# MATH GR 5320
# Financial Risk Management and Regulation

## Lecture 5: Model Risk Management

Department of Mathematics
Columbia University

Harvey J. Stein

Head, Quantitative Risk Analytics
Bloomberg LP

Fall 2016

Compilation: September 30, 2016 at 16:03

If errors are found, please return them to hjstein@columbia.edu.

# Outline

# Review

# Review

Market risk – Impact of market moves.

VaR computation complications:

- Intermediate events.
- Formulas become impossible.
- Model selection and parameter estimation:
  - Mean hard to estimate.
  - Variance is easier.
  - Weighted averaging is useful.
  - Use GBM for nonnegative risk factors.
  - Use ABM for everything else.
  - Maybe factor models.

# Review

Types of VaR:

- Parametric – Formulas for approximations.
- Historical – Monte Carlo with historical perturbations. applied
- Monte Carlo – Fit models to risk factors and calculate VaR by simulation.

Mean and variance of sum of lognormals:

$$
\begin{aligned}
V_t &= aS_{1,t} + bS_{2,t} \\
dS_i &= \mu_i S_i dt + \sigma_i S_i dW_i \\
dW_1 dW_2 &= \rho dt \\
E[S_{1,t}S_{2,t}] &= S_{1,0}S_{2,0}e^{(\mu_1+\mu_2+\rho\sigma_1\sigma_2)t} \\
E[V_t] &= aS_{1,0}e^{\mu_1 t} + bS_{2,0}e^{\mu_2 t} \\
E[V_t^2] &= a^2 S_{1,0}^2 e^{(2\mu_1+\sigma_1^2)t} + b^2 S_{2,0}^2 e^{(2\mu_2+\sigma_2^2)t} \\
&\quad + 2abS_{1,0}S_{2,0}e^{(\mu_1+\mu_2+\rho\sigma_1\sigma_2)t} \\
\mathrm{var}[V_t] &= E[V_t^2] - E[V_t]^2
\end{aligned}
$$

# Introduction

# Introduction

Post-crisis, regulations regarding model risk management have been greatly expanded [OCC11; Bas13].

Pre-crisis:

- Focus on model validation.

Post crisis:

- Focus on model risk management.

Model risk management requirements (as per [OCC11]):

- Effective governance.
- Robust model development, implementation and use.
- Sound model validation practices.

Post crisis, model risk management is the fastest growing area in finance.

Additional references: [OCC14], [FED11b], [FED09], [Bas13], [FED11a].

# Governance, policies and controls

# Governance

Governance, policies and controls:

- Large part of regulations.
- Ensuring structure of firm and policies in place promote sound model risk management.
- Avoid conflict of interests and moral hazards.
- To promote a firm-wide "safety first" mindset.

Requirements:

- Independent risk control unit.
- Strong board and senior management support and respect for risk management.
- Regular auditing of risk management and general firm practices.

# Governance

Board and senior management:

- Must be actively involved in the risk control process.
- Must regard risk control as essential.
- Must devote significant resources to it.
- Must insure that risk reports are reviewed at a level that can enforce risk reductions.
- Must review risk reports on a monthly basis.

Policies:

- Firm-wide risk appetite statement.
- Risk management practices and policies must be documented, sound, and adhered to.
- All aspects of model risk management must be covered.
- Testing and accuracy must be emphasized.

# Model development

# Model development

Model risk management during the model development phase breaks down into two components:

1. Pre-deployment analysis, documentation and testing.
2. Post-deployment follow-up analysis.

# Pre-deployment

Pre-deployment risk management = software development best practices:

1. A clear statement of purpose for the model (i.e. a requirements document).
2. Design documentation.
3. Data analysis.
4. Testing.
5. System analysis and testing.

# Requirements document

Requirements document:

- The purpose of the model.
- The scope of use of the model.
- Performance requirements of the model.

