RiskWorkbench — Phase 3
Déploiement d'une interface Qt interactive pour le pricing et les Greeks

Interface Qt, Monte Carlo, Greeks & Stress Testing

September 17, 2025

Contents

1 Objectif de la phase 3

Mettre en place une application Qt complète permettant :

- la saisie des paramètres de marché et d'instrument (vanille européenne),
- le *pricing* MC avec suivi de la *convergence* et comparaison au prix Black–Scholes,
- le calcul des *Greeks* via plusieurs estimateurs (BRV, PW, LRM),
- un *Stress testing* multi-chocs (ΔS, $\Delta\sigma$, Δr, ΔT) avec CRN,
- la *sauvegarde/rechargement* de projets JSON,
- un *harness de tests d'acceptation* (Validation).

2 Architecture et threading

2.1 Vue d'ensemble (MainWindow, QThread, Worker)

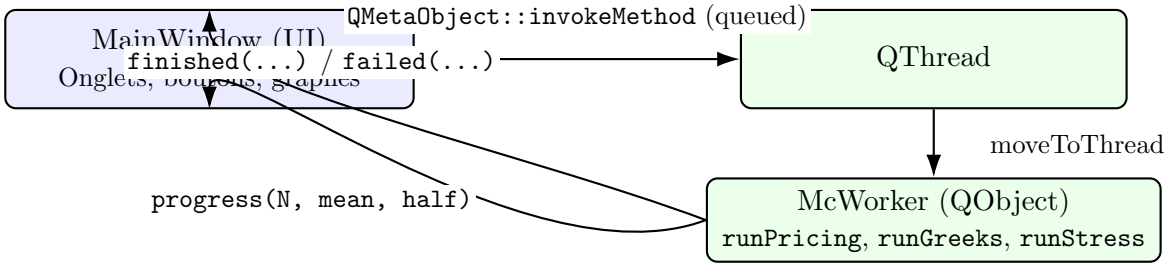


Figure 1: Threading : l'UI ne bloque jamais; le *Worker* tourne dans un `QThread`, les retours se font par signaux.

2.2 Chaîne Validation (tests enchaînés)

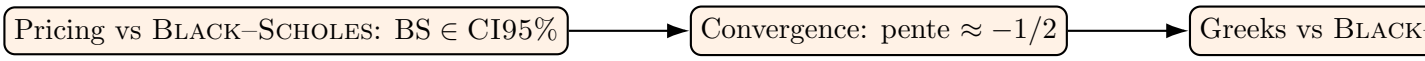


Figure 2: Le bouton *Run all tests* déclenche la séquence; chaque étape connecte/déconnecte proprement ses signaux.

3 Blocs de la phase 3

Bloc 1 — UI Qt, onglets

But de la page Offrir une navigation claire par `QTabWidget` :

1. **Market_Instrument** : S_0, r, q, σ et instrument (K, T, call/put).
2. **Pricing** : bouton *Run MC*, résultats (prix, SE, CI95%, N_{eff}, temps), graphe de convergence.
3. **Greeks** : méthode (BRV/PW/LRM), options CRN/antithétiques, bumps, bar chart.
4. **Stress** : sliders chocs $\Delta S\%$, $\Delta\sigma$ (pts), Δr (bps), ΔT (jours), P&L MC vs Taylor.
5. **Validation** : tableau des tests, log détaillé.

Bloc 2 — Lancer MC sans bloquer l'UI

But de la page Déporter les calculs dans `McWorker` (dans un `QThread`). Publier `progress(N, mean, halfwidth95)` pour alimenter le graphe; conclure via `finished(price, se, ci_low, ci_high, N_eff, elapsed_ms)`.

Bloc 3 — Onglet Pricing

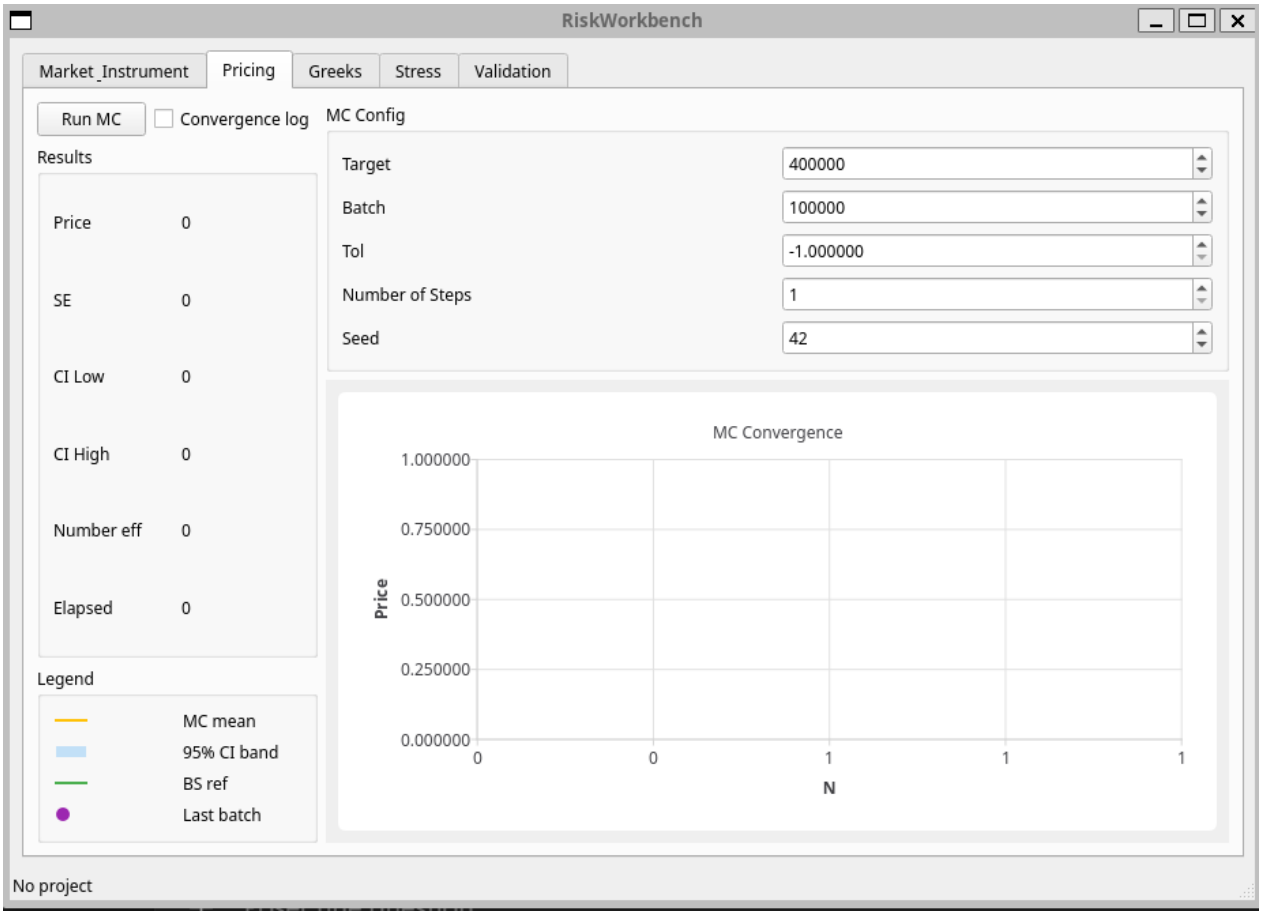
But de la page Calculer le prix MC d'un vanille européen et visualiser la **convergence** vers la valeur Black–Scholes.

Éléments de la page

- **Bouton *Run MC*** : crée un `McConfig` (target, batch, tol, steps, seed), lance `runPricing`.
- **Panneau Résultats** : *Price*, *SE*, *CI Low/High*, *Number eff*, *Elapsed*.
- **Graphe Convergence** : moyenne (orange), bande CI95% (bleu), référence Black–Scholes (vert), dernier point (violet).

