

UNIVERSITÉ PARIS DAUPHINE - PSL

Master 272 - Parcours Quant

Note Technique

Backtesting des Stratégies de Trading Algorithmiques



Auteurs : Yassine MANNAI & Issam FRADI

Tuteur : Laurent DAVOUST

Résumé Exécutif

Ce document technique détaille la conception, l'implémentation et les fondements théoriques de **StratlabX**, une application de backtesting financier développée en C# .NET 8.0.

L'objectif du projet est de fournir un environnement robuste pour l'évaluation de stratégies de trading algorithmiques (Pairs Trading, Moyennes Mobiles). Au-delà de la simulation historique, StratlabX intègre des méthodologies avancées issues de la recherche académique, notamment l'optimisation paramétrique par *Grid Search* et la validation de robustesse par simulations de **Monte Carlo**, basées sur les travaux de Joubert, Lopez de Prado et al. (2024).

Sur le plan technique, le projet utilise des concepts clés de la Programmation Orientée Objet (Polymorphisme, Encapsulation), de l'architecture en couches, du multithreading asynchrone et de la conception d'interfaces graphiques modernes (GDI+).

Table des matières

1	Introduction et Philosophie	5
1.1	Contexte	5
1.2	La solution StratlabX	5
2	Architecture Logicielle	6
2.1	Structure des Namespaces	6
2.2	Choix Techniques Clés	6
2.2.1	Le Moteur Événementiel (Event-Driven)	6
2.2.2	Précision Monétaire	6
3	Interface et Expérience Utilisateur (UX)	7
3.1	Philosophie de Design	7
3.2	Architecture de l'Interface	7
3.2.1	Le Menu Principal	7
3.2.2	Composant Personnalisé : Le VolumeSlider	8
3.3	Workflow de Configuration	8
3.4	Immersion Sonore et Interopérabilité	9
4	Algorithmes de Trading	10
4.1	Moving Average Crossover	10
4.1.1	Fonctionnement Mathématique	10
4.1.2	Logique de Signal	10
4.2	Pairs Trading (Arbitrage Statistique)	10
4.2.1	Modélisation du Spread	11
4.2.2	Calcul du Z-Score	11
4.2.3	Règles d'Exécution	11
5	Optimisation par Grid Search	12
5.1	Méthodologie Grid Search	12
5.2	Visualisation	12
6	Validation de Robustesse Monte Carlo	13
6.1	La Décomposition de Cholesky	13
6.2	Fondement Théorique : Types de Backtests	13
6.3	Approche Choisie pour StratlabX	14
6.3.1	Algorithme de Simulation	14
6.4	Interprétation des Résultats	14

7	Visualisation et Métriques	15
7.1	Dashboard Principal	15
7.2	Analyse Avancée	16
8	Conclusion	17
8.1	Acquis Techniques	17
8.2	Perspectives d'Évolution	17

Table des figures

1.1	L'approche narrative : Le personnage de Pierre Chartier	5
3.1	Menu Principal : Interface immersive et contrôle sonore	8
3.2	Séquence de configuration du Backtest	9
5.1	Résultats de l'optimisation : Classement des combinaisons par Sharpe Ratio . .	12
6.1	Histogramme de distribution des Sharpe Ratios (1000 simulations)	14
7.1	Dashboard principal des résultats	15
7.2	Analyse de la distribution des gains	16
7.3	Equity Curve et Drawdown Plot	16

Chapitre 1

Introduction et Philosophie

1.1 Contexte

Le développement de stratégies d'investissement quantitatives repose sur le *Backtesting* : la simulation d'un algorithme sur des données passées. Cependant, les outils existants sont souvent soit trop simplistes, soit trop complexes et austères.

1.2 La solution StratlabX

StratlabX propose une approche hybride innovante :

- **Gamification (UX)** : Une interface "Pixel Art" immersive avec une narration guidée par le personnage "Pierre Chartier", rendant l'analyse quantitative accessible.
- **Rigueur Scientifique (Backend)** : Un moteur événementiel précis, des métriques de risque institutionnelles (Sharpe, VaR) et des tests de robustesse statistique.



FIGURE 1.1 – L'approche narrative : Le personnage de Pierre Chartier

Chapitre 2

Architecture Logicielle

2.1 Structure des Namespaces

- **StratlabX.Core** : Le moteur. Contient le **BacktestRunner** (orchestrateur) et la classe générique **Backtest**.
- **StratlabX.Models** : Objets de données (DTO) comme **PriceBar** (OHLCV) et **Order**.
- **StratlabX.Analytics** : Moteur mathématique. Les classes **Performance** et **Risk** héritent de **Metrics**.
- **StratlabX.Strategies** : Logique métier des algorithmes.
- **StratlabX (App)** : Interface graphique (Windows Forms).

