Fourier Option Pricing – Sprint 2 (Implementation Phase)

Week 2–3 Tasks (November 4 – November 17): Implementation

Intro

The repository for the *Carr–Madan FFT Option Pricing* project is now fully set up. This sprint focuses on the actual **implementation phase**, where each member works independently on a specific component of the pricing engine.

To make coordination easier, tasks are designed so that everyone can start **immediately and in parallel**, without depending on others' progress. Each section below specifies:

- what can be done **independently** right away,
- what later needs to be **integrated** once other modules are ready.

The repository structure and base files are already live on the `main` branch. Each member should create their own branch from `main`, work locally, and push commits there.

—

Fundamental Goal (Entire Team)

Implement and validate the **Fourier pricing engine** for the Black–Scholes (BS) model following *Carr & Madan (1999)*. By the end of this sprint, we aim to have:

- a complete pipeline:

$$\text{Characteristic function} \Rightarrow \text{Integral formulation} \Rightarrow \text{FFT pricing} \Rightarrow \text{Validation},$$

- numerical sweeps confirming stability and accuracy for various parameters.

—

Specific Tasks (Members)

Achille Galante – Parameter Sensitivity Study

Independent phase:

- Set up the notebook structure for parameter sweeps.
- Prepare functions to generate grids for α, η, and N.
- Create placeholders for error metrics and timing results.

Integration phase:

- Once Coppetti and Lorello push the integral and FFT pricers, connect your script to those modules.
- Run the sweeps, plot the error vs. parameters, and recommend default stable configurations.

Goal: establish numerical ranges where the FFT is stable and efficient.

—

Bottan – Black–Scholes Characteristic Function Module

Independent phase:

- Implement the log-price characteristic function:

$$\Phi(u) = \exp\Big(\mathrm{i}u(\ln S_0 + (r - \tfrac{1}{2}\sigma^2)T) - \tfrac{1}{2}\sigma^2 u^2 T\Big).$$

- Validate independently with plots of $\Re\Phi(u)$, $\Im\Phi(u)$, and check $\Phi(0) = 1$.

Integration phase:

- Push your validated function to the repository so Coppetti and Lorello can import it.
- Add a docstring and ensure it's vectorized (NumPy-compatible).

Goal: provide a reusable and reliable $\Phi(u)$ callable for the entire project.

—

Coppetti – Carr–Madan Integral (Trapezoidal Implementation)

Independent phase:

- Implement the damped pricing integral using the analytical $\Phi(u)$ from Bottan's formula (you can start even before his push by copying it).
- Test with dummy data to ensure correct numerical integration.

The damped integral form is:

$$C(K) = e^{-\alpha k}\frac{1}{\pi}\int_0^\infty e^{-\mathrm{i}uk}\,\frac{\Phi(u - \mathrm{i}(\alpha+1))}{\alpha^2 + \alpha - u^2 + \mathrm{i}(2\alpha+1)u}\,\mathrm{d}u.$$

Integration phase:

- Once Bottan's official $\Phi(u)$ module is merged, import it directly.
- Compare your results with the analytical Black–Scholes formula and fine-tune the damping α.

Goal: produce a validated numerical version of the continuous Carr–Madan integral.

—

Lorello – FFT Implementation and Visualization

Independent phase:

- Implement the FFT version of the pricing formula using `numpy.fft.fft`.
- Test with a locally defined $\Phi(u)$ (copy Bottan's expression) to verify the transformation works.

Integration phase:

- Replace your local $\Phi(u)$ with Bottan's imported one.
- Compare FFT prices vs. Coppetti's trapezoidal results and vs. the BS closed form.
- Generate final plots: price vs. strike, convergence rate, runtime comparison.

Goal: implement a robust, vectorized FFT pricer and visualize its performance.

—

Davide Piana – Validation and Theoretical Consistency

Independent phase:

- Build a test suite structure in `tests/`.
- Implement functions to check put–call parity, monotonicity, and convexity using synthetic data.

Integration phase:

- Once Coppetti and Lorello push their pricing outputs, feed them into your tests.
- Verify that both the integral and FFT pricers satisfy the theoretical properties.
- Add a short note linking $\Phi(u) \to \Psi(u) \to C(K)$ and mention where *Carr & Madan (1999)* applies Fubini–Tonelli.

Goal: guarantee theoretical and numerical correctness across all modules.

—

Branch & Workflow Guidelines

The repository is ready. Please follow this exact workflow:

```
# 1. Make sure you are on main and up to date
git checkout main
git pull

# 2. Create your personal branch
git checkout -b <your-branch-name>
git push -u origin <your-branch-name>

# 3. Work and commit normally
git add .
git commit -m "feat: implement <task>"
git push

# 4. When finished, open a Pull Request to merge into:
# base: implementation-merge
# compare: <your-branch-name>
```

Each branch is independent. You can start your work right away; once others push their modules, simply pull and integrate as needed.

—

Timeline (2 Weeks Total)

Week	Focus	Members
Week 2 (Nov 4–10)	Core implementation (can start in parallel)	Bottan, Coppetti, Lorello
Week 3 (Nov 11–17)	Validation & numerical analysis (builds on results)	Piana, Galante
Final Milestone	Full Carr–Madan FFT pipeline tested and validated	All members

—

Next Steps

The `main` branch already contains the full folder structure and configuration file. Each member should now create their personal branch from `main` and begin working. Modules can be developed in parallel and finalized once dependent parts are merged. At the end of the sprint, all components will be merged into `implementation-merge` for full testing.