making, while Monte Carlo offers the fidelity necessary for commissioning and high-stakes verification. A hybrid paradigm—leveraging pencil beam for immediate feedback and Monte Carlo for selective refinement—emerges as a pragmatic path toward routine clinical adaptivity.

Implementation success depends not only on algorithmic choices but equally on rigorous validation, commissioning, and software engineering that meets clinical and regulatory standards. Run-time code generation and scripting-based orchestration can reconcile developer productivity with the performance needs of GPUs, but they must be paired with deterministic operational modes for clinical reproducibility. Future work should explore automated triage systems that identify regions requiring Monte Carlo verification, hybrid algorithms that reduce Monte Carlo footprints, and continued investments in portable, maintainable software frameworks.

Ultimately, the integration of GPU-accelerated dose calculation into clinical practice stands to significantly enhance the responsiveness and precision of proton therapy. By aligning computational design with clinical workflows and validation imperatives, centers can translate hardware and algorithmic advances into improved patient care.

## REFERENCES

1. Larsen, E.; McAllister, D. Fast matrix multiplies using graphics hardware. In Proceedings of the 2001 ACM/IEEE Conference on Supercomputing, SC’01, Denver, CO, USA, 10–16 November 2001. CrossRef.

2. Owens, J.; Luebke, D.; Govindaraju, N.; Harris, M.; Krüger, J.; Lefohn, A.; Purcell, T. A Survey of General-Purpose Computation on Graphics Hardware. Comput. Graph. Forum 2007, 26, 80–113. CrossRef.

3. Nanz, S.; Furia, C. A Comparative Study of Programming Languages in Rosetta Code. In Proceedings

of the IEEE International Conference on Software Engineering, Florence, Italy, 16–24 May 2015; Volume 1, pp. 778–788. CrossRef.

4. Klöckner, A.; Pinto, N.; Lee, Y.; Catanzaro, B.; Ivanov, P.; Fasih, A. PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation. Parallel Comput. 2012, 38, 157–174. CrossRef.

5. Asanović, K.; Bodik, R.; Catanzaro, B.; Gebis, J.; Husbands, P.; Keutzer, K.; Patterson, D.; Plishker, W.; Shalf, J.; Williams, S.; et al. The Landscape of Parallel Computing Research: A View from Berkeley; Technical Report; EECS Department, University of California: Berkeley, CA, USA, 2006.

6. Brodtkorb, A. Simplified Ocean Models on the GPU. Available online: https://sintef.brage.unit.no/sintef-xmlui/bitstream/handle/11250/2565319/500-Article2bText-1025-1-10-20180815.pdf?sequence=2&isAllowed=y (accessed on 7 January 2020).

7. Holm, H.; Brodtkorb, A.; Christensen, K.; Broström, G.; Sætra, M. Evaluation of Selected Finite-Difference and Finite-Volume Approaches to Rotational Shallow-Water Flow. Commun. Comput. Phys. 2019, accepted.

8. Hagen, T.; Henriksen, M.; Hjelmervik, J.; Lie, K.A. How to Solve Systems of Conservation Laws Numerically Using the Graphics Processor as a High-Performance Computational Engine. In Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF; Hasle, G., Lie, K.A., Quak, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 211–264. CrossRef.

9. Brodtkorb, A.; Sætra, M.; Altinakar, M. Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation. Comput. Fluids 2012, 55, 1–12. CrossRef.

10. Perl, J.; Shin, J.; Schümann, J.; Faddegon, B.; Paganetti, H. TOPAS: an innovative proton Monte Carlo platform for research and clinical applications. Med Phys. 2012;39(11):6818-6837. doi:10.1118/1.4758060.

11. Lee, H.; Shin, J.; Verburg, J.M.; et al. MOQUI: an open-source GPU-based Monte Carlo code for proton dose calculation with efficient data structure. Phys Med Biol. 2022;67(17):174001. doi:10.1088/1361-6560/ac8716.

12. Soukup, M.; Fippel, M.; Alber, M. A pencil beam algorithm for intensity modulated proton therapy derived from Monte Carlo simulations. Phys Med Biol. 2005;50(21):5089-5104. doi:10.1088/0031-9155/50/21/010.

13. da Silva, J.; Ansorge, R.; Jena, R. Fast pencil beam dose calculation for proton therapy using a double-Gaussian beam model. Front Oncol. 2015;5:281. doi:10.3389/fonc.2015.00281.

14. Lulla, K. Python-based GPU testing pipelines: Enabling zero-failure production lines. Journal of Information Systems Engineering and Management. 2025;10:978-994.

15. Da Silva, J.; Ansorge, R.; Jena, R. Sub-second pencil beam dose calculation on GPU for adaptive proton therapy. Phys Med Biol. 2015;60(12):4777-4795. doi:10.1088/0031-9155/60/12/4777.

**16.** Fujimoto, R.; Kurihara, T.; Nagamine, Y. GPU-based fast pencil beam algorithm for proton therapy. Phys Med Biol. 2011;56(5):1319-1328. doi:10.1088/0031-9155/56/5/006.

**17.** Schreuder, A.N.; Bridges, D.S.; Rigsby, L.; et al. Validation of the RayStation Monte Carlo dose calculation algorithm using realistic animal tissue phantoms. J Appl Clin Med Phys. 2019;20(10):160-171. doi:10.1002/acm2.12733.

**18.** Paganetti, H.; Botas, P.; Sharp, G.C.; Winey, B. Adaptive proton therapy. Phys Med Biol. 2021;66(22):22TR01. doi:10.1088/1361-6560/ac344f.

**19.** Azcona, J.D.; Aguilar, B.; Perales, Á.; et al. Commissioning of a synchrotron-based proton beam therapy system for use with a Monte Carlo treatment planning system. Radiation Physics and Chemistry. 2023;204:110708. doi:10.1016/j.radphyschem.2022.110708.

**20.** Goma, C.; Almeida, I.P.; Verhaegen, F. Revisiting the single-energy CT calibration for proton therapy treatment planning: a critical look at the stoichiometric method. Phys Med Biol. 2018;63(23):235011. doi:10.1088/1361-6560/aaede5.

**21.** Schneider, U.; Pedroni, E.; Lomax, A. The calibration of CT Hounsfield units for radiotherapy treatment planning. Phys Med Biol. 1996;41(1):111-124. doi:10.1088/0031-9155/41/1/009.