2.2 Choix Techniques Clés

2.2.1 Le Moteur Événementiel (Event-Driven)

Contrairement à un backtest vectoriel qui calcule tout matriciellement, StratlabX itère barre par barre.

```
1 public void Run() {  
2     foreach (var bar in _data) {  
3         _strategy.OnData(bar); // Simulation temps reel  
4         _portfolio.UpdateTotalValue(...);  
5     }  
6 }
```

Cela évite le biais de "Look-Ahead" (regarder le futur) et permet une simulation réaliste du flux de prix.

2.2.2 Précision Monétaire

Le type **decimal** (128-bits) est utilisé partout pour les prix et le cash. Le type **double** est banni pour le stockage de valeur car ses erreurs d'arrondi binaire sont inacceptables en finance.

Chapitre 3

Interface et Expérience Utilisateur (UX)

L'interface de StratlabX a été conçue en rupture avec les standards austères de l'industrie financière (Bloomberg, Excel). L'objectif est de réduire la charge cognitive de l'utilisateur grâce à une approche ludique ("Gamification"), tout en conservant une ergonomie fonctionnelle.

3.1 Philosophie de Design

Le choix d'une esthétique "Pixel Art" et d'une palette de couleurs sombres (*Dark Mode*) répond à deux exigences :

- **Immersion** : Plonger l'utilisateur dans l'ambiance d'une salle de marché nocturne, rappelant l'esthétique "Minecraft" ou les terminaux de trading rétro.
- **Confort visuel** : Minimiser la fatigue oculaire lors de longues sessions d'analyse de données.

La navigation est guidée par un narrateur, "Pierre Chartier", qui contextualise chaque étape de la configuration, transformant un processus technique fastidieux en une quête interactive.

3.2 Architecture de l'Interface

L'application utilise le framework Windows Forms. Cependant, pour éviter l'aspect "gris" standard de Windows, la quasi-totalité des composants natifs a été stylisée ou remplacée.

3.2.1 Le Menu Principal

Le menu (Fig. 3.1) agit comme le "Hub" central. Il gère l'état global de l'application et l'accès aux sous-modules (Backtest, À propos, Quitter) via une gestion événementielle asynchrone pour garantir la fluidité.



FIGURE 3.1 – Menu Principal : Interface immersive et contrôle sonore

3.2.2 Composant Personnalisé : Le VolumeSlider

Pour s'intégrer au design sombre, les contrôles natifs de Windows (TrackBar) étaient inadaptés graphiquement. Nous avons donc développé un UserControl personnalisé.

Techniquement, ce composant repose sur la surcharge de la méthode de peinture du système (GDI+). Au lieu de laisser Windows dessiner le composant, le code redessine vectoriellement, pixel par pixel :

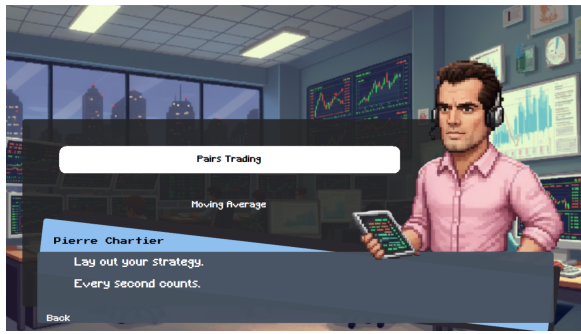
- La barre de fond (Track) avec des extrémités arrondies.
- Le curseur (Thumb) avec un effet de lueur externe pour simuler un néon.
- L'activation de l'anti-aliasing pour garantir des courbes lisses.

Cette approche démontre une maîtrise de la bibliothèque graphique système `System.Drawing` et de la gestion bas niveau des événements souris.

3.3 Workflow de Configuration

La configuration d'un backtest est séquentielle, empêchant l'utilisateur de lancer une simulation avec des paramètres incohérents ou manquants.

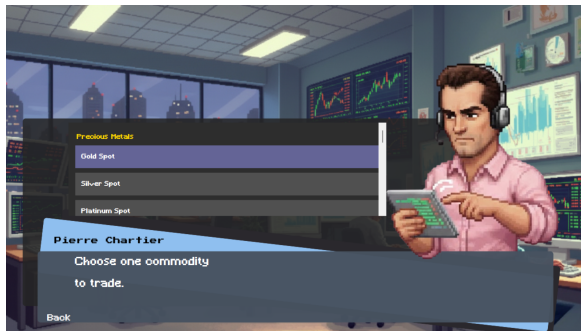
1. **Sélection de la Stratégie** : Le système charge dynamiquement les classes disponibles implémentant la classe abstraite `Strategy`.
2. **Univers d'Investissement** : Choix entre Actions, Indices ou Commodities (Fig. 3.2b).
3. **Sélection des Actifs** : Les données sont chargées depuis des fichiers Excel et affichées dans un `FlowLayoutPanel` permettant une sélection multiple fluide (Fig. 3.2c).
4. **Horizon Temporel** : Définition de la profondeur historique (3, 5 ou 10 ans) qui déterminera la taille des échantillons de données (Fig. 3.2d).



