# Semi-Analytic Conditional Expectations (GMM-DCKE)

- Data Driven, Model Free, with Applications -

Bayes Financial Engineering Workshop,

London, November 2022


Jörg Kienitz

https://www.acadia.inc/our-people/jorg-kienitz, joerg.kienitz@acadia.inc

**acadia**, BUW, UCT, and, **Finciraptor**

finciraptor.de, joerg.kienitz@finciraptor.de





# Disclaimer

This presentation and any accompanying material are being provided solely for information and general illustrative purposes. The author will not be responsible for the consequences of reliance upon any information contained in or derived from the presentation or for any omission of information therefrom and hereby excludes all liability for loss or damage (including, without limitation, direct, indirect, foreseeable, or consequential loss or damage and including loss or profit and even if advised of the possibility of such damages or if such damages were foreseeable) that may be incurred or suffered by any person in connection with the presentation, including (without limitation) for the consequences of reliance upon any results derived therefrom or any error or omission whether negligent or not. No representation or warranty is made or given by the author that the presentation or any content thereof will be error free, updated, complete or that inaccuracies, errors or defects will be corrected.

The views are solely that of the authors and not of any affiliate institution. The Chatham House rules apply.

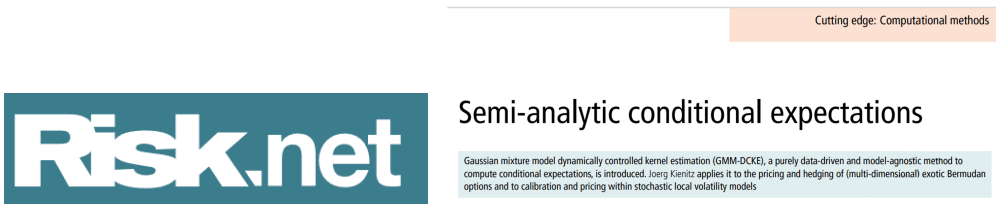
# Overview

# Introduction

# Introduction and Objectives

- Conditional Expectations and Financial Applications
- Examples
  - Bermudan Options
  - Stochastic Local Volatility Models
  - xVA/Exposure
- GMM-DCKE is a suggestion to solve these tasks in a semi-analytic, model free, data driven fast way

Based on the publication (**Cutting Edge - RISK, 7/2022**):
**GMM-DCKE - Semi Analytic Conditional Expectations**,
`https://www.risk.net/cutting-edge/7950701/semi-analytic-conditional-expectations`
`https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3902490`




## Examples:

`https://github.com/Lapsilago/GMM_DCKE`

# Introduction

- Calculating conditional expectations is notoriously difficult with just a few exceptions including the Gaussian case.
- Often necessary to apply Monte Carlo simulation (high dimensions, complex dynamics, ...)
- Nested simulations can be used but they are computational expensive and, thus, slow
- Market standard: Least squares regression (LSM)

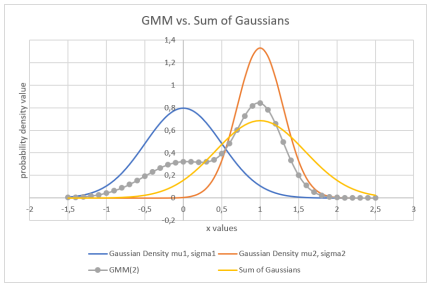
# GMM DCKE

# Gaussian Mean Mixture (GMM)

- Let $K \in \mathbb{N}$, and $Z_k$, $k = 1, \ldots, K$ be $d$-dimensional random variables, $Z_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, with densities $n_{Z_k}$
- Definition:

Consider the *Gaussian Mean Mixture* $Z$ determined by the density

$$p_Z = \sum_{k=1}^{K} \pi_k n_{Z_k},\ \pi_k > 0,\ \sum_{k=1}^{K} \pi_k = 1.$$




- $\pi_k$ - *mixing weights*, GMM(K)- $K$ component GMM with

$$\mu_Z = \sum_{k=1}^{K} \pi_k \mu_k,\ \mathbb{V}(Z) = \sum_{k,j=1}^{K} \pi_k \pi_j \mathbb{V}[Z_k Z_j]$$

# GMM DCKE - Input

- *Training set* $\mathcal{X} = (x_1, \ldots, x_N)$, $x_n \in \mathbf{R}^d$,
- with labels $\mathcal{Y} = (y_1, \ldots, y_N)$, $y_n \in \mathbf{R}$,
- $(x_n, y_n)$ are joint realizations of random variables $X$, $Y$.
- $X$ are the underlying risk factors for some $t < T$ and $Y$ is a function of $X$, e.g. a payoff at maturity $T$ for Bermudan derivative or the variance values conditioned on the underlying in a Stochastic Local Volatility model.
- $\mathcal{X}^* = (x_1^*, \ldots, x_M^*)$, $x_n^* \in \mathbf{R}^d$, be the test set.

```
dimS = 5

# marginals at time t and T
S1t = tspot(S[tindex][:,0], t_spot)
S2t = tspot(S[tindex][:,1], t_spot)
S3t = tspot(S[tindex][:,2], t_spot)
S4t = tspot(S[tindex][:,3], t_spot)
S5t = tspot(S[tindex][:,4], t_spot)
St = [S1t, S2t, S3t, S4t, S5t]

S1T = tspot(S[Tindex][:,0], t_spot)
S2T = tspot(S[Tindex][:,1], t_spot)
S3T = tspot(S[Tindex][:,2], t_spot)
S4T = tspot(S[Tindex][:,3], t_spot)
S5T = tspot(S[Tindex][:,4], t_spot)
ST = [S1T, S2T, S3T, S4T, S5T]
```

```
# option prices at T
Ot = tprice(BasketPayoff(S[tindex],weights,basketK, leverage), t_price)
OT = df * tprice(BasketPayoff(S[Tindex],weights,basketK, leverage), t_price)
```