Théorie déclenchée Pour un call/put européen sous GBM :

$$\hat{P}_N = \frac{1}{N}\sum_{i=1}^{N} e^{-rT}\Phi(S_T^{(i)}), \qquad \text{SE} = \frac{\hat{\sigma}}{\sqrt{N}}, \qquad \text{CI95\%} = [\hat{P}_N \pm 1.96\,\text{SE}].$$



Capture

Bloc 4 — Onglet Greeks

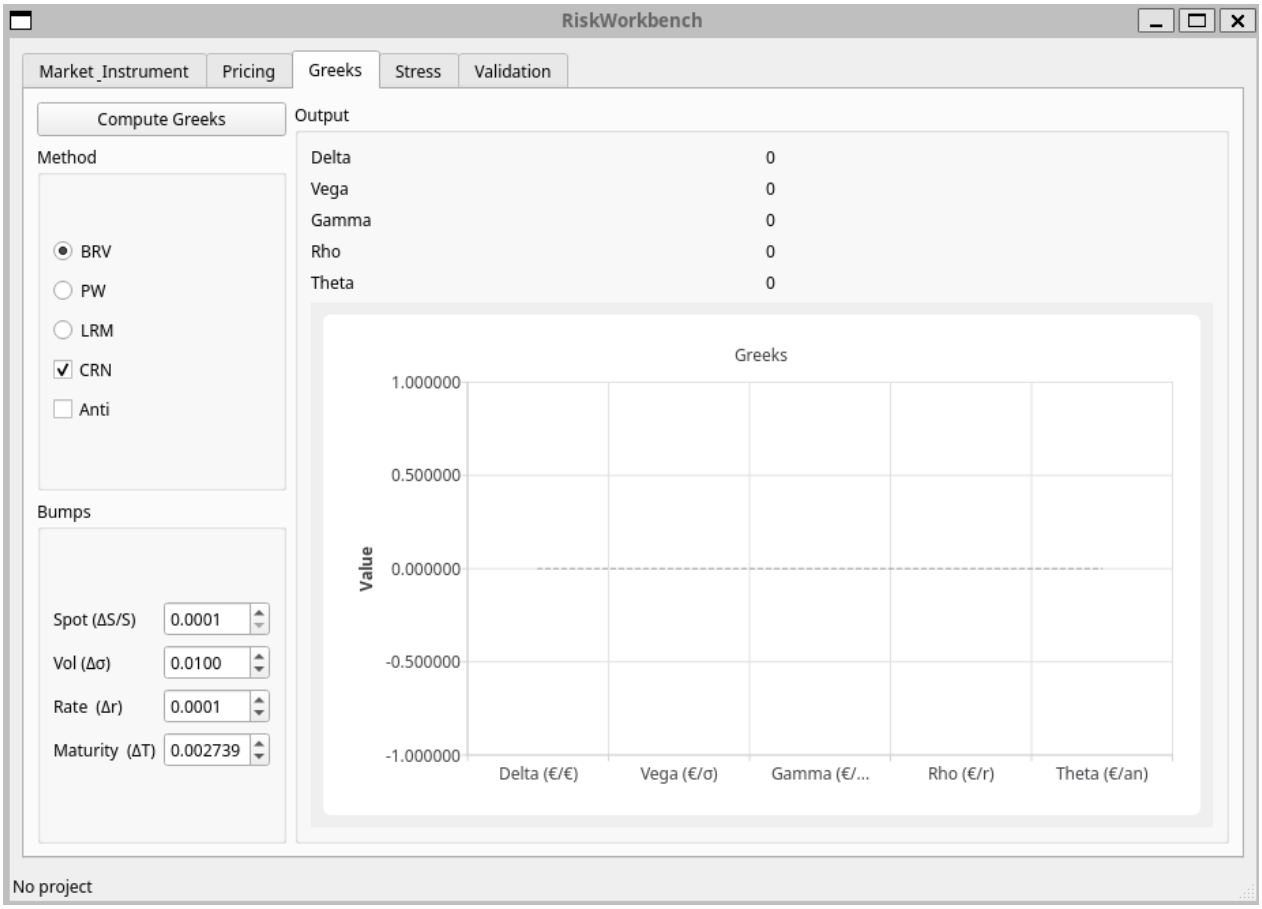
But de la page Estimer Δ, Vega, Γ, ρ, Θ avec un bon compromis biais/variance.

Éléments de la page

- **Méthode** : BRV (bump-and-revalue), PW (pathwise), LRM (likelihood ratio).
- **Options** : CRN, Antithetic variates; *Bumps* ε pour S_0, σ, r, T.
- **Output** : valeurs et bar chart.

Théorie déclenchée

- **BRV** : différences centrées (CRN $\Rightarrow$ variance réduite).
- **PW** : $\Delta = \mathbb{E}[\partial \text{payoff}/\partial S_0]$ si la dérivée existe.
- **LRM** : dérivée de la densité (utile pour Vega sans bump).



Capture

Bloc 5 — Onglet Stress Testing

But de la page Mesurer une **P&L** sous chocs et la comparer à l'approximation **Taylor** d'ordre 2.

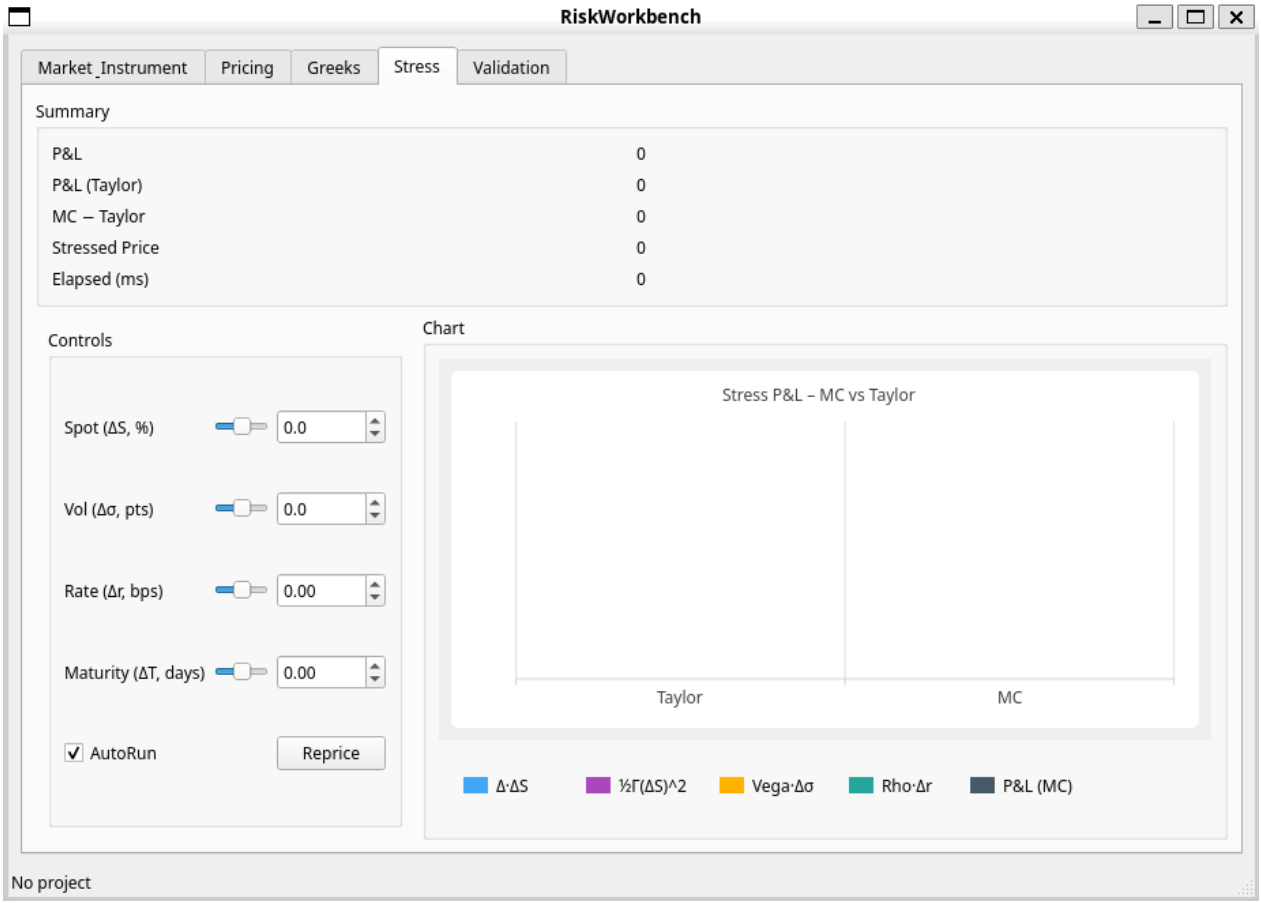
Éléments de la page

- **Sliders** : ΔS (%), $\Delta\sigma$ (pts), Δr (bps), ΔT (jours).
- **AutoRun** : déclenche `runStress` avec *debounce*.
- **Graphe** : pile Taylor $(\Delta, \frac{1}{2}\Gamma(\Delta S)^2, \text{Vega}\cdot\Delta\sigma, \rho\cdot\Delta r)$ et barre MC.

