FM216 Project: Rules and Regulations

1. Project Overview

This project requires you to design, implement, and evaluate a novel optimization algorithm for machine learning applications. Working in groups of four students, you will develop a gradient-based optimizer, test it on benchmark problems, write a comprehensive report, and present your findings through a poster presentation.

2. Important Dates and Timeline

Milestone	Date
Project Announcement	October 27, 2025
Report Submission Deadline	November 30, 2025 (11:59 PM)
Poster Presentations	December 1-3, 2025

3. Team Formation and Group Work

3.1 Group Size: Each group consists of 4 (in some cases 3) students.

3.2 Group formation: Groups will be decided by instructor and teaching fellows. You will know your group members on Oct 27th.

3.3 Equal Participation: All group members are expected to contribute equally. The report must include a brief statement describing each member's contributions.

3.4 Collaboration: While collaboration within your group is expected, collaboration between different groups is not allowed. Your optimizer must be your group's original work.

4. Project Requirements

4.1 Novel Optimizer Design

Your optimizer must:
• Be based on gradient descent or related first-order methods
• Include at least one novel mechanism not present in standard optimizers (Adam, SGD, RMSprop, etc.)
• Have clear mathematical formulation with update rules
• Be implementable in Python using NumPy or PyTorch

Examples of novel mechanisms (you can choose others):
• Adaptive momentum based on loss landscape features
• Learning rate scheduling using gradient statistics
• Hybrid approaches combining multiple existing techniques in novel ways
• Gradient preprocessing or transformation methods

4.2 Experimental Evaluation

You MUST test your optimizer on the following four benchmark problems:

1. Well-conditioned Quadratic Function:
   - Form: $f(x) = \frac{1}{2}x^TQx$ where Q has condition number <= 10
   - Report: Number of iterations to reach $f(x) < 1e-6$
   - Dimension: At least 10 variables

2. Ill-conditioned Quadratic Function:
   - Form: $f(x) = \frac{1}{2}x^TQx$ where Q has condition number ≥ 50
   - Report: Number of iterations to reach $f(x) < 1e-6$
   - Dimension: At least 10 variables

3. Rosenbrock Function:
   - Form: $f(x,y) = (1-x)^2 + 100(y-x^2)^2$
   - Report: Number of iterations to reach $f(x,y) < 1e-4$
   - Starting point: (-1.2, 1.0)

4. MNIST Neural Network Training:
   - Architecture: 784-128-64-10 (or similar)
   - Report: Test accuracy after 20 epochs and training time
   - Batch size: 64
   - Compare training loss curves and test accuracy

For each experiment, compare your optimizer against:
• Adam (default parameters: $\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$)
• SGD with Momentum ($\beta=0.9$, $\alpha=0.01$ or tuned)
• At least one other relevant optimizer (RMSprop, AdaGrad, etc.)

5. Report Format and Content

Page Limit: 6-8 pages (excluding references)
Format: PDF, 12pt font, 1-inch margins, single-spaced

Required Sections:

Title Page:
- Optimizer name (creative and descriptive)
- All group member names in alphabetical order by surname
- Course name and semester

Section 1: Introduction and Novel Idea (1.5-2 pages)
- Motivation: What problem does your optimizer address?
- Key Innovation: What is novel about your approach?
- Inspiration: How did you develop this idea?

Section 2: Algorithm Description (2-3 pages)
- Mathematical formulation with clear notation
- Pseudo-code or algorithm box
- Hyperparameters and their default values
- Relationship to existing optimizers (comparison and contrast)
- Theoretical insights (if any)

Section 3: Experimental Results (2-3 pages)
- Detailed results for all three benchmark problems
- Comparison tables and convergence plots
- Statistical analysis (mean ± std over multiple runs)
- Discussion of when your optimizer works well vs. poorly

Section 4: Conclusion (0.5 page)
- Summary of contributions
- Limitations and future work

References: Include all sources cited

6. Poster Presentation Requirements

Format: A0 size (33.1 × 46.8 inches or 841 × 1189 mm), portrait orientation
Content: Must include visual summary of your optimizer, key equations, and main experimental results

Required Elements:
- Title and author names
- Problem statement and motivation
- Algorithm overview (equations + flowchart/diagram)
- Key experimental results (plots and tables)
- Main conclusions

Presentation:
- All group members must be present
- Each member should be able to explain the project
- Duration: 5-minute presentation + 5-minute Q&A per group
- Schedule will be announced by November 25, 2025

7. Grading Rubric (Total: 20% of course grade)

Component	Points	Criteria
Novelty and Innovation	5	Originality of idea, clear innovation beyond existing methods
Algorithm Design	4	Mathematical rigor, clarity of formulation, implementability
Experimental Rigor	5	All benchmarks tested, proper comparisons, statistical analysis
Report Quality	3	Writing clarity, organization, figures/tables quality
Poster Presentation	2	Visual appeal, clarity, effective communication
Oral Presentation	1	Group presentation skills, Q&A handling
Total	20	

8. Submission Guidelines

Report Submission:
• Submit via Moodle by November 30, 2025, 11:59 PM
• File format: PDF only
• File naming: GroupX_OptimizerName_Report.pdf (where X is your group number)
• Include source code as supplementary material (Python .py or .ipynb files)
• Late submissions: -1 point per hour (max 10 hours, after which no submission accepted)

Poster Submission:
• Should be ready before presentation
• Poster will be printed by the university (no cost to students)

Code Submission:
• Submit clean, documented Python code
• Include README with instructions to reproduce experiments
• Use standard libraries (NumPy, PyTorch, Matplotlib)

9. Academic Integrity

• Your optimizer must be your group's original work
• You may use existing implementations of benchmark optimizers (Adam, SGD) for comparison
• Properly cite any external sources, papers, or code snippets used
• Do not share your optimizer code with other groups
• Plagiarism will result in zero credit for all group members and potential disciplinary action

10. Resources

• Course lecture notes on gradient descent, momentum, and adaptive methods
• Reference papers on Adam, RMSprop, and other optimizers (posted on course website)
• Python libraries: NumPy, PyTorch, Matplotlib, scikit-learn
• Office hours: Available for discussing ideas and debugging issues
• Discussion forum: Use for general questions (do not share your optimizer details publicly)

11. Contact Information

For questions regarding the project:
• Instructor office hours: Monday & Wednesday, 1 to 2 pm, HDFC innovation building
• Teaching fellows: Vijay and Ankita,

Good luck with your projects! We look forward to seeing your innovative optimizer designs.