**Remark:**
**The training set with labels need not necessarily be samples from a specific model/class of models (SDEs), thus, the method is model free and data driven!**

**In case of using a model adjoint differentiation to construct the CV might be better suited.**

# GMM-DCKE - Parameters and Output

**Parameters**:

- The number of Gaussian distributions $K$
- Realizations $\mathcal{Z} = (Z_1, \ldots, Z_N)$ of control variates determining for each $x$, $\mu_Z(x) := \mathrm{E}[Z \mid X = x]$.
  (Remark:
  Minimal variance delta if $\mathcal{Z}$ being the discounted assets)

**Output:**

- Predictions $y^* = (y_1^*, \ldots, y_M^*)$, $y_j^* \approx \mathrm{E}[Y \mid X = x_j^*] \in \mathbf{R}$, i.e. conditional expected values of a Bermudan derivative, or expected variance given the risk factor value in a Stochastic Volatility model.
- For in-sample performance we take $\mathcal{X}^* = \mathcal{X}$.

# GMM-DCKE - Calculation

- Fit GMM(K) to approx. the joint distribution of $X$, $Y$ and $Z$ using EM, **numeric!**.
- Calculate cond. distribution for $Y$ given $X$.
- GMM(K) for variance reduction with CV $Z$ given by

$$Y^*(x) := Y(x) + \beta(x)(Z(x) - m_Z(x)) \qquad (1)$$

$m_Z$ (cond. mean) and $\beta \in \mathbf{R}^d$ for all $x \in \mathcal{X}$ (cond. proxy hedge), **analytic!**, with

$$\beta(x) := -\frac{COV[Y, Z \mid X = x]}{\mathrm{Var}[Z \mid X = x]}. \qquad (2)$$

$(x)$ indicates the conditioning on $X$.

```
# gather data for fitting GMM(K)
ngmm = 5
X_estimate = np.vstack((CV, OT,S1t, S2t, S3t, S4t, S5t)).T   # set up the traini

# fit GMM(K)
gmm_x = mixture.GaussianMixture(n_components=ngmm).fit(X_estimate)

# calculate the conditional distributions
gmm_cond_estimate = GMM_Conditional(means=gmm_x.means_,
                          covariances=gmm_x.covariances_,
                          weights=gmm_x.weights_,
                          n_components=gmm_x.n_components,
                          D1=2, D2=5)


# CV calculations

for m in cond_models_list:
    Srange = Srange_m[:,j]   # values for the CV
    llist_mu = []            # local list for mu
    llist_delta = []         # local list for delta
    for s in Srange:
        # calculated conditional values given s for CV
        mu_cond, covar_cond, weights_cond = m.par_x1_cond_x2(s)
        llist_mu.append(calcm(mu_cond, covar_cond, weights_cond))
        llist_delta.append(calcdelta(mu_cond, covar_cond, weights_cond))
    glist_mu.append(np.asarray(llist_mu))
    glist_delta.append(np.asarray(llist_delta))

garray_mu = np.asarray(glist_mu)       # global list with cond mus
garray_delta = np.asarray(glist_delta) # global list with cond deltas

# calculated CVs and overall CV adjustment
CV_c = np.zeros(len(Srange_m[:,0]))
i = 0
for d, m in zip(garray_delta, garray_mu):
    CV_c += d * (Srange_m[:,i] - m[:,0])
    i = i + 1
```

**$\beta$: Analytic, model free, data driven, time discrete cond. minimal-variance $\delta$ if $Z$ is underlying!**

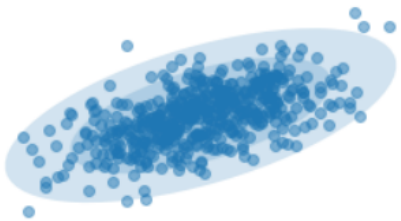
# The Parameters - Start

- $K$, $\mathcal{X}$ of size $N$ with $x \in \mathbb{R}^d$, $\mathcal{X}$ is a $N \times d$ matrix.
- Goal: Optimize *(log-)likelihood function* $l(\theta)$, for GMM(K),

$$l(\theta) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k n(x_n, \mu_k, \Sigma_k) \right)$$

- Take derivatives wrt $\mu_k$, $\Sigma_k$ and $\pi_k$ and set them to 0.
- The results are not in closed form and depend on all the parameters in a complex way.

GMM(K) defines $K$ clusters according to $(\pi_k, \mu_k, \Sigma_k)_{k=1,\ldots,K}$ - $d = 2$ clusters corresponds to ellipses and generalize to ellipsoids in higher dimensions settings.