Théorie déclenchée

$$\text{P\&L} \approx \Delta\cdot\Delta S + \tfrac{1}{2}\Gamma(\Delta S)^2 + \text{Vega}\cdot\Delta\sigma + \rho\cdot\Delta r.$$

Avec **CRN**, prix *base* et *stressé* partagent les mêmes tirages $\{Z_k\}$ $\Rightarrow$ variance de P&L fortement réduite.



Capture

Bloc 6 — Sauvegarde / Rechargement

But de la page Rendre les scénarios **reproductibles** et partageables via un JSON lisible.

```
{
  "market": {"S0":100,"r":0.02,"q":0.0,"sigma":0.2},
  "instrument": {"type":"call","K":100,"T":1.0},
  "mc": {"seed":42,"n_paths_target":400000,"batch_size":100000,
         "tolerance":-1,"n_steps":1,"use_antithetic":true,"use_crn":true},
  "ui": {"last_tab":"Pricing"}
}
```

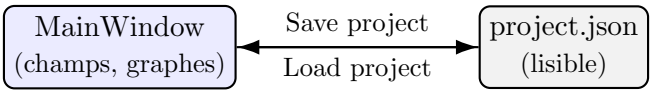


Figure 3: Sauvegarde/reload des paramètres et du dernier onglet visité.

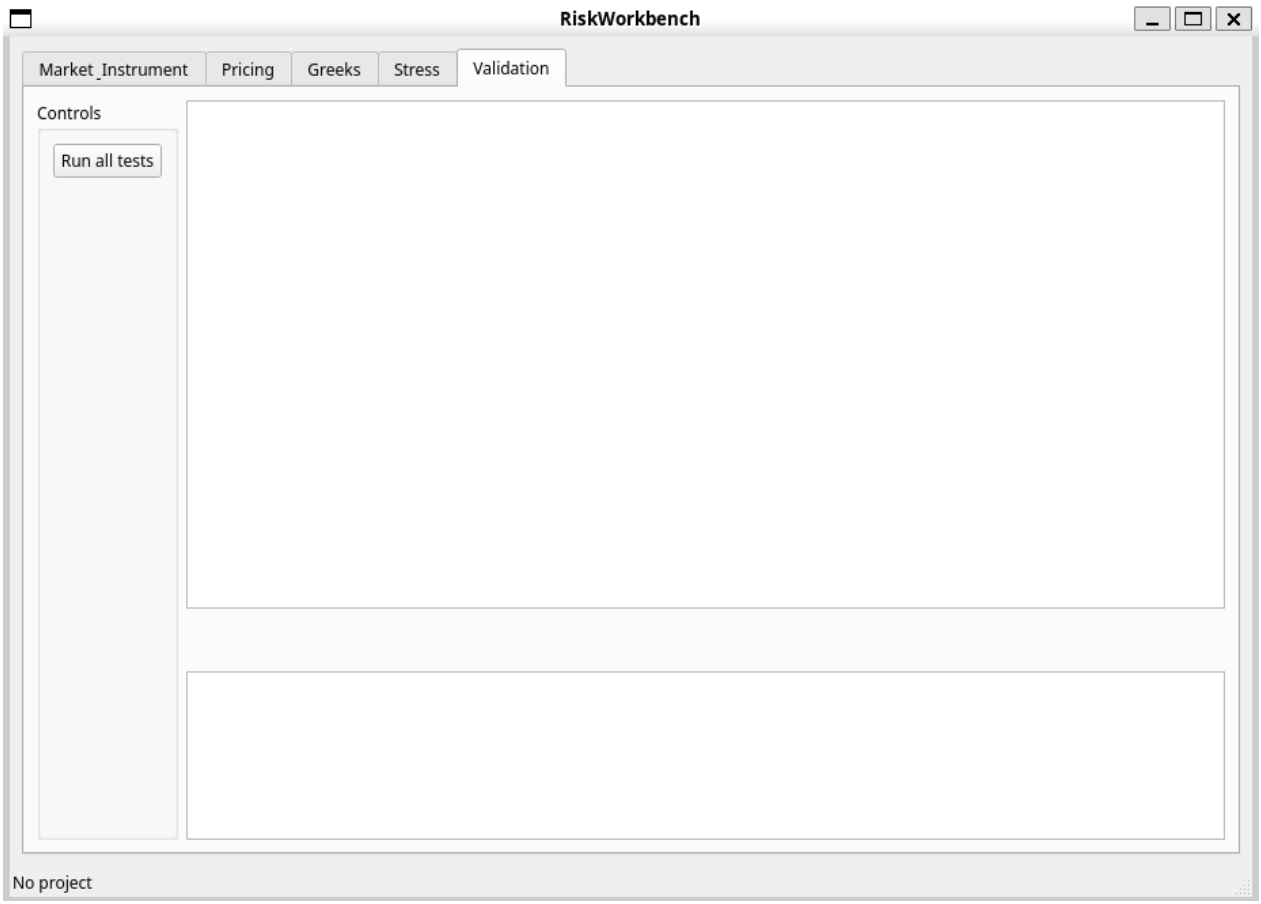
Schéma

Bloc 7 — Tests & Validation

But de la page Proposer un bouton unique qui vérifie toute la chaîne (pricing, pente, greeks, CRN, save/load).

Règles d'acceptation

1. **Pricing vs Black–Scholes** : le Black–Scholes est inclus dans l'intervalle CI95%.
2. **Greeks vs Black–Scholes** : $\max|z| \leq 2$ (moyenne MC vs Black–Scholes, SE par réplicas).
3. **Stress (CRN)** : $\mathrm{Var}(\mathrm{P\&L}) \ll \mathrm{Var}(P)$.
4. **Convergence** : pente $\log(\text{halfwidth})$ vs $\log N$ proche de $-\frac{1}{2}$.
5. **Save/Load** : re-pricing identique ($|\Delta| \leq 10^{-9}$).



Capture

4 Processus détaillés (TikZ)

4.1 Pipeline Pricing MC

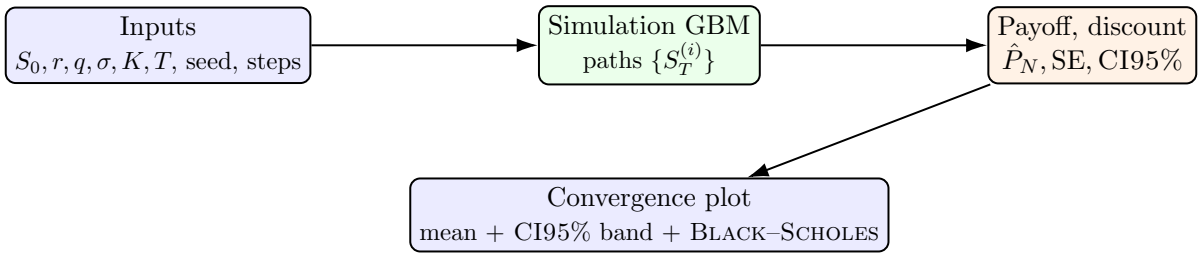


Figure 4: Chaîne de pricing MC.