# Design document

The design document justifies the choice of model, including:

- Justification of model choice by reference to published research and/or sound documented industry practices.
- Explanation of the mathematical specification and numerical techniques in detail.
- Analysis of the assumptions, merits and limitations, including the extent to which risks are captured.
- Comparison with alternative approaches.
- Documentation and validation of subjective components (such as fine tuning based on expert knowledge).

# Data validation

Data is a critical component.

- Problematic data = questionable results.

Data analysis includes:

- Documentation of the data being used.
- Rigorous assessment of the data quality.
- Demonstration that the data is suitable for the model and consistent with the theory and methodology.
- Identification, justification and documentation of all proxies used.
- Documentation and analysis of all assumptions made when adjusting data (data cleaning, data smoothing, data averaging, etc).

# Implementation

Implementation best practices are expected:

- Software design documentation.
- Frequent code reviews.
- Enforcement of written coding standards.
- Testing.

# Software design documentation

Design documentation:

- Detailed description of software architecture.
- Module documentation:
  - Purpose.
  - Assumptions.
  - Interfaces.
  - Data structures.
- Data flow documentation:
  - Operation of system at the module level.
- Document components comprising each module.

# Code review

Code review

- Important component of validation.

Look for:

- Bugs.
- Design weaknesses.
- Hidden assumptions.
- Common coding errors (e.g. equality comparison between floating point numbers).
- Unit tests.
- Appropriately chosen algorithms.
- Each module/function/routine has a simple, well defined purpose.
- Separation of data gathering from calibration from computation.

# Testing

Testing is expected to be fairly comprehensive and professionally handled.

Test preparation:

- Documentation of purpose, design and execution of test plans.
- Summary of results with commentary and evaluation.
- Detailed analysis of informative samples.

# Testing

Component testing:

- Analysis of components to show that they are appropriate for the intended purpose, conceptually sound and mathematically and statistically correct.
- Testing of accuracy of model (e.g. pricing is accurate or risks are accurately reflected).

# Testing

Robustness testing:

- Testing the stability and robustness of the model (sensitivity to changes in market conditions and to deal specifications).
- Assessment of the limitations of the model (parameter ranges, assumptions, etc).
- Assessing behavior over a large range of inputs.
- Assessing impact of assumptions.
- Identifying situations where the model performs poorly or becomes unreliable.
- Testing under actual circumstances and with a variety of market conditions, extreme values, etc.
- Assessing impact of model results that feed other models.
- Application of a variety of tests (statistical, accuracy, etc).

# System analysis

Model risk management of the entire system is required:

- Documentation of the system.
- Analysis of how the system ensures data and reporting integrity.
- Controls and testing to ensure proper implementation, effective integration and appropriate use.

# Independent review

Each stage requires review:

- Review by independent teams not vested in the model release.
  - The model design committee
  - The model validation team
  - Senior management
  - Board of directors
- Entire process is subject to audit.
- Implementation does not begin until data selection, the model, and the numerical methods have been approved.
- If model doesn't pass, shortcomings must be addressed and model must be revised.

# Post-deployment

Once a model is deployed, feedback must be obtained regarding the performance of the model. The development team must:

- Elicit and incorporate feedback from model users.
- Elicit and incorporate feedback from independent sources.
- Confirm that reports produced are clear and comprehensive.
- Reports produced should indicate and adjust for accuracy and uncertainty in results, either by output of ranges of results, or by making results more conservative.
- Confirm that users of the models understand the model limitations, uncertainty and inaccuracy.

# Change management

Good change management is also required:

- Changes must be justified, logged and tested.
- Testing and revalidation, based on the magnitude of the change.
- Small implementation changes: unit testing and review.
- Large changes: full documentation, review, and testing akin to new model development.

Similarly, changes in use of the model must be documented, justified and tested as above.

# Model validation

# Model validation

Model validation:

- Verifying that the models are performing as expected.
- Verifying that the models are in line with their design objectives and business uses.
- Carried out by an independent the model validation team.

Guiding principal - “effective challenge”:

- Objective, informed parties.
- Applying critical analysis to models and their use.
- Identifying model limitations.
- Findings are taken seriously and promptly enacted upon.