# The Parameters - Equations

For $\mu_k$ this leads to consider

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \underbrace{\frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l N(x_n|\mu_l, \Sigma_l)}}_{:=\gamma(u_n k)} x_n, \quad N_k := \sum_{n=1}^{N} \gamma(u_{nk})$$

$N_k$ is the effective number of points assigned to component $k$.
For $\Sigma_k$ and $\pi_k$ we obtain:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(u_{nk})(x_n - \mu_k)(x_n - \mu_k)^\top, \; \pi_k = \frac{N_k}{N}$$

Solve via *Expectation Maximization* (EM), Dempster A. P., Larid N. M., Rubin D. B. (1977):

# The Parameters - EM

- Initialize, evaluate with new parameters (E-M steps) and check for convergence!
- **E-step**: Evaluate

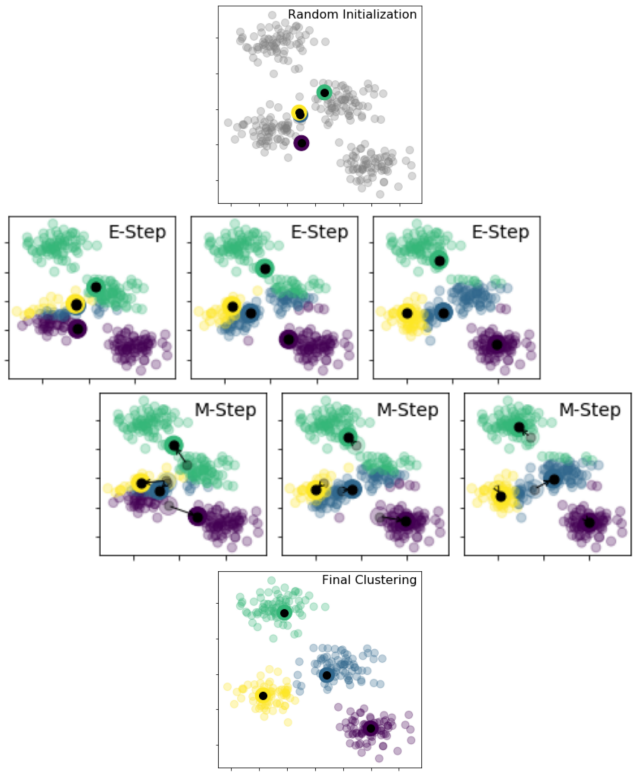
$$\gamma(u_k) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{l=1}^{K} \phi_l N(x_n|\mu_l, \Sigma_l)}$$

- **M-step**: Estimation of the parameters

$$\mu_k^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(u_{nk}) x_n, \; \pi_k^* = \frac{\Gamma_k}{N}, \; \Gamma_k = \sum_{n=1}^{N} \gamma(u_{nk})$$

$$\Sigma_k^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(u_{nk}) \sum_{n=1}^{N} (x_n - \mu_k^*)(x_n - \mu_k^*)^\top$$

EM mechanism illustrated:



# Components

$K$ for GMM(K) via empirical results or statistical methods:

- **silhouette scores**, The silhouette score is calculated by taking the mean distance between a sample and all other points in the same cluster, resp. the mean distance between this sample and all other points in the next nearest cluster. This method exploits the compactness and separability of the constituting clusters.,
- **Information criteria**, *Akaike information criterion* - *aic*, resp. *Bayesian information criterion* - *bic*,

$$aic := 2K - 2\ln(\hat{L}); \quad bic = K\ln(N) - 2\ln(\hat{L})$$

$\hat{L}$ is the value of the likelihood function with parameters $\hat{\theta}$ maximizing it. Then, choose the model with aic, resp. bic gbeing smallest. Can lead to models with a large number of components, and thus, overfitting!, or,

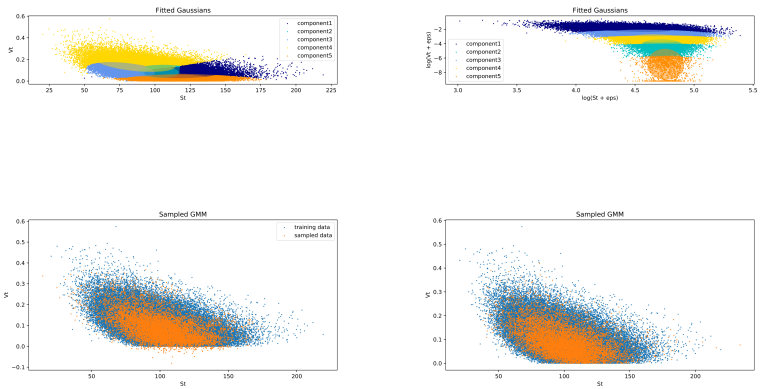
- **Gradients**, To this end gradients of the quantities are considered and we choose the number of components where the gradient starts to get flat. For instance we can use a condition like $|\nabla bic| \leq \epsilon$.