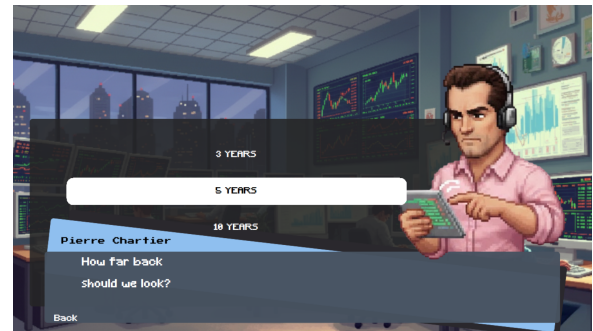
(a) Choix Stratégie



(b) Choix Marché



(c) Sélection Actif



(d) Horizon Temporel

FIGURE 3.2 – Séquence de configuration du Backtest

3.4 Immersion Sonore et Interopérabilité

Pour renforcer l'immersion, StratlabX intègre un moteur audio capable de jouer une boucle musicale et des effets sonores (clics, survols).

Plutôt que d'utiliser des lecteurs lourds (comme Windows Media Player), la gestion audio est réalisée par Interopérabilité (P/Invoke). L'application communique directement avec l'API native de Windows (`winmm.dll`) pour envoyer des commandes MCI (Media Control Interface). Cela permet une gestion légère et performante du son, sans dépendance externe majeure.

Chapitre 4

Algorithmes de Trading

Ce chapitre détaille la logique mathématique implémentée dans `StratlabX.Strategies`.

4.1 Moving Average Crossover

Cette stratégie de "Suivi de Tendances" (Trend Following) cherche à capturer les grands mouvements de marché.

4.1.1 Fonctionnement Mathématique

On calcule deux Moyennes Mobiles Simples (SMA) à l'instant t :

$$SMA_t(N) = \frac{1}{N} \sum_{i=0}^{N-1} P_{t-i}$$

Où P est le prix de clôture et N la période (ex : 10 et 20 jours).

4.1.2 Logique de Signal

- **Signal d'Achat (Golden Cross)** : Si $SMA_{court} > SMA_{long}$ alors que la veille $SMA_{court} \leq SMA_{long}$.
- **Signal de Vente (Death Cross)** : Si $SMA_{court} < SMA_{long}$.

L'implémentation utilise une file (Queue) de taille fixe pour optimiser le calcul des moyennes sans recalculer toute l'historique à chaque pas de temps.

4.2 Pairs Trading (Arbitrage Statistique)

Cette stratégie est plus sophistiquée. Elle exploite la **co-intégration** entre deux actifs (ex : Or et Argent). L'idée est que si les prix s'écartent anormalement, ils finiront par converger (Retour à la moyenne).

4.2.1 Modélisation du Spread

On définit le ratio de prix entre l'actif A et l'actif B :

$$Ratio_t = \frac{P_{A,t}}{P_{B,t}}$$

4.2.2 Calcul du Z-Score

Pour déterminer si le ratio est "anormal", on le normalise statistiquement en calculant son Z-Score sur une fenêtre glissante (Lookback Period) :

$$Z_t = \frac{Ratio_t - \mu_{rolling}}{\sigma_{rolling}}$$

Où $\mu_{rolling}$ est la moyenne du ratio sur les N derniers jours et $\sigma_{rolling}$ son écart-type.

4.2.3 Règles d'Exécution

La stratégie utilise des seuils statistiques (ex : ± 2 écarts-types) :

- **Entrée Short** ($Z_t > 2.0$) : L'actif A est cher par rapport à B. On vend A et on achète B.
- **Entrée Long** ($Z_t < -2.0$) : L'actif A est bon marché. On achète A et on vend B.
- **Sortie** ($|Z_t| < 0.5$) : Retour à l'équilibre, on liquide les positions pour prendre le profit.

Chapitre 5

Optimisation par Grid Search

Comment savoir si une moyenne mobile de 20 jours est meilleure qu'une de 50 ? Ou si le seuil d'entrée du Pairs Trading doit être 2.0 ou 2.5 ? C'est le rôle du module d'optimisation.