# Validation management

The regulations require sound management practices as well:

Model validation must:

- Be independent from model development and model use.
- Have no stake in model validity.
- Be supported by a separate reporting line from users of the model and from model development.
- Have a high level of expertise and experience.
- Have a significant degree of familiarity with the business lines using the model and the intended uses.

# Validation management

There are also requirements on the reporting structure and the corporate culture:

- Incentives and compensation of the model validation team must promote high quality, critical work.
- The corporate culture must encourage objective thinking, and the questioning and challenging of ideas.
- The model validation team must have explicit authority to challenge developers and users and to elevate issues and deficiencies.
- The model validation team must report to someone with sufficient influence to ensure that issues and deficiencies are quickly mitigated.
- Management must disallow the use of a model if significant deficiencies are found.

# Validation activity

Core activities of model validation:

- Evaluation of conceptual soundness.
- Ongoing monitoring.
- Outcome analysis and backtesting.

We will review each aspect in detail.

# Conceptual soundness

Evaluation of conceptual soundness:

- Careful review of documentation by independent experts.
- Ensure that model is appropriate for task at hand.

Goals and activities:

- Assess the quality of the model design and construction.
- Review the documentation and empirical evidence justifying the model and it's validity and accuracy.
- Ensure sound judgment was used in model development.
- Review model *before* it goes into use.
- Review model whenever material changes are made.
- Ensure documentation and testing done during model development was adequate and valid.
- Perform sensitivity analysis to ensure the model is stable in the face of multiple simultaneous changes of model inputs.
- Develop a clear action plan based on testing and evaluation results, such as modifying model properties, putting less reliance on its outputs, placing limits on model use, or developing a new approach.

# Ongoing monitoring

Ongoing monitoring:

- Critical to ensuring that the model continues to perform well in the face of day to day changes in:
  - products,
  - exposures,
  - firm activities, and
  - the markets.

Goals:

- Ensure that the model is appropriately implemented.
- Ensure that model is being used as intended.
- Ensure that model is performing adequately in current scope and market.

# Ongoing monitoring

Ongoing monitoring requirements:

- Design a program of ongoing testing and evaluation, including verification and bench marking.
- Model limitations should be regularly reassessed.
- Verify internal and external data inputs continue to be accurate, complete and consistent with the model purpose and design.
- Ensure rigorous quality and change control procedures are applied to the code.
- Perform system integration testing.
- Perform ongoing sensitivity, robustness and stability testing.
- Monitor model input values to detect when inputs start to approach model limitations.
- Monitor overrides applied to models. High rates of overrides indicate a model needs revision or redevelopment.
- Compare inputs and output to other sources.
- Compare model outputs to other models.

In all cases which indicate the potential deterioration of model performance, issues must be escalated and addressed.

# Outcome analysis

Outcome analysis and backtesting

- Compare model outputs to actual outcomes.
- Confirm that the model is realistic and performs well.
- Take action to address any shortcomings that are found.

# Outcome analysis

Outcome analysis activities:

- Document tests run, justification for selected tests, testing results, and analysis of testing results.
- Confirm that model forecasts are accurate.
- Confirm that model outputs conform to expected range of outcomes.
- For VaR, do backtesting - confirm that percentage of times VaR is exceeded (VaR exceptions) are close to VaR percentile, that exceptions are not clustering, or growing, test a variety of percentiles, etc.
- For other models, develop tests analogous to VaR backtesting tests, such as P&L attribution tests for pricing models.

# Outcome analysis

Outcome analysis scope:

- Thoroughness of testing must be commensurate with the impact of model failures.
- Tests must be tailored to the situation – type of model, usage of model, impact of model outputs. No test applies to all situations.
- Apply regression tests to changes in data, code or other adjustments to confirm that changes improve outcomes. If changes do not outperform, additional changes or a wholesale redesign are called for.
- Develop early warning tests for long forecast horizons (i.e. don't wait for long forecasts to prove unreliable).