# Enhancements

- **Bagging** (bootstrap aggregation) - an ensemble method, see Bishop C. M. (2006). For GMM choose $k, n \in \mathbb{N}$ and perform it $k_{bag}$ times with $N_{bag}$ observations randomly sampled from $\mathcal{X}$ with replacement. Finally, we average over the $k$ experiments. It is a dimension reduction technique!
- **Dimensionality** - Curse of dimensionality - EM algorithm performs $N \times d$ computations for the means and $N \times \frac{d \times (d-1)}{2}$ for the covariances. Thus, it is at least quadratic in the number of dimensions. Parametric Covariances or dedicated research on EM can be applied to our setting to mitigate the curse.
- **Weight Redistribution** - define weight threshold $w_t$, if $w_i \leq w_t$, the sum of these weights is redistributed. Let $\mathcal{T}$ be the set of indices, s.t. $w_i \leq w_t$, $i \in \mathcal{T}$, denoting $w_{in} := \sum_{j \notin \mathcal{T}} w_j$, $w_{out} = \sum_{j \in \mathcal{T}} w_j$, for $i \notin \mathcal{T}$ is given by $w_i \times \frac{w_{in}}{w_{out}}$.

# Starting Values and Transformations

- Sometimes a particular initial choice of starting points for the clusters can be helpful.
- For some applications only positive values should be considered (option prices, stock prices, etc.).
- Gaussians accommodate negative values!
- Strictly positive values $\mathcal{X}$, work with $\mathcal{X}_{\epsilon,\log} = \log(\mathcal{X} + \epsilon)$.
- Transformations like LambertW or Yeo-Johnson for creating *near normal* data!



# Related Work

- Huge, B. N., Savine, A. (2020) - Differential Machine Learning - Using a DNN and incorporating the path-wise differential into estimating of future price distributions
- Halperin, I. (2017) - Q-Learner in Black-Scholes-Merton World - Highly recommended for establishisng the framework that links between Q-Learning and option pricing!
- Potters et al. (2001) Hedged Monte-Carlo: Least Squares (LSM) Low Variance Derivatives Pricing with Objective Probabilitis - Inclusion of the hedging strategy reduces the variance in pricing
- Geng Q., Kienitz J., Lee G. T., Nowaczyk N. (2022) DCKE: Kernel density estimation, control variates and Gaussian Process Regression.

**GMM-DCKE incorporates concepts of all these methods!**
**BUT: Techniques are different and it is semi-analytic!**

# Comparison to Related Work

GMM-DCKE compared to Geng Q., Kienitz J., Lee G. T., Nowaczyk N. (2022); Halperin, I. (2017); Potters, M., Sestovic, D., Bouchaud, J.-P. (2001); Huge, B. N., Savine, A. (2020)

- ... uses hedges for stabilizing and improving results
- ... is semi-analytic with all calculations for conditional values being analytic
- ... uses standard statistically sound methods for parameter estimation
- ... is not as data hungry as reinforcement learning or artificial neural network training
- ... provides a more direct access to discrete time option pricing and hedging than Q-learning

# Comparison to Related Work - Regression

For *basis functions* $(f_k)_{k=0,\dots,N_b}$ approximate the conditional expectation at time $s$ using values at time $t$ by

$$\mathbb{E}[v_t|x_s] = \sum_{k=0}^{N_b} a_k f_k(x_i) + \varepsilon \approx \sum_{k=0}^{N_b} a_k f_k(x_i).$$

Vector of coefficients via minimizing the least squares error, i.e. $\underline{a} = (F^\top F)^{-1} F^\top \underline{y}$, with labels $\underline{y}$, $F$ matrix with columns $F_i = f_i(x_j)$.

- may fail to converge properly even with large number of paths
- dependent on the choice of basis functions, resp. bandwidth for local regression, resp. kernel density estimation
- hedge sensitivities/quantiles are more error prone and heavily depend on local estimates.
- GMM-DCKE does not incorporate local estimates and leads to smooth estimates

# Hedging - a key to control variates

A portfolio with a risky asset $S$ and bank account $B$ at time $t_i$ replicates the payoff $V_{t_T}$ when:

$$\Pi_{t_i} = B_{t_i} + \phi_{t_i} S_{t_i},\ \phi_{t_T} = 0, \quad B_{t_T} = V_{t_T}. \tag{3}$$

$\phi_{t_i}$ represents a position in $S_{t_i}$. For a self-financing discrete hedge looking backwards and observe that the portfolio value is:

$$\Pi_{t_i} = e^{-r(t_{i+1}-t_i)} \left( B_{t_{i+1}} + \left( \phi_{t_{i+1}} - \phi_{t_i} \right) S_{t_{i+1}} \right) + \phi_{t_i} S_{t_i} \tag{4}$$

$$= e^{-r(t_{i+1}-t_i)} \Pi_{t_{i+1}} - \phi_{t_i} \left( e^{-r(t_{i+1}-t_i)} S_{t_{i+1}} - S_{t_i} \right) \tag{5}$$

The trading strategy $\phi_{t_i}, i \in \{0, \ldots, T\}$ determines the value.[1]

# Hedging - a key to control variates

How should we choose the trading strategy?
We could try using the variance minimizing hedging strategy, i.e.

$$\phi^*_{t_i} = \frac{\text{cov}\left[\Pi_{t_{i+1}}, \Delta S_{t_i} \mid S_{t_i}, t_i\right]}{\text{Var}\left[\Delta S_{t_i} \mid S_{t_i}, t_i\right]}, \tag{6}$$

(6) is the optimal proxy hedge, given model and with $S$ being the CV this is the time discrete minimal variance delta.
Minimising the estimation error of the cond. expectation using CV is the same as minimising the hedging error in a discrete options pricing framework!
The difference $t_{i+1} - t_i$ determines the re-balancing period. Thus, this time-lag determines the quality of the hedge. Small time-lags lead to the standard continuous delta hedges.