4.2 Greeks (BRV/PW/LRM)

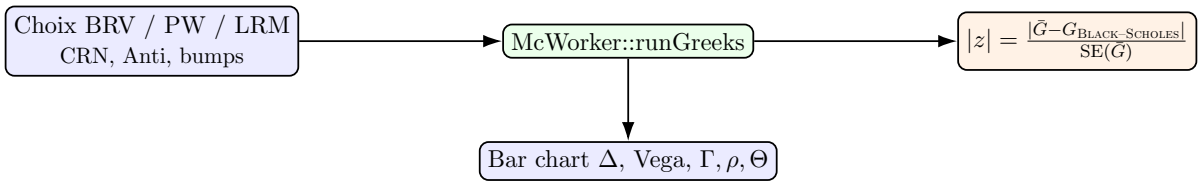


Figure 5: Calcul des greeks et comparaison Black–Scholes.

4.3 Stress avec CRN

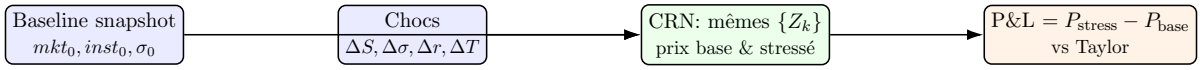
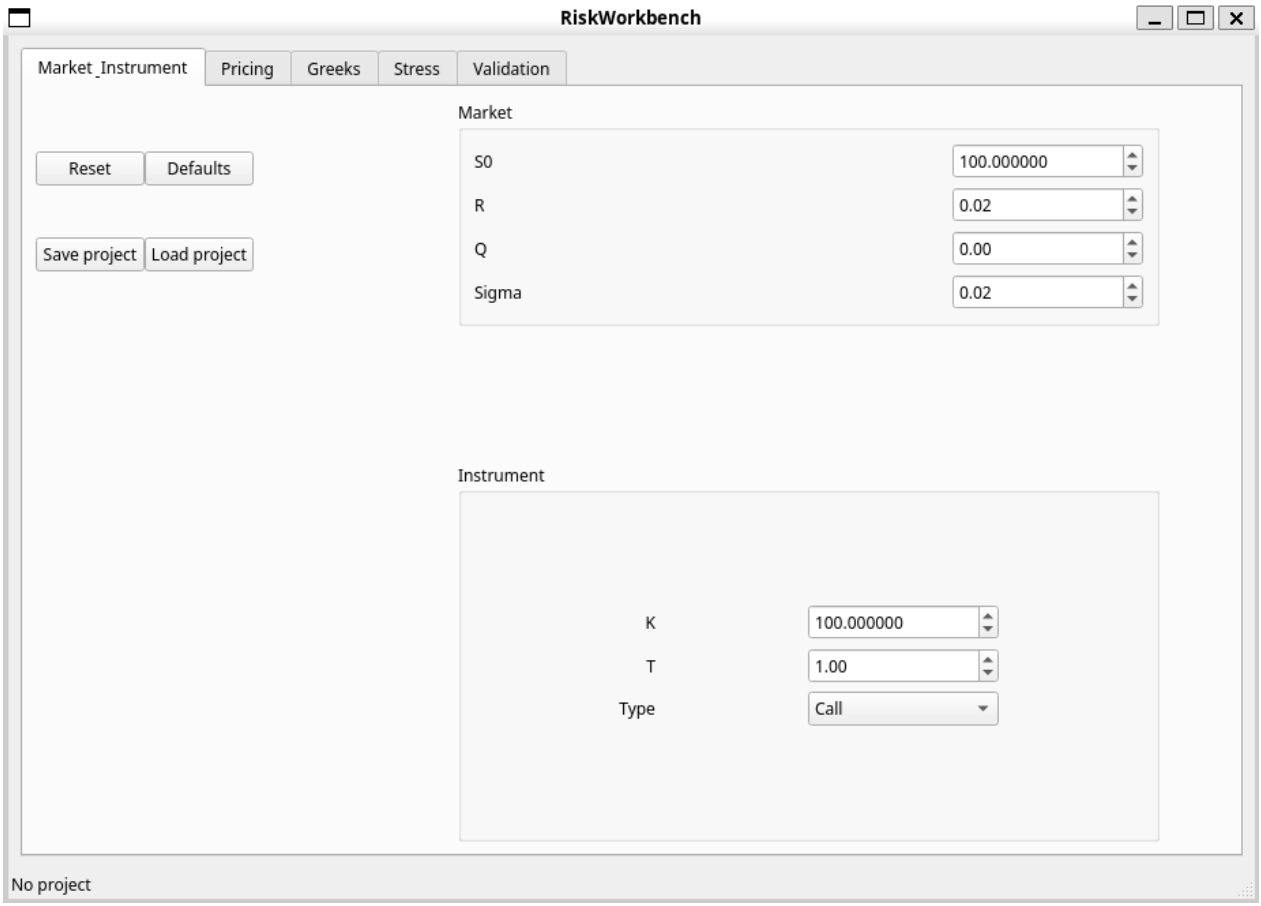


Figure 6: Stress avec Common Random Numbers.

5 Description détaillée des pages (+ théorie derrière les boutons)

5.1 Market_Instrument



But de la page Saisir les paramètres de marché et l'instrument de base.

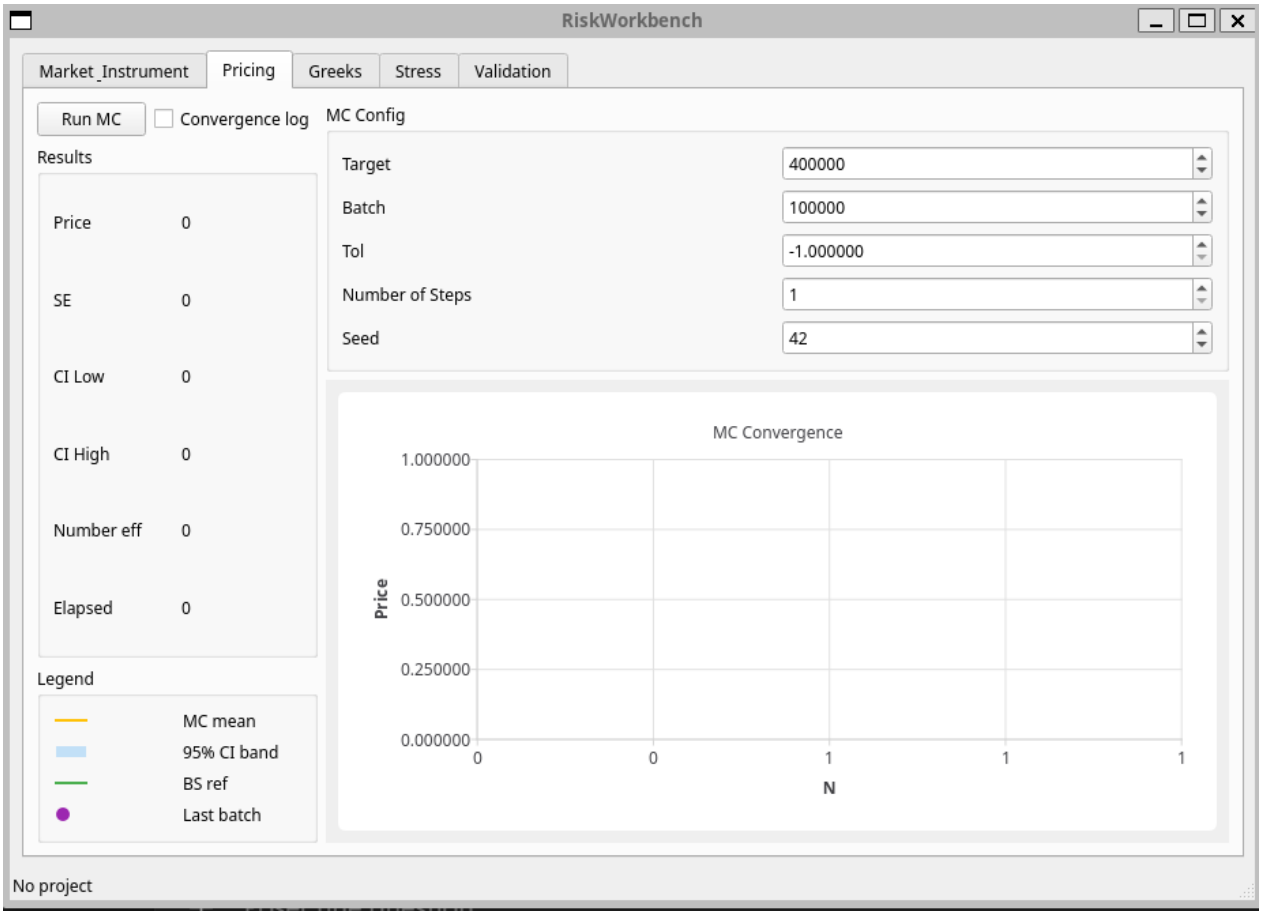
Ce qu'on voit

- Bloc **Market** : S_0 (spot), r (taux), q (div), σ (vol).
- Bloc **Instrument** : K (strike), T (maturité, années), *Type* (Call/Put).
- Boutons **Reset**, **Defaults**, **Save/Load project**.

Théorie/boutons

- **Defaults** charge un set cohérent pour tester la chaîne complète.
- **Save/Load** sérialise la configuration (§3).

5.2 Pricing



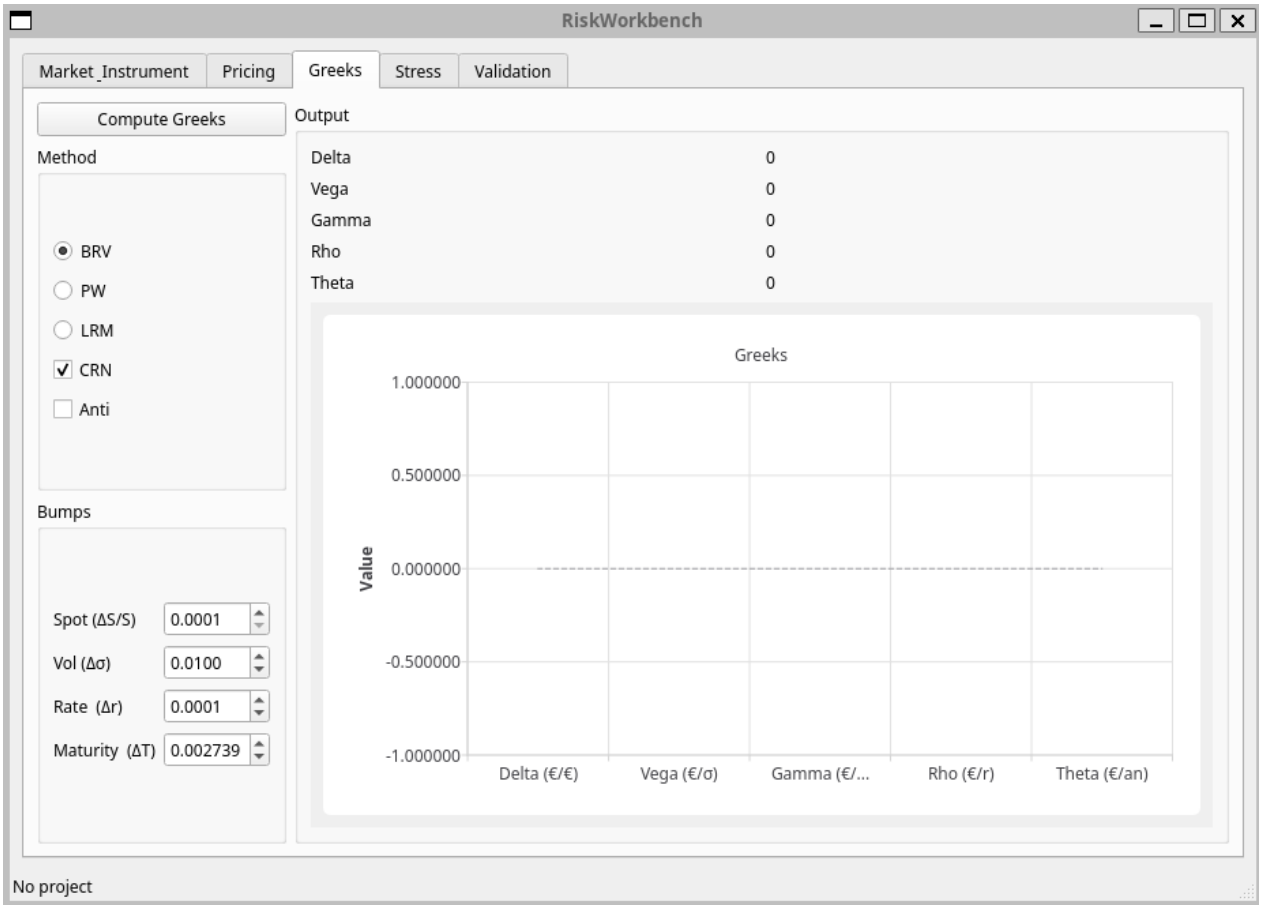
But de la page Lancer un pricing MC, suivre la convergence et comparer au prix Black–Scholes.

Boutons & panneaux

- **Run MC** lance `runPricing`. La moyenne par batch nourrit le graphe en direct (`progress`).
- **MC Config** : *Target*, *Batch*, *Tol* (SE target si > 0), *Steps*, *Seed*.
- **Legend** : code couleur MC/CI95%/Black–Scholes/dernier point.

Théorie Rappel SE $\propto 1/\sqrt{N}$. En Validation, la pente $\approx -1/2$.

5.3 Greeks



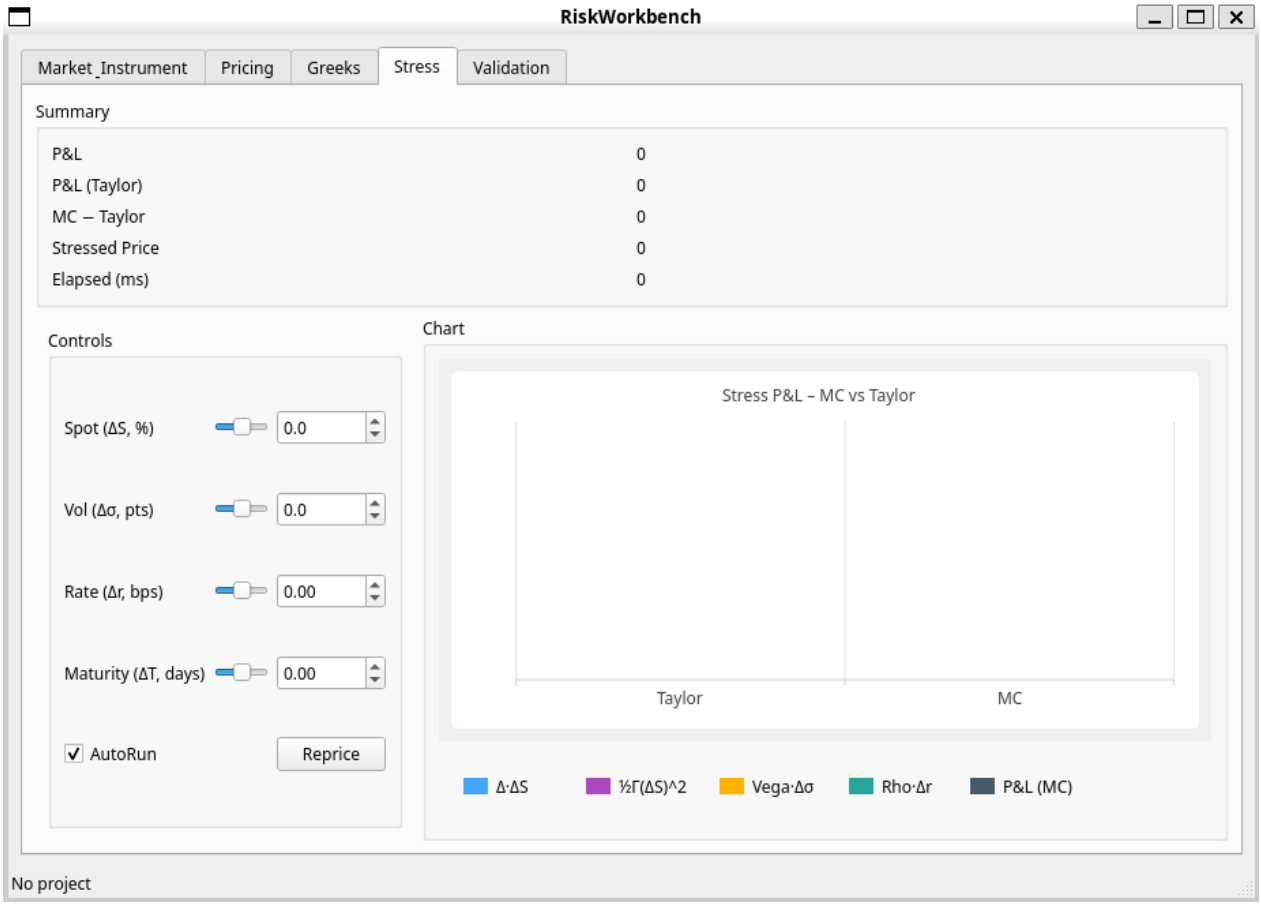
But de la page Afficher les sensibilités principales avec la méthode choisie.

Boutons & options

- **Compute Greeks** : appelle `runGreeks`.
- **BRV/PW/LRM** : trois estimateurs complémentaires.
- **CRN/Anti** : réduction de variance.

Théorie Comparaison aux formules Black–Scholes; z-scores via réplicas indépendants.

5.4 Stress



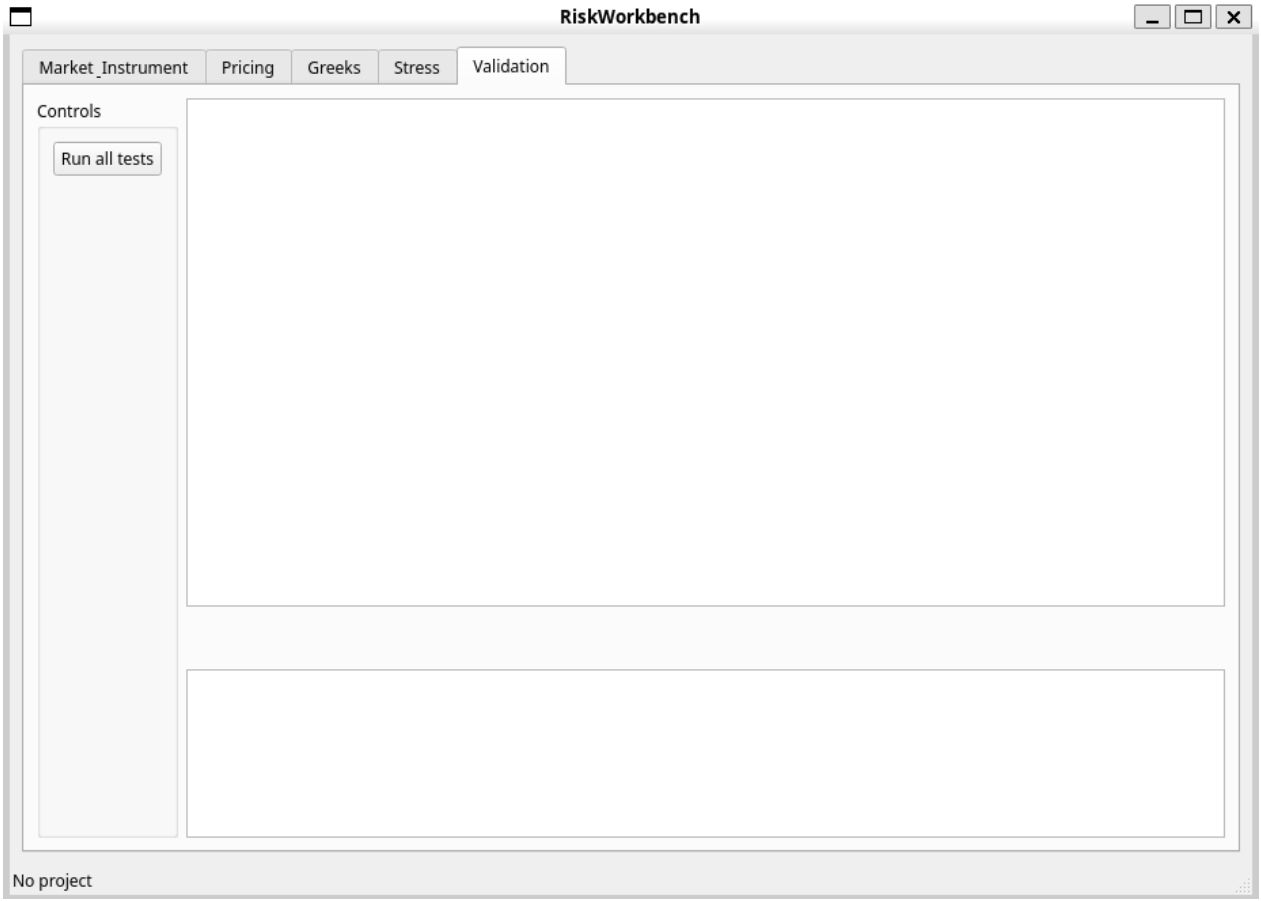
But de la page Visualiser P&L MC sous chocs et sa décomposition de Taylor.

Boutons & graphe

- **Sliders** pilotent les chocs; **AutoRun** applique un *debounce* avant `runStress`.
- **Barres Taylor** : $\Delta \cdot \Delta S$, $\frac{1}{2}\Gamma(\Delta S)^2$, Vega $\cdot\, \Delta\sigma$, $\rho \cdot \Delta r$.
- **Barre MC** : P&L simulée avec CRN.

Théorie Le CRN fait chuter $\mathrm{Var}(P_{\text{stress}} - P_{\text{base}})$ car la corrélation est proche de 1.

5.5 Validation



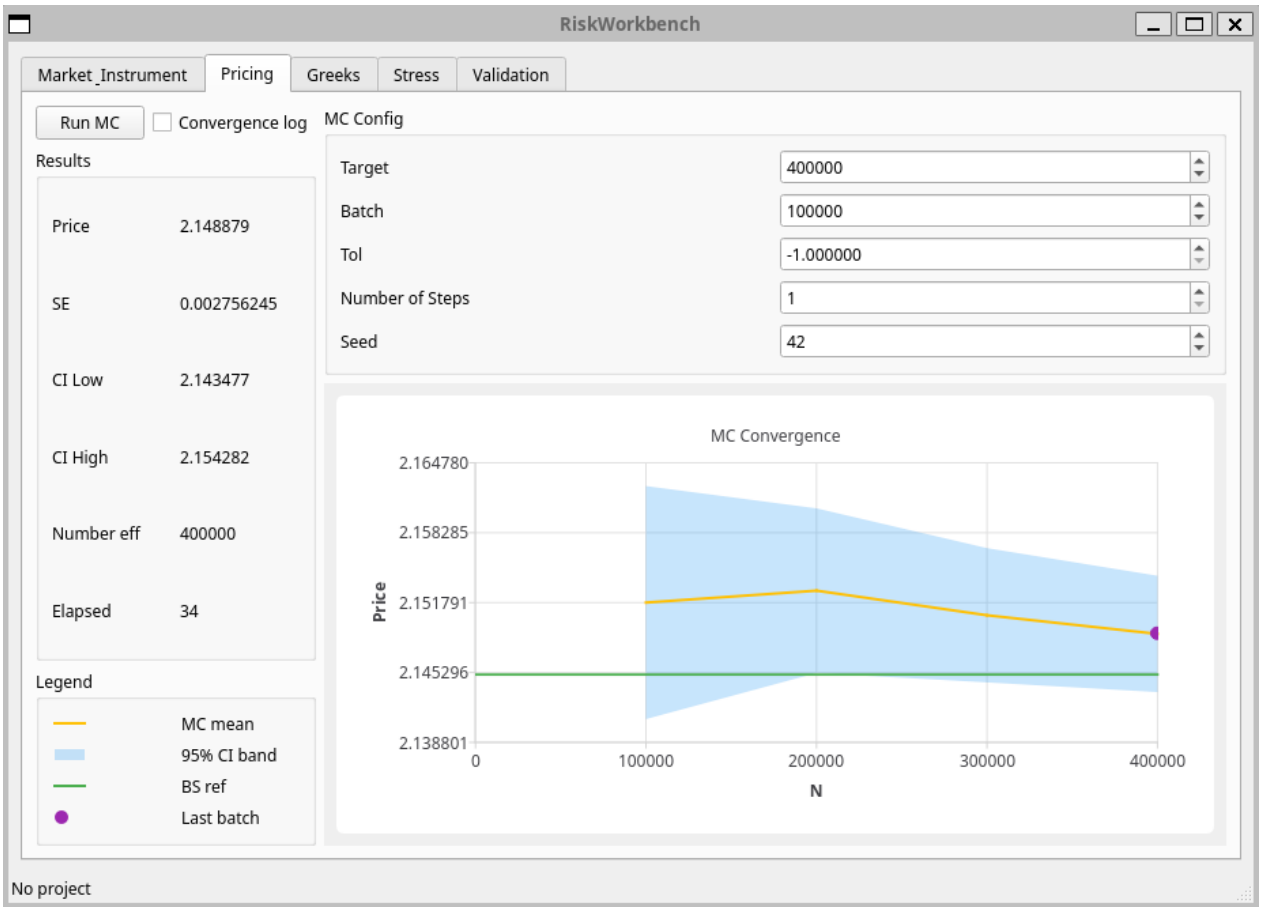
But de la page Fournir un bouton unique *Run all tests* pour valider la chaîne.

Boutons & tableau

- **Run all tests** enchaîne les 5 tests (prix, pente, greeks, CRN, save/load).
- **Table** : Test, métrique, valeur, seuil, verdict; **Log** détaille les nombres.

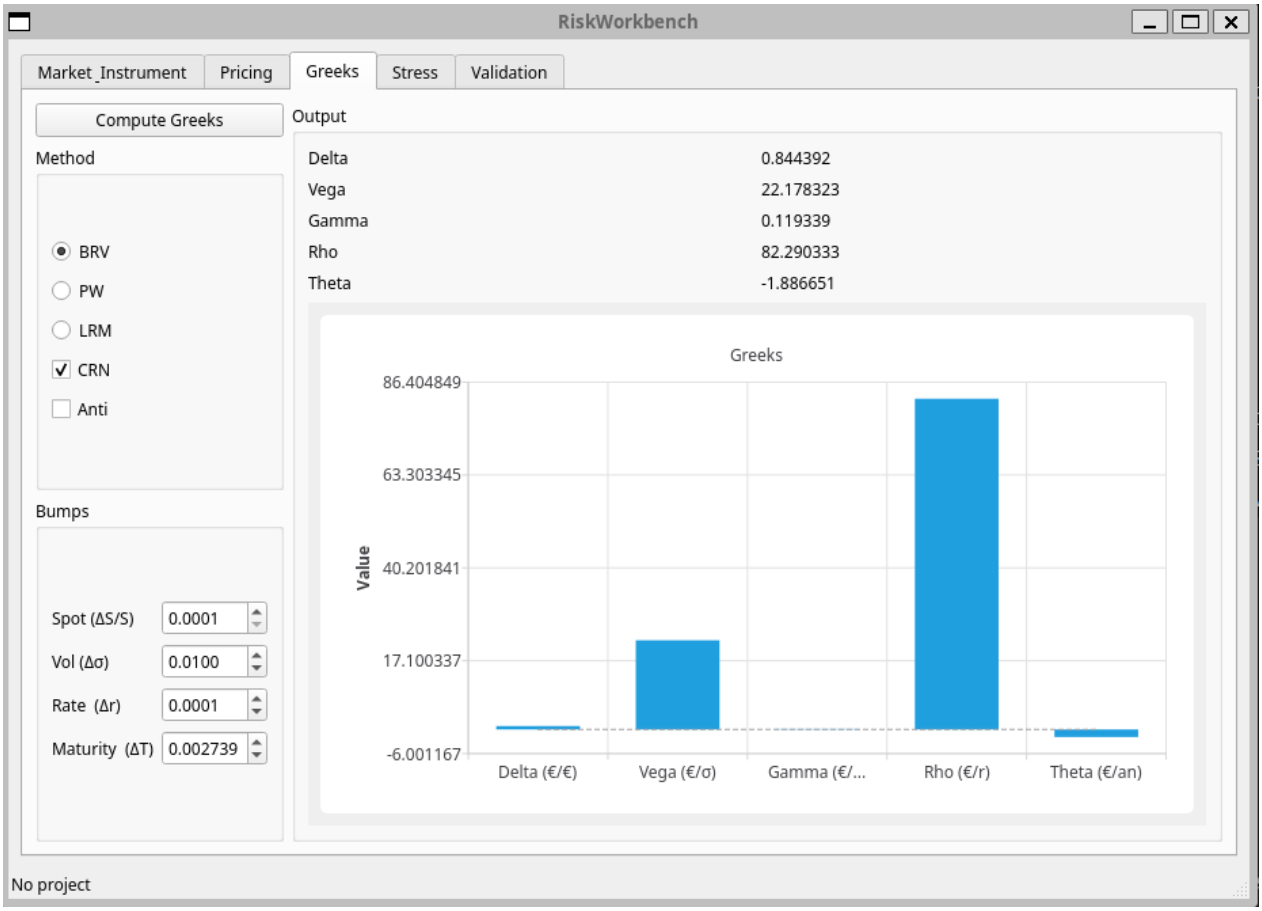
6 Résultats (interprétation des figures)

6.1 Pricing — résultats



On observe un prix MC $\hat{P} \approx 2.149$ avec SE $\simeq 2.76\times10^{-3}$. Le prix Black–Scholes $P_{\text{Black–Scholes}} \approx 2.1451$ est *dans* l'intervalle CI95%, ce qui valide le simulateur et le discount. La bande CI95% se resserre comme $1/\sqrt{N}$.

6.2 Greeks — résultats

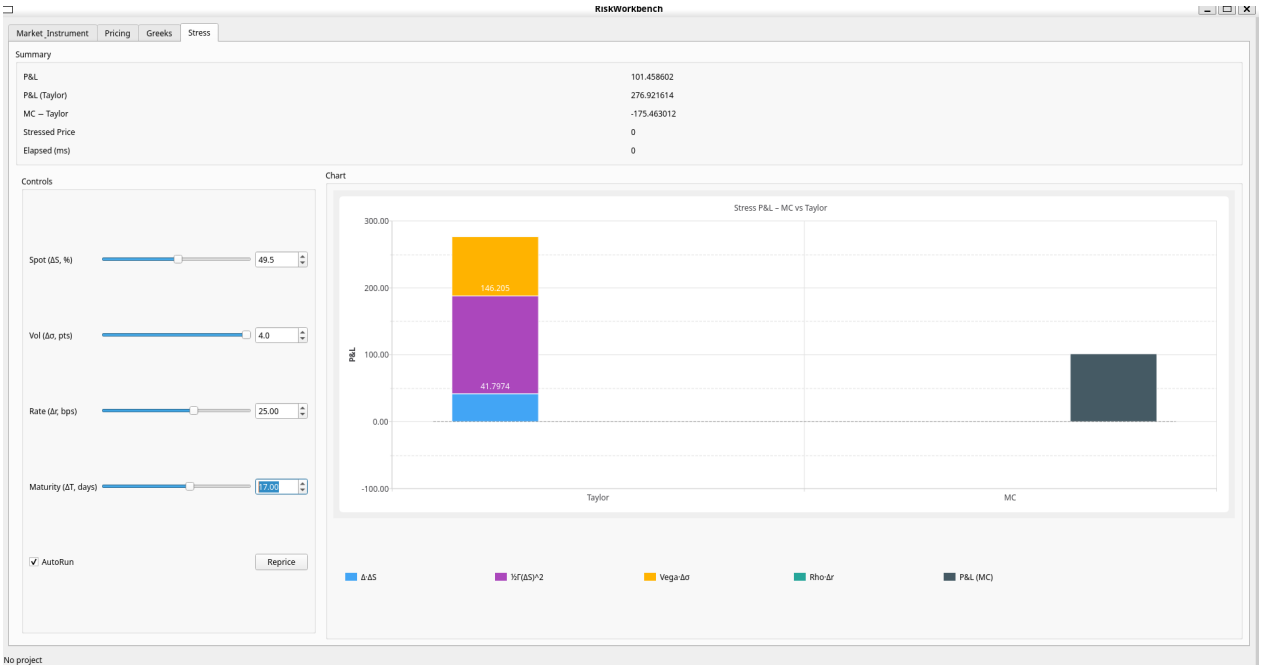


Pour la configuration donnée:

$$(\Delta,\ \text{Vega},\ \Gamma,\ \rho,\ \Theta) \approx (0.844,\ 22.18,\ 0.1199,\ 82.29,\ -1.8866).$$

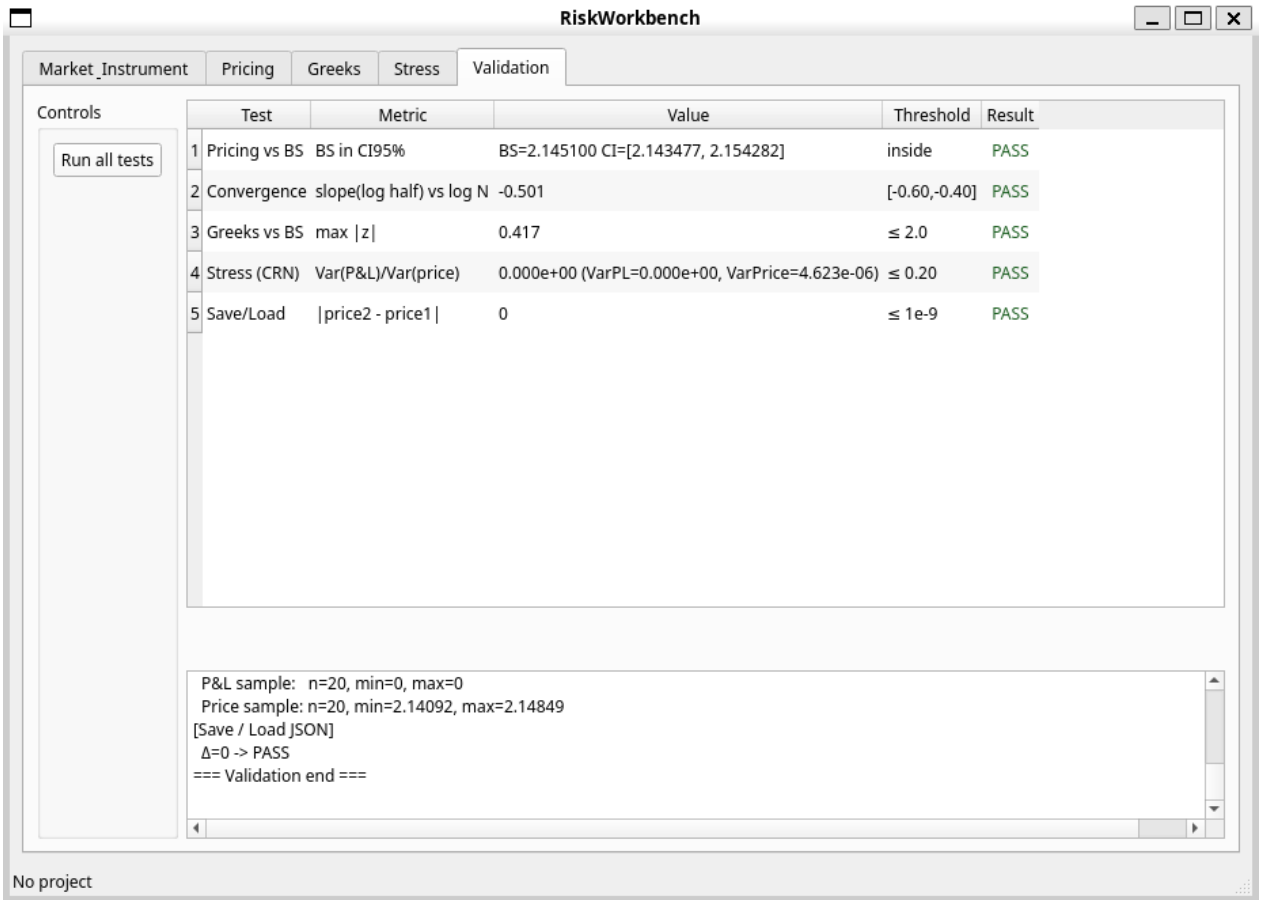
Avec CRN et un bump σ *clampé* (p.ex. 10^{-3}), les z-scores restent modestes; $\max|z| \ll 2$.

6.3 Stress — résultats



La barre MC (grisée) est comparée à la pile Taylor (couleurs). Pour $\Delta S > 0$, la brique $\Delta \cdot \Delta S$ domine; Γ devient visible pour grands $|\Delta S|$. Vega et ρ répondent à $\Delta\sigma$ et Δr. CRN stabilise la P&L.

6.4 Validation — résultats



- **Pricing vs Black–Scholes** : PASS (Black–Scholes dans CI95%).
- **Convergence** : pente $\approx -0.501 \in [-0.60, -0.40]$.
- **Greeks vs Black–Scholes** : $\max |z| \approx 0.42 \leq 2$.
- **Stress (CRN)** : Var(P&L)/Var(P) très faible.
- **Save/Load** : $|P_2 - P_1| = 0$ (à $\leq 10^{-9}$).

7 Problèmes rencontrés & solutions

1. **Segfault dans la variance** (lecture hors borne pour $n < 2$).
   Fix : implémentation Welford, garde $n < 2 \to 0$.
2. **Segfaults intermittents (AddressSanitizer)** dus à signaux encore émis après destruction / doubles connexions.
   Fix : cycle Worker strict (`start/stop`), `Qt::UniqueConnection`, `disconnect()` explicites, état dans `std::shared_ptr`.
3. **Erreur de copie** `McConfig` (operator= supprimé via champs `const`).
   Fix : stocker uniquement les *scalaires* et reconstruire `McConfig` à la volée.
4. **Stress (CRN) FAIL** selon chocs/seed/steps.
   Fix : `onSnapBaseline()`, mêmes RNG pour base/stress, même #steps, cas $T = 0$ géré, chocs par défaut si sliders nuls.
5. **Greeks instables (Vega/Theta)** pour bump σ trop grand.
   Fix : clamp $\varepsilon_\sigma \approx 10^{-3}$, CRN, réplicas indépendants.
6. **Fuites QtCharts** (objets sans parent).
   Fix : parents systématiques pour `QChart`, séries, axes, `QChartView`, ou insertion via layout parenté.
7. **Fuite** `libdbus` **~1KB** à l'arrêt (externe). *Note* : inoffensif; possible suppression via LSAN.
8. **Logs** `QString::arg` **mal formés**.
   Fix : `QString::asprintf` partout pour éviter les erreurs d'index.
9. **Réentrance Validation**.
   Fix : enchaînement strict (disconnect/stop avant test suivant).