# Testing numerical code

# Testing numerical code

Testing of numerical software is especially difficult:

- Non-numerical software:
  - Fairly straight forward.
  - Errors are usually obvious.
- Numerical software:
  - Output is just a numerical value
  - Hard for a person to judge if the value is correct.

Need catalog of testing techniques. We will discuss:

- Failure modes.
- Use of special cases.
- Confirming expected behavior.
- Robustness testing.
- Backtesting.
- Data testing.

# Failure modes

It is important to be familiar with the failure modes of numerical computations.

Ask during development and during code reviews:

- Where can underflows and overflows occur and how will they propagate through the calculation?
- How do approximations propagate through the calculations and impact results?
- Are sums or differences being computed of values that are too close together or too far apart?
- Are we following best practices for numerical code and numerical programming?

# Beware addition

Sums and differences:

- Important when approximating derivatives using the difference in function values.
- $B - A = B$ if $B$ is too large relative to $A$.
- $B - A$ is noise if $A$ and $B$ are close but noisy (and noise is independent).
- Errors can increase or decrease depending on whether they are positively or negatively correlated.

# Approximations

Numerical code is always an approximation.

Finite difference and lattice methods:

- Price functions are piecewise linear (or log-linear, or exponential, etc) with respect to at least some of the inputs.
- Difference derivatives for small $\Delta x$:
  - the slope of the piece in question
  - jumps between pieces
- Second derivatives:
  - Zero (or otherwise misleading results) when the interval lies on one of the pieces
  - Spikes when the interval straddles a corner.

"Exact" solutions:

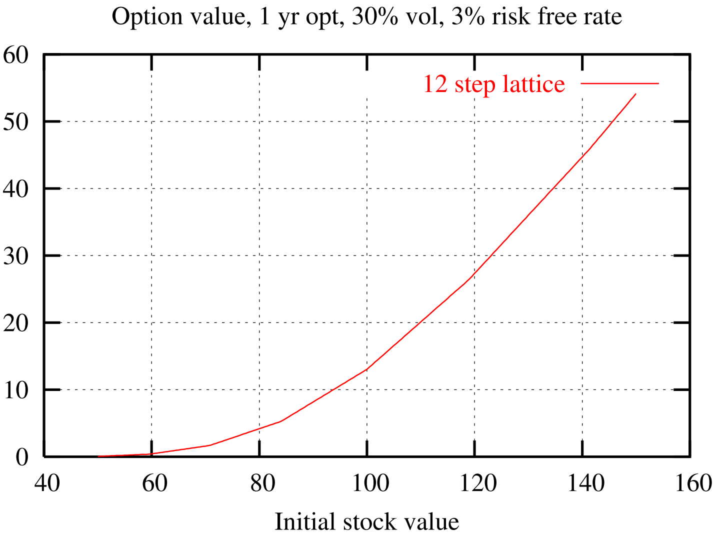
- Still have limited accuracy.
- Errors can grow as the number of arithmetic operations grows.

General Greeks:

- Pick $\Delta x$ to balance convexity error and cancellation.

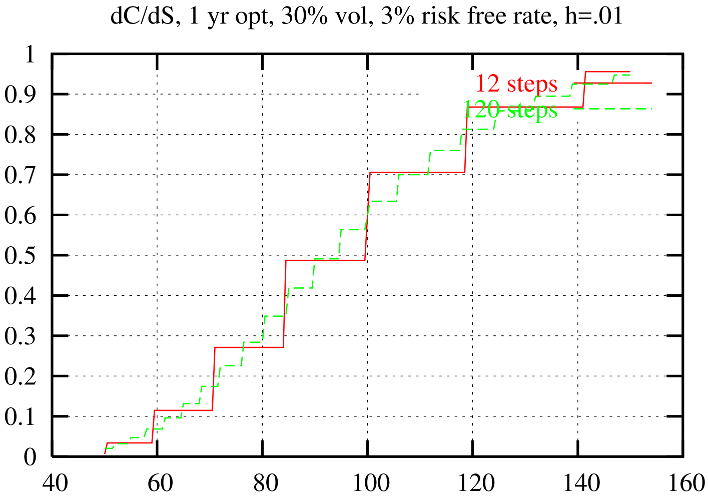
# Example price plot

Consider computing an option price with a coarse lattice. The price graph appears reasonable:



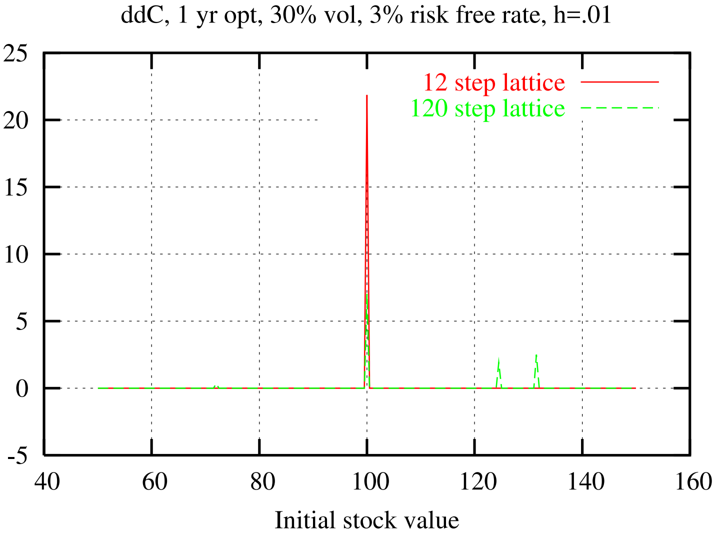
# Difference derivative plot

Computing and graphing the difference derivative with a step size of 0.01 highlights that the calculation is piecewise linear, and that delta requires care:



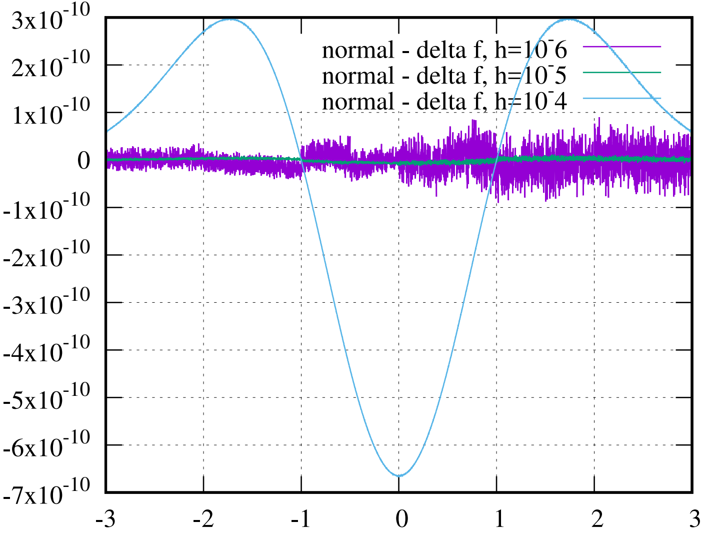
# 2nd order difference derivative plot

Gamma will be complete garbage unless action is taken:



# Analytic errors

Even analytic calculation are subject to such errors.



# Failure modes

Quantitative developers should be familiar with the strengths and weaknesses of different numerical methods.

Best practices:

- See standard numerical analysis texts (not Numerical Recipes).
- Study *What Every Computer Scientist Should Know About Floating-Point Arithmetic*, Goldberg, [Gol91].
- See *Risky Measures of Risk: Error Analysis of Numerical Differentiation*, Stein, [Ste05].

# Special cases

Knowledge and information about special cases can help to confirm the correctness of model implementations.

- Replicate numbers quoted in the literature.
- Compare to known solutions.
- Compare to simpler models and simpler cases.
- Compare to a different implementation (prototype, Monte Carlo, older version, etc).

# Comparisons

Comparisons:

- Confirm identical behavior when complex model reduces to simpler model.
    - Zero volatility
    - Zero strike
    - Flat volatility
    - Zero vol of vol
    - ...
- Confirm behavior of complex model relative to simpler model.
    - Using knowledge that more complex model prices certain options higher.
- Compare different implementations:
    - Prototype.
    - MC vs finite difference.
    - New version vs validated older version.
    - Internal Greeks vs bump and recompute.

# Behavioral confirmation

Facts that one knows about models should be checked.

- Document facts.
- Confirm that implementation exhibits the predicted behavior.

Examples:

- Monotonicity of the price as a function of various inputs.
- Limiting case behavior.
- Reasonable behavior between limiting cases.
- Comparisons between different cases.
- Confirm model implications hold.
- Internal consistency tests.
- Plotting results.

# Behaviors

Limiting cases:

- Call option prices tending to spot or zero as strike goes to zero or infinity.
- Price goes to intrinsic as vol goes to zero.
- Reasonable behavior between limiting cases.
  Comparisons between different cases:
  - American price >= Bermudan >= European.
  - Getting European price from American pricer on a European option.

# Behaviors

Confirm model implications hold:

- Put-call parity holds.
- Arbitrage is not violated.
- Dominating contracts have higher prices.

Internal consistency tests:

- Forward distributions and conditional distributions match analytic results
- Forward rates of different tenors make sense.

Plotting:

- Plot and inspect prices and first and second order difference derivatives.
- First derivative will confirm or refute monotonicity.
- Pricing noise will show up in 1st and 2nd derivatives.

# Robustness testing

There are various ways to check that models, numerical methods, and implementations are at least precise and robust, if not accurate.

1. Convergence tests.
2. Inversion tests.
3. Robustness and stability tests.

# Convergence tests

Convergence testing:

- Monte Carlo error estimates.
- Price as a function of number of scenarios.
- Price as a function of grid size (for finite difference).

Should yield an estimate of accuracy when running in production.

# Inversion tests

Inversion testing:

- Confirm that numerical inversions work.
- Check error bounds from simple root finders.
- Confirm calculated solutions are in fact solutions.
- Confirm calibrated models reprice inputs.

# Robustness and stability

In good models, parameters are stable.

- Small perturbations of model inputs should not lead to large changes in internal model parameters.
- Internal model parameters should be relatively stable over time.
- Model outputs should be relatively stable over time. goes for model outputs.

Failures could be due to numerical method instabilities or model inappropriateness or instability.

# Robustness and stability

There are a number of different sorts of backtests that can be used to compare models.

1. P&L attribution.
2. VaR backtesting.
3. Hedge effectiveness.

# P&L attribution

P&L attribution:

- Confirms that modeled price changes match observed price changes.
- Discrepancies can indicate missing risk factors

Methodology:

- Price portfolio each day.
- Compare computed price changes to actual price changes.
- Expect unexplained P&L to have zero mean and low variance (relative to actual P&L variance).
- Final FRTB Basel rule [Bas16]:
  - Mean $\pm 10\%$ constitutes a breach.
  - Relative variance $> 20\%$ constitutes a breach.
  - Reject model and revert to standard approach if $\geq 4$ breaches of last 12 monthly calculations.

# VaR backtesting

VaR testing:

- Compare actual P&L to VaR to confirm accuracy of VaR.
- P&L exceeding VaR is an exception.
- Expect exceptions in $(1-p)\%$ of the cases for $p$ percentile VaR.
- Final FRTB Basel rule [Bas16]:
  - Compute exceptions daily over last 250 days.
  - Excessive exceptions increase capital requirements.
  - $>30$ 97.5% VaR exceptions, must revert to standard approach.
  - $>12$ 99% VaR exceptions, must revert to standard approach.
- For more sophisticated tests and result analysis, see Cogneau [Cog15].

# Hedge effectiveness

Hedge effectiveness.

- Backtest hedging strategy for a variety of options.
- Compute P&L.
- Nonzero mean means model is biased.
- Large variance means poor model.

# Data testing

Data testing is done to confirm that the data itself is of high quality.

Tests include:

1. Confirming "reasonableness" of individual historical time series.
2. Comparing pricing sources for particular tickers.
3. Comparing values and histories of related tickers.
4. Investigating consistency compared to models.

# Reasonableness

Reasonableness:

- Reasonable number of outliers in historical time series.
- Values do not oscillate.
- Different pricing sources are in agreement.

# Like comparisons

Compare similar contracts:

- Compare price histories for different points on the same curve.
- Compare price histories for stocks and indices.

# Model analysis

Model analysis:

- Bonds whose OASs are outliers might be suspect.
- Futures and forwards can be compared.
- Curves can be stripped and checked for anomalies such as negative probabilities (CDS) or negative or spiking forward rates (IR).

# Summary

# Summary

Model risk management:

- Fastest growing area in finance.
- 3 components:
  1. Effective governance.
  2. Robust model development, implementation and use.
  3. Sound model validation practices.

Governance:

- Firm-wide respect for model risk – "Safety first".
- Independent reporting lines.
- Auditing.

# Summary

Model development:

- Follow sound development practices:
  - Specification document.
  - Model design document.
  - Data analysis.
  - Software design document.
  - Code review and coding standards.
  - Testing.
  - System analysis and testing.
  - Change management.
- Followup on performance after deployment.

# Summary

Model validation:

- Separate team.
- Separate line of reporting.
- Critical review of models.
- Principal of "effective challenge".

# Summary

Testing:

- Numerical code is more difficult to test than other code.
- Know the failure modes.
- Check special cases.
- Check overall behavior.
- Robustness testing.
- Backtesting.
- Data testing.

# References I

[Cog15] Philippe Cogneau. *Market Risk Management: Backtesting Value-at-Risk Models*. Tech. rep. Business and Decision, June 2015. URL: http://www.businessdecision.fr/include/viewFile.php?idtf=7390&path=94\%2F7390_266_RTC-White-Paper-Market-Risk-Management-Backtesting-Value-at-Risk-models.pdf.

[Gol91] David Goldberg. "What Every Computer Scientist Should Know About Floating-Point Arithmetic." In: *Computing Surveys* (Mar. 1991). URL: http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html.

[Ste05] Harvey J. Stein. "Risky Measures of Risk: Error Analysis of Numerical Differentiation." In: *SSRN* (June 2005). DOI: 10.2139/ssrn.955724. URL: http://ssrn.com/abstract=955724.

# References II

[Bas13] Basel Committee on Banking Supervision. *Fundamental review of the trading book — second consultative document.* Tech. rep. Bank of International Settlement, Oct. 2013. URL: http://www.bis.org/publ/bcbs265.htm.

[Bas16] Basel Committee on Banking Supervision. *Minimum capital requirements for market risk.* Tech. rep. Bank for International Settlements, Jan. 2016. 88 pp. URL: https://www.bis.org/bcbs/publ/d352.pdf.

[FED09] FED. *Application of the Market Risk Rule in Bank Holding Companies and State Member Banks.* Tech. rep. Board of Governors of the Federal Reserve System, Jan. 2009.

[FED11a] FED. *Financial Performance.* Tech. rep. Board of Governors of the Federal Reserve System, 2011.

# References III

[FED11b] FED. *Guidance on Model Risk Management*. Tech. rep. Board of Governors of the Federal Reserve System, Apr. 2011.

[OCC11] OCC. *Supervisory Guidance on Model Risk Management*. Tech. rep. Office of the Comptroller of the Currency, US Treasury, Apr. 2011. URL: http://www.occ.treas.gov/news-issuances/bulletins/2011/bulletin-2011-12a.pdf.

[OCC14] OCC. *OCC Guidelines Establishing Heightened Standards for Certain Large Insured National Banks, Insured Federal Savings Associations, and Insured Federal Branches*. Tech. rep. Office of the Comptroller of the Currency, US Treasury, Sept. 2014.