# Examples

# Setting

- We chose the Heston, the rough Bergomi, the Bates and the Variance Gamma, Heston, S. (1993); Madan, D. and Seneta, E. (1990); Bates, D. (1996); Bayer, C., Friz, P., and Gatheral, J. (2016) models for illustration.
- We denote the asset prices as $S(t)$, resp. $X_t := \log S_t$ for the logarithmic prices.
- For the multi-dimensional setting we choose the Heston model
- For xVA/exposure we choose the Trolle-Schwartz (Cheyette type with SV) model with constant parameters

# The Heston/Bates Model

- $S(0) = S_0$, $v(0) = v_0$ as the initial values for the asset price and the variance.
- SDEs governing the stochastic evolution of the Heston model are given by:

$$\begin{aligned} \frac{dS(t)}{S(t)} &= rdt + \sqrt{v(t)}dW_1(t), \\ dv(t) &= \kappa(\theta - v(t))dt + \nu\sqrt{v(t)}dW_2(t). \end{aligned} \tag{7}$$

- The Bates model has an additional (jump) term for $S(t)$ in (7), namely $(Y-1)dN(t)$.
- $N$ being a Poisson process with intensity $\lambda \geq 0$ and $Y \sim N(\mu_j, \sigma_j)$.

# The rough Bergomi Model

- For the instantaneous variance we consider the instantaneous forward variance $\xi_t^u, u \geq t$, for date $u$ observed at $t$.
- Let $\alpha = H - \frac{1}{2} \in \left(-\frac{1}{2}, 0\right]$ be a negative exponent depending on the *Hurst exponent* $H \in (0, \frac{1}{2}]$.
- For $\eta > 0$ we consider the rough Bergomi model is given by

$$\begin{aligned} dX_t &= -\frac{1}{2} V_t dt + \sqrt{V_t} dW_1(t), \\ d\xi_t^u &= \xi_t^u \eta \sqrt{2\alpha + 1} (u - t)^\alpha dW_2(t). \end{aligned} \tag{8}$$

- The variance is obtained from $\xi$ in (8) by considering $V = \xi e^{\eta Y - \frac{1}{2}\eta^2 t^{2\alpha+1}}$ with $Y$ being the corresponding realization of a Volterra process.

# The Variance Gamma Model

Let us consider a Γ-proces. We call $X_t$ a Γ-process (with independent gamma distributed increments) if for all time points $s, t$, $s \leq t$ we have that $X_t - X_s$ has a Γ-distribution. The parameters, the scaling - often $\lambda$ - and the shape - often $\gamma$ -, control the rate of the jump arrivals and the size of the jumps (inversely) and .
The variance gamma process can be defined in two equivalent ways:

(1) The difference of two Gamma processes $U(t)$ and $D(t)$, $X(t) := U(t) - D(t)$.

(2) A time-changed Brownian motion, $X(t) := W(Y(t))$, with a Gamma process $Y(t)$.

# Conditional Expectation - Heston Model

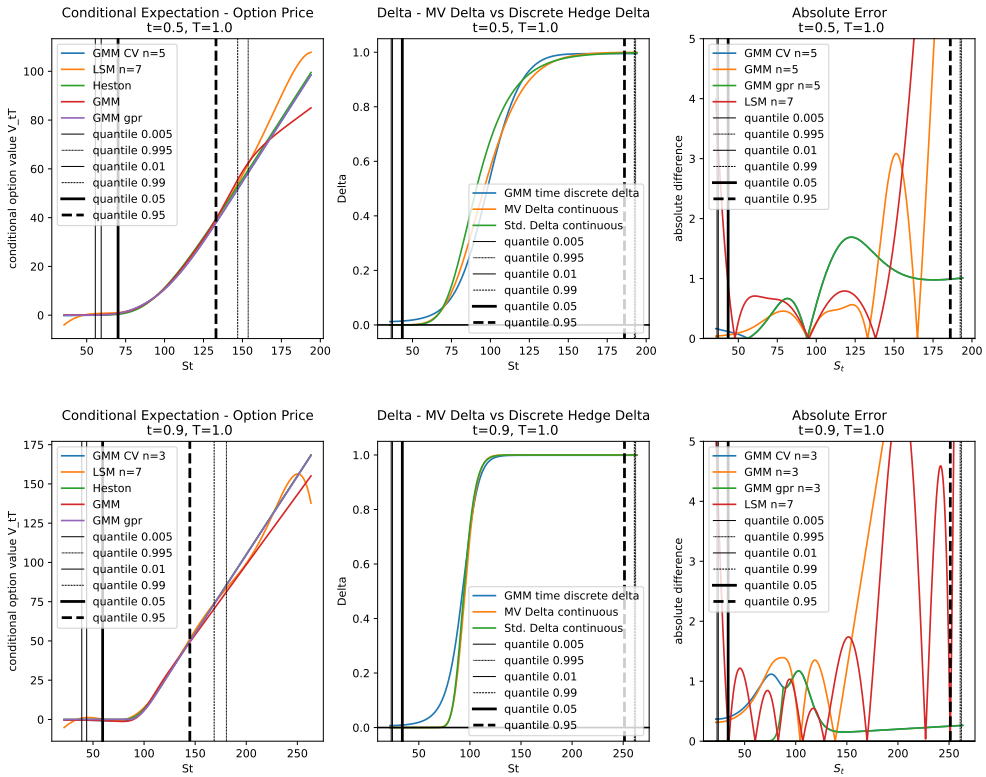



Figure: Conditional expectation and delta calculation for t=0.5, T=1 and t=0.9, T=1 for the Heston model.