5.1 Méthodologie Grid Search

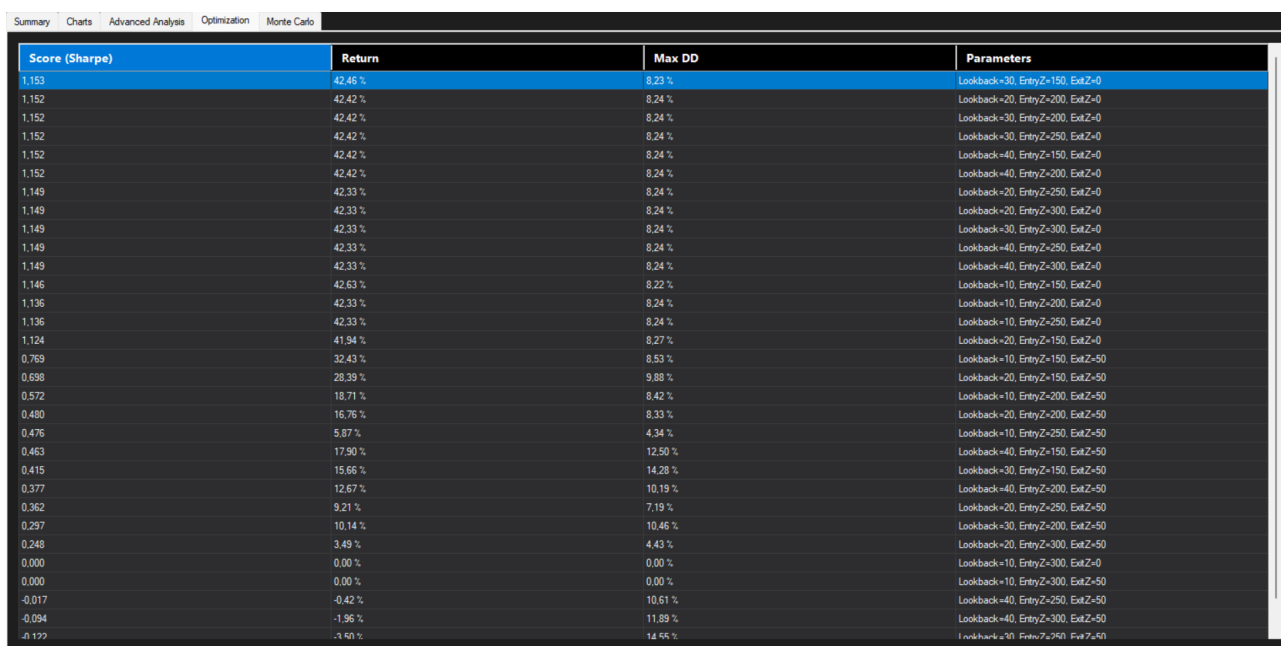
L'optimiseur (`Optimizer.cs`) effectue une recherche exhaustive sur une grille de paramètres définie. Pour le Pairs Trading, l'espace de recherche est tridimensionnel :

1. **Lookback Period** : {10, 20, 30, 40}
2. **Entry Threshold** : {1.5, 2.0, 2.5, 3.0}
3. **Exit Threshold** : {0.0, 0.5}

Cela génère $4 \times 4 \times 2 = 32$ backtests indépendants.

5.2 Visualisation

Les résultats sont présentés dans un `DataGridView` triable (Fig. 5.1), permettant d'identifier immédiatement la combinaison maximisant le Ratio de Sharpe.



Score (Sharpe)	Return	Max DD	Parameters
1.153	42.46 %	8.23 %	Lookback=30, EntryZ=150, ExitZ=0
1.152	42.42 %	8.24 %	Lookback=20, EntryZ=200, ExitZ=0
1.152	42.42 %	8.24 %	Lookback=30, EntryZ=200, ExitZ=0
1.152	42.42 %	8.24 %	Lookback=30, EntryZ=250, ExitZ=0
1.152	42.42 %	8.24 %	Lookback=40, EntryZ=150, ExitZ=0
1.152	42.42 %	8.24 %	Lookback=40, EntryZ=200, ExitZ=0
1.149	42.33 %	8.24 %	Lookback=20, EntryZ=250, ExitZ=0
1.149	42.33 %	8.24 %	Lookback=20, EntryZ=300, ExitZ=0
1.149	42.33 %	8.24 %	Lookback=30, EntryZ=300, ExitZ=0
1.149	42.33 %	8.24 %	Lookback=40, EntryZ=250, ExitZ=0
1.149	42.33 %	8.24 %	Lookback=40, EntryZ=300, ExitZ=0
1.146	42.63 %	8.22 %	Lookback=10, EntryZ=150, ExitZ=0
1.136	42.33 %	8.24 %	Lookback=10, EntryZ=200, ExitZ=0
1.136	42.33 %	8.24 %	Lookback=10, EntryZ=250, ExitZ=0
1.124	41.94 %	8.27 %	Lookback=20, EntryZ=150, ExitZ=0
0.769	32.43 %	8.53 %	Lookback=10, EntryZ=150, ExitZ=50
0.698	28.39 %	9.88 %	Lookback=20, EntryZ=150, ExitZ=50
0.572	18.71 %	8.42 %	Lookback=10, EntryZ=200, ExitZ=50
0.480	16.76 %	8.33 %	Lookback=20, EntryZ=200, ExitZ=50
0.476	5.87 %	4.34 %	Lookback=10, EntryZ=250, ExitZ=50
0.463	17.90 %	12.50 %	Lookback=40, EntryZ=150, ExitZ=50
0.415	15.66 %	14.28 %	Lookback=30, EntryZ=150, ExitZ=50
0.377	12.67 %	10.19 %	Lookback=40, EntryZ=200, ExitZ=50
0.362	9.21 %	7.19 %	Lookback=20, EntryZ=250, ExitZ=50
0.297	10.14 %	10.46 %	Lookback=30, EntryZ=200, ExitZ=50
0.248	3.49 %	4.43 %	Lookback=20, EntryZ=300, ExitZ=50
0.000	0.00 %	0.00 %	Lookback=10, EntryZ=300, ExitZ=0
0.000	0.00 %	0.00 %	Lookback=10, EntryZ=300, ExitZ=50
-0.017	-0.42 %	10.61 %	Lookback=40, EntryZ=250, ExitZ=50
-0.094	-1.96 %	11.89 %	Lookback=40, EntryZ=300, ExitZ=50
-0.122	-3.40 %	14.56 %	Lookback=10, EntryZ=250, ExitZ=50

FIGURE 5.1 – Résultats de l'optimisation : Classement des combinaisons par Sharpe Ratio

Chapitre 6

Validation de Robustesse Monte Carlo

L'optimisation paramétrique comporte un risque majeur : l'**Overfitting** (sur-apprentissage). Une stratégie peut sembler excellente simplement parce qu'elle a été "tunée" pour coller parfaitement au bruit du passé.

Pour contrer cela, StratlabX implémente une validation basée sur le papier de recherche "*The Three Types of Backtests*" (Joubert, Sestovic, Lopez de Prado, 2024).

6.1 La Décomposition de Cholesky

Lors de simulations Monte Carlo impliquant plusieurs actifs corrélés (comme dans le Pairs Trading), il est crucial de préserver la structure de corrélation entre les actifs. C'est ici qu'intervient la Décomposition de Cholesky.

La décomposition de Cholesky permet de décomposer une matrice de covariance Σ (symétrique et définie positive) en un produit d'une matrice triangulaire inférieure L et de sa transposée L^T :

$$\Sigma = LL^T$$

Cette propriété est utilisée pour générer des bruits aléatoires corrélés ϵ_{corr} à partir de bruits aléatoires indépendants ϵ_{indep} (gaussiens standard) :

$$\epsilon_{corr} = L \times \epsilon_{indep}$$

Dans StratlabX, bien que la validation actuelle utilise principalement le ré-échantillonnage (Resampling) des rendements historiques, la compréhension de cette méthode est essentielle pour l'extension vers des générateurs de scénarios purement synthétiques.

6.2 Fondement Théorique : Types de Backtests

Le papier de référence identifie trois types de backtests :

1. **Walk-Forward (Historique)** : Test sur une seule trajectoire passée.
2. **Resampling (Ré-échantillonnage)** : Techniques comme le Bootstrap ou la validation croisée.
3. **Monte Carlo** : Génération de données synthétiques.

6.3 Approche Choisie pour StratlabX

StratlabX implémente une méthode hybride de **Ré-échantillonnage combinatoire (Combinatorial Resampling)**. L'objectif est de vérifier si le Ratio de Sharpe obtenu est un résultat robuste ou une anomalie statistique.

6.3.1 Algorithme de Simulation

La classe `MonteCarloValidator` exécute l'algorithme suivant $N = 1000$ fois :

- **Extraction** : On récupère la série des rendements journaliers (*Returns*) de la stratégie.
- **Shuffling (Mélange)** : On mélange aléatoirement l'ordre de ces rendements. Cela brise la dépendance temporelle tout en conservant les propriétés statistiques (moyenne, écart-type).
- **Reconstruction** : On génère une nouvelle Equity Curve fictive.
- **Mesure** : On calcule le Ratio de Sharpe de cette courbe fictive.

6.4 Interprétation des Résultats

Nous obtenons une distribution de 1000 Ratios de Sharpe possibles (Fig. 6.1). Si le Sharpe original de notre stratégie est nettement supérieur à la moyenne des simulations et se situe dans l'extrême droite de la distribution, alors la stratégie exploite une véritable inefficience de marché.

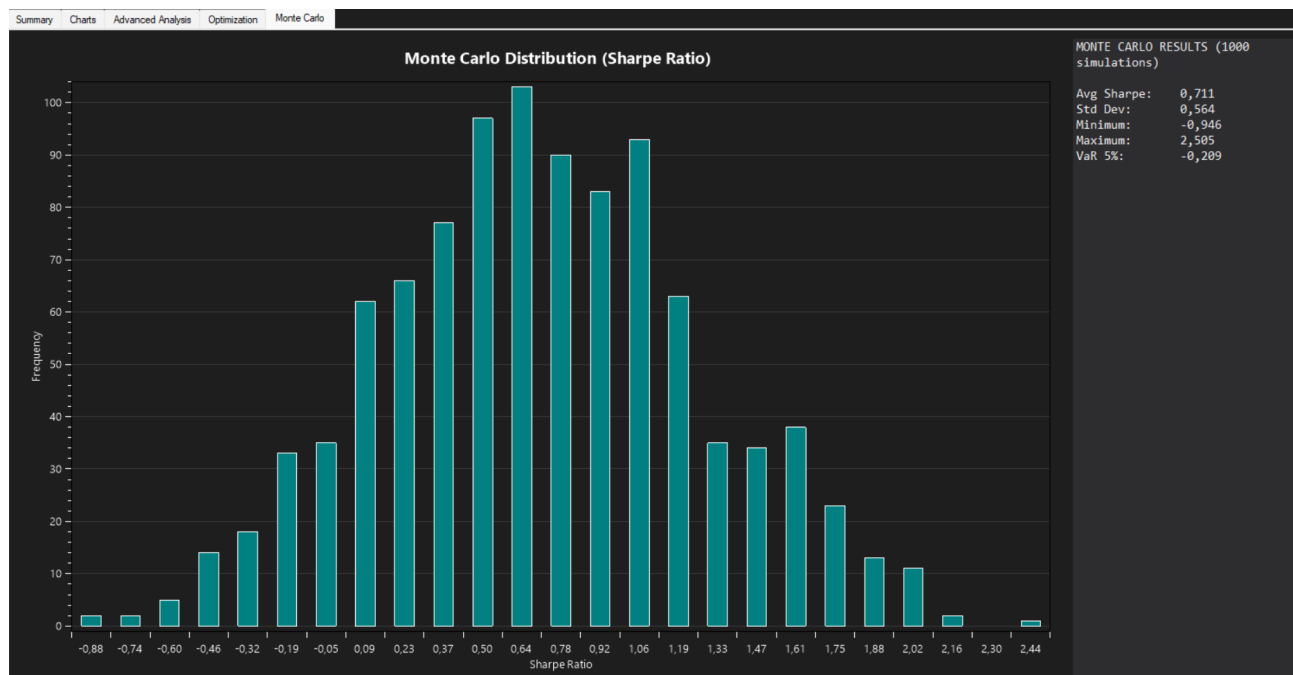


FIGURE 6.1 – Histogramme de distribution des Sharpe Ratios (1000 simulations)

Chapitre 7

Visualisation et Métriques

StratlabX utilise la librairie **OxyPlot** pour offrir des visualisations professionnelles.

7.1 Dashboard Principal

Le résumé (Fig. 7.1) présente l'Equity Curve (évolution de la richesse) et les métriques clés :

- **Sharpe Ratio** : Rendement ajusté du risque.
- **Sortino Ratio** : Rendement ajusté du risque baissier.
- **Max Drawdown** : Perte maximale historique du sommet au creux.

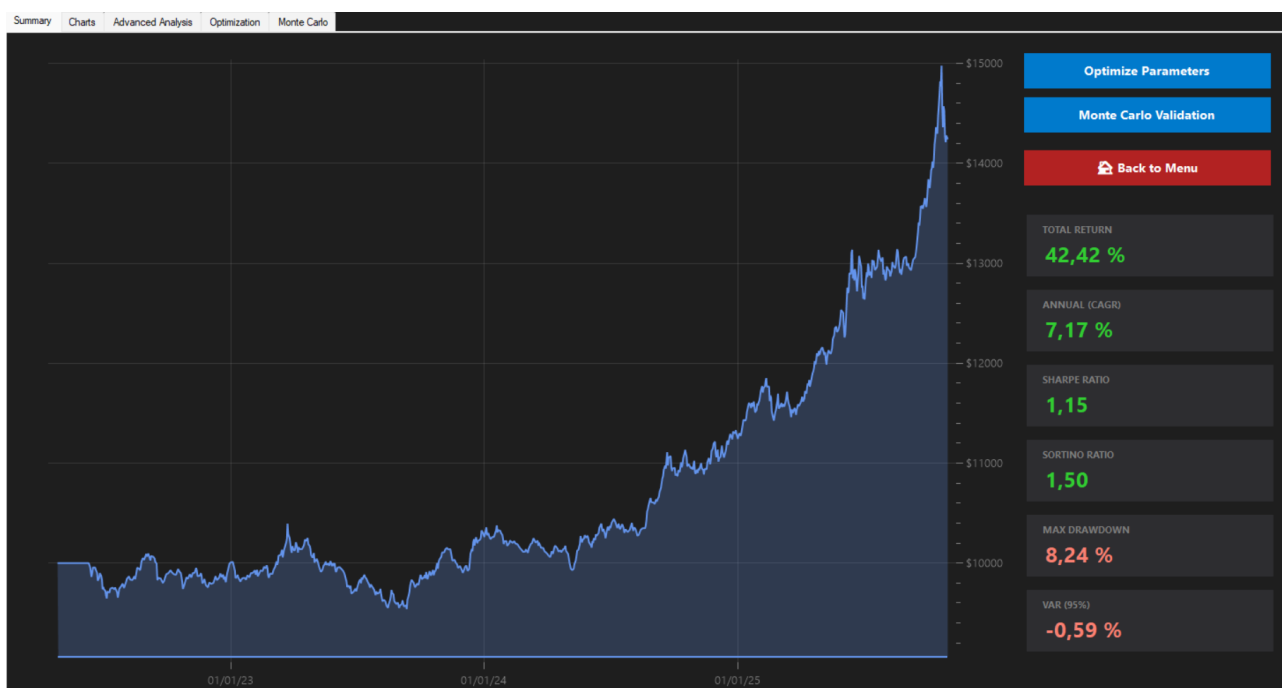


FIGURE 7.1 – Dashboard principal des résultats

7.2 Analyse Avancée

Pour comprendre la dynamique de la stratégie, des graphiques dédiés sont proposés :

- **Pie Chart Dynamique** (Fig. 7.2) : Ratio Jours Gagnants vs Jours Perdants.
- **Bar Chart Annuel** : Rendements par année civile.

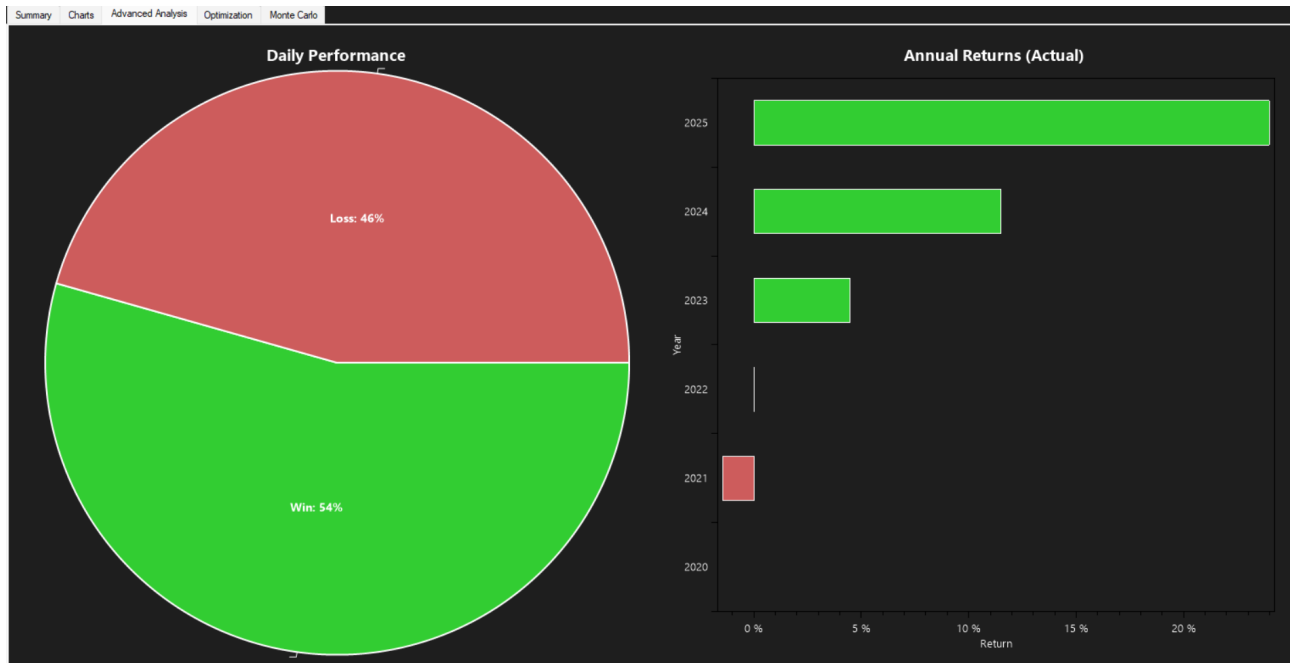


FIGURE 7.2 – Analyse de la distribution des gains

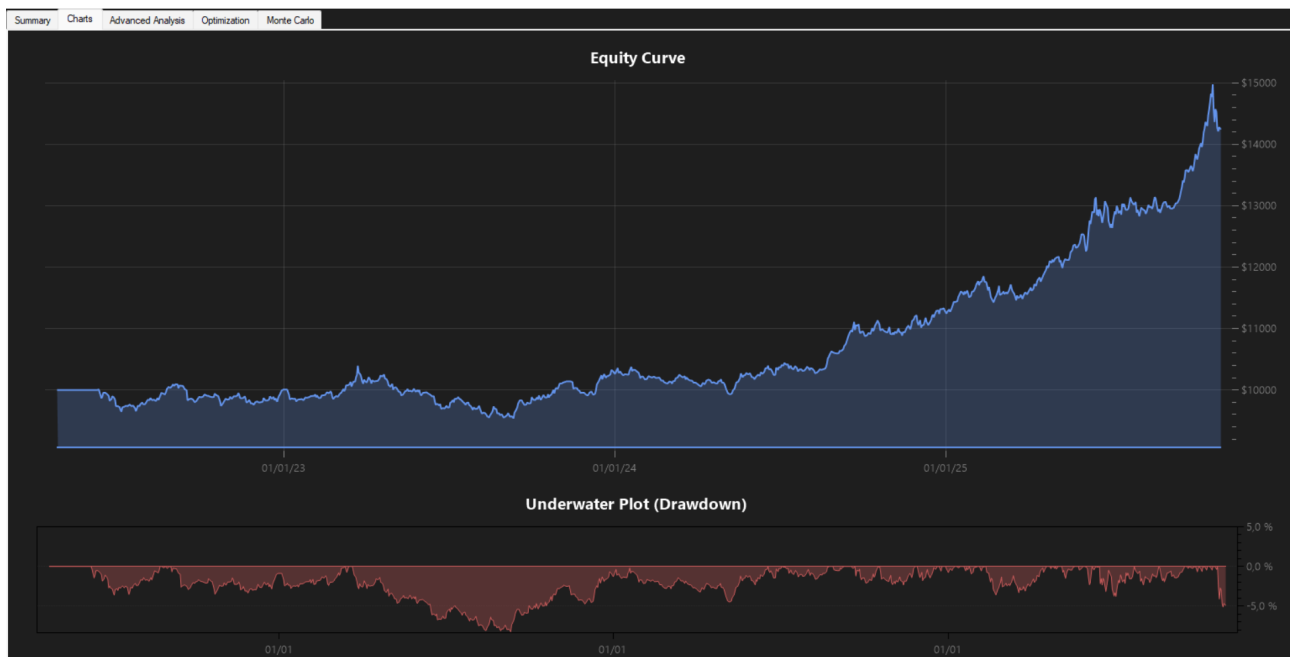


FIGURE 7.3 – Equity Curve et Drawdown Plot

Chapitre 8

Conclusion

StratlabX démontre qu'il est possible de concilier complexité financière et expérience utilisateur fluide au sein d'une application Windows Forms.

8.1 Acquis Techniques

Ce projet a permis de consolider plusieurs compétences clés :

- **Architecture N-Tiers** : Séparation propre UI / Logique / Données.
- **Algorithmique Financière** : Implémentation fidèle de modèles mathématiques (Z-Score, VaR, Sharpe, Cholesky).
- **Programmation Asynchrone** : Gestion des threads pour l'optimisation.
- **Data Visualization** : Intégration poussée d'OxyPlot pour des rendus professionnels.

8.2 Perspectives d'Évolution

Pour transformer ce prototype académique en outil de production, les développements futurs pourraient inclure :

- Intégration d'une base de données SQLite pour remplacer les fichiers Excel.
- Ajout des coûts de transaction (Slippage et Commissions) pour plus de réalisme.
- Connexion à une API de marché pour des données en temps réel.

Bibliographie

- [1] Jacques Joubert, Dragan Sestovic, Illya Barziy, Walter Distaso, Marcos Lopez de Prado. *The Three Types of Backtests*. Abu Dhabi Investment Authority, 2024.
- [2] Marcos Lopez de Prado. *Advances in Financial Machine Learning*. John Wiley & Sons, 2018.
- [3] William F. Sharpe. *The Sharpe Ratio*. The Journal of Portfolio Management, 1994.
- [4] Ernie Chan. *Quantitative Trading : How to Build Your Own Algorithmic Trading Business*. John Wiley & Sons, 2009.
- [5] Raja Velu, Maxence Hardy, Daniel Nehren. *Algorithmic Trading and Quantitative Strategies*. CRC Press, 2020.