# Conditional Expectation - rough Bergomi Model

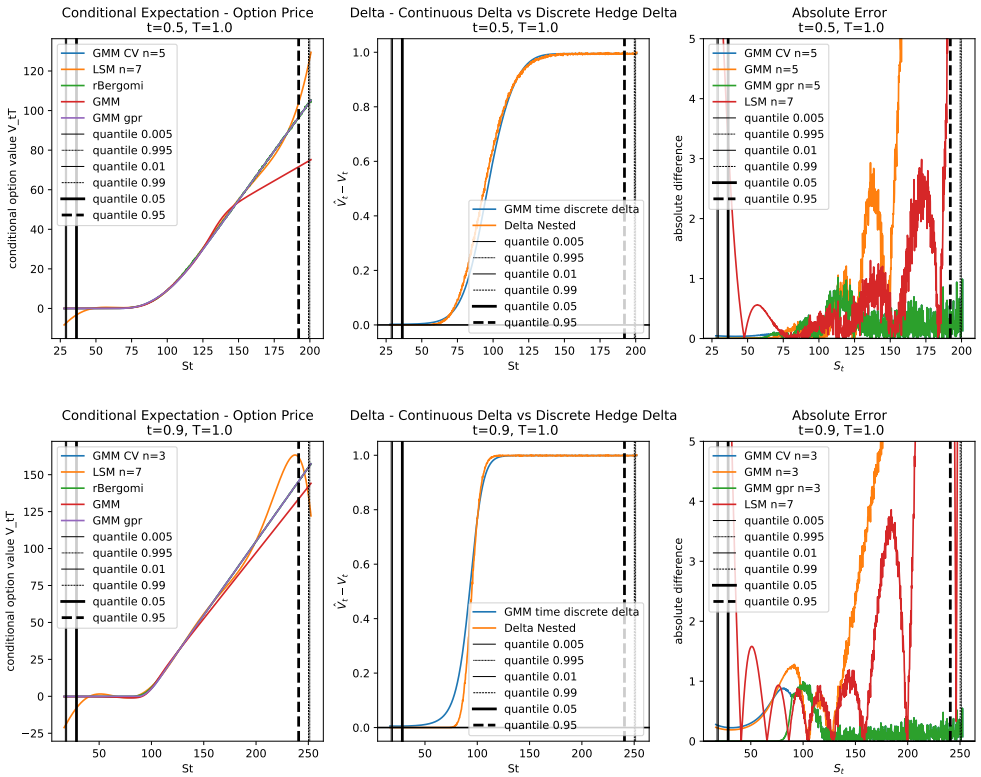



Figure: Conditional expectation and delta calculation for t=0.5, T=1 and t=0.9, T=1 for the rough Bergomi model.

# Conditional Expectation - Bates Model

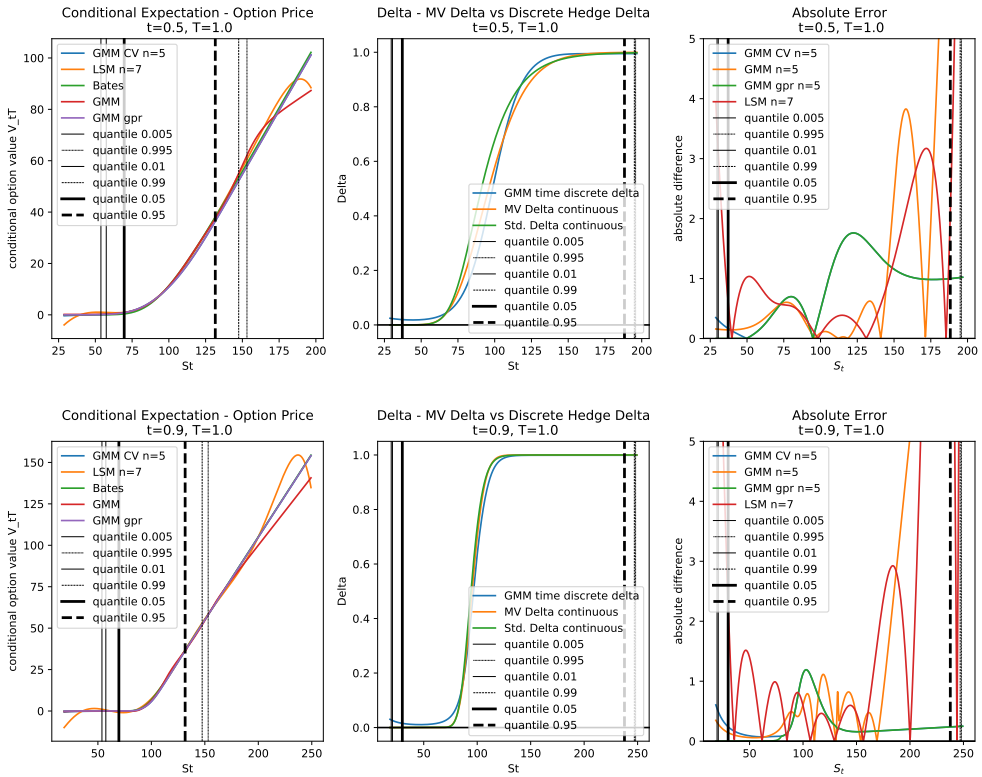



Figure: Conditional expectation and delta calculation for t=0.5, T=1 and t=0.9, T=1 for the Bates model.

# Conditional Expectation - Variance Gamma Model

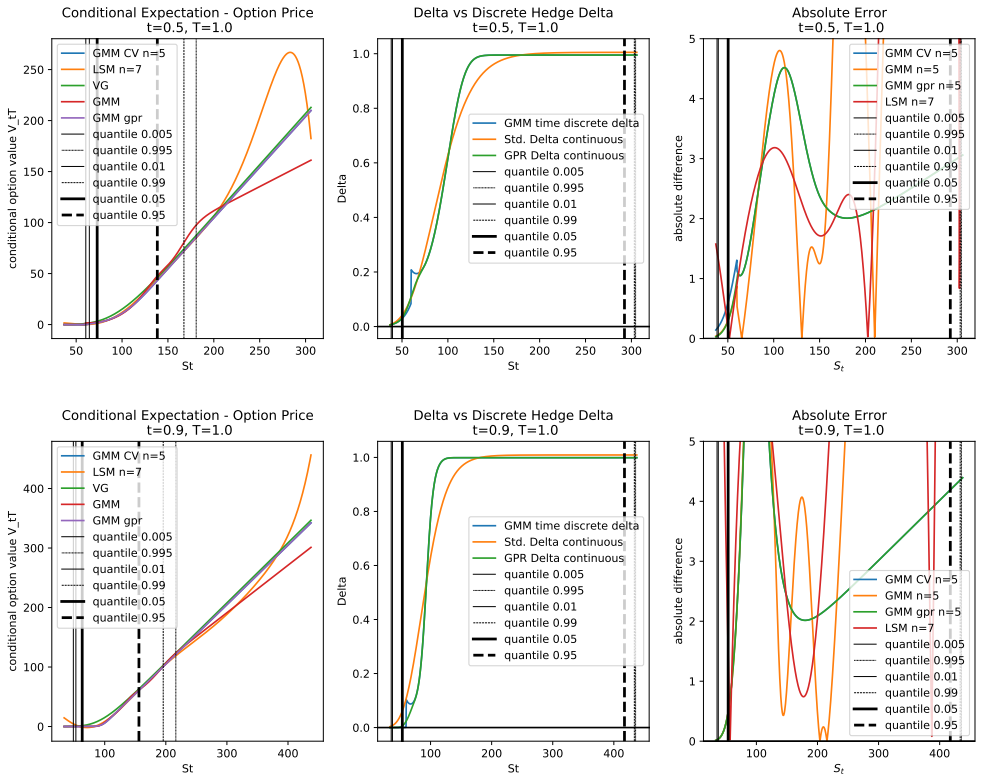



Figure: Conditional expectation and delta calculation for t=0.5, T=1 and t=0.9, T=1 for the Variance Gamma model.

# Observations - Bermudan Derivatives 1F

- Analytic smooth deltas and option prices
- The effect of the CV would be also for extrapolation, namely flat extrapolation (left tail) and linear interpolation wrt slope (right tail)
- Model free - we do not assume a specific model nor that it is a diffusion or a pure jump model
- Data driven - a GMM(K) fit to observed data points at $t = t_1$ and $t = t_2$
- Splitted GMM approximation demonstrated for the VG model, needs regression, e.g. Gaussian Process Regression (GPR), to interpolate between the splitted regions.

# Examples - Multiple Underlyings

As examples we consider *basket* and *rainbow options* with payoffs given by

$$h_{t_T}^{\text{basket}} = \left( \sum_{u=1}^{d} \omega_u S_{u t_T} - K \right)^+, \tag{9}$$

$$h_{t_T}^{\text{rainbow}} = \left( max \left( \frac{S_{1,t_T}}{S_{1,t_0}}, \ldots, \frac{S_{d,t_T}}{S_{d,t_0}} \right) - K \right)^+ \tag{10}$$

We take $S_{u,t_0} = 100$, $r = 0.01$, $K_{\text{basket}} = 95$, resp. $K_{\text{rainbow}} = .9$, $u = 1, \ldots, 5$. For the correlation we apply two different settings one with constant values of 0.6 and one with constant values of $-0.2$.

# Examples - Multiple Underlyings

The 5 dimensional Heston model: Asset correlation matrix (a-a) and asset-variance (a-v) correlations:

$$C_{a-a} = \begin{pmatrix} 1.0 & 0.2 & 0.0 & 0.5 & 0.7 \\ 0.2 & 1.0 & 0.4 & 0.0 & 0.1 \\ 0.0 & 0.4 & 1.0 & 0.3 & 0.2 \\ 0.5 & 0.0 & 0.3 & 1.0 & 0.25 \\ 0.7 & 0.1 & 0.2 & 0.25 & 1.0 \end{pmatrix}, \quad C_{a-v} = \text{diag} \begin{pmatrix} -0.7 \\ -0.8 \\ -0.9 \\ -0.8 \\ -0.3 \end{pmatrix}$$

The full correlation matrix is determined by applying a positive definite completion using $C_{a-a}$ and $C_{a-v}$.

# Example - Multiple Underlyings (Rainbow/Basket Option)

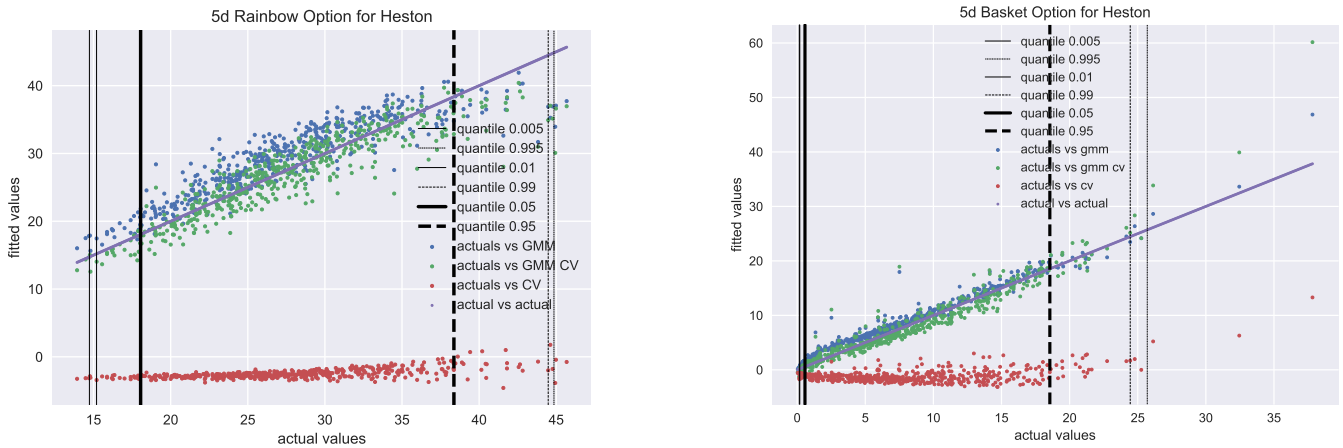



Figure: Valuation of a rainbow and a basket option using the 5-d Heston model with and without control variates. The used correction from the applying control variates is also displayed.

# Example - Multiple Underlyings (Rainbow/Basket Option)

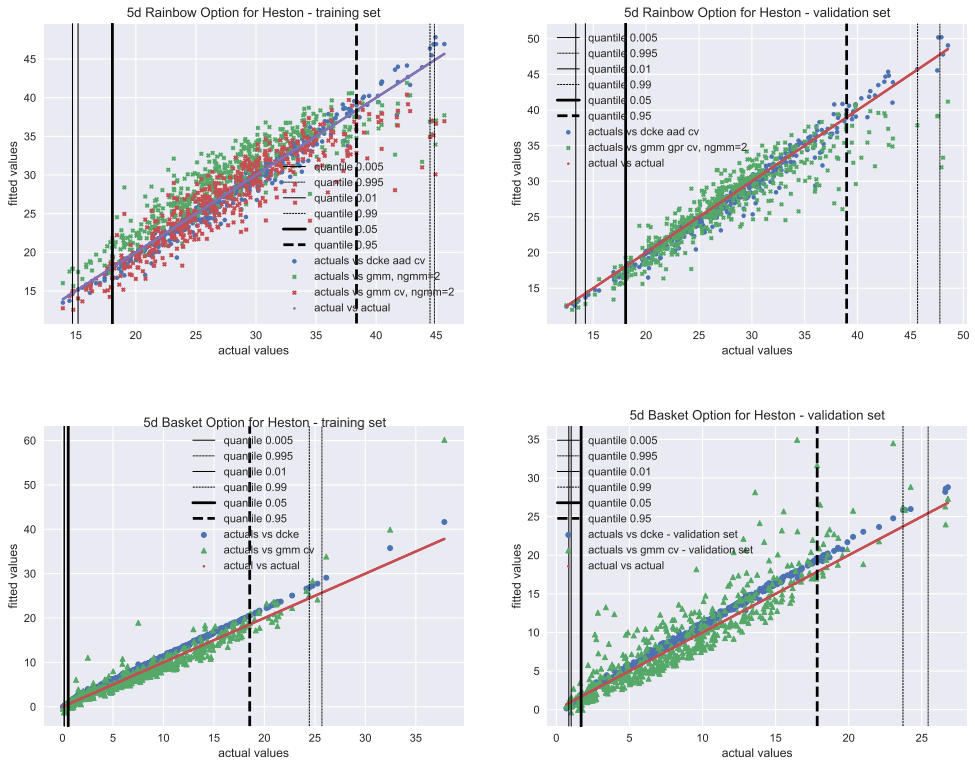



Figure: Rainbow and basket options in a 5-d Heston model on a test and validation set. Results from applying GMM-DCKE and DCKE are plotted. DCKE results used the model delta (via adjoints, resp. nested simulation). Thus, comparison a bit unfair since DCKE becomes model dependent!

# Observations - Bermudan Derivatives $n$F

- Analytic smooth deltas and option prices in multiple dimensions
- Model free and data driven - as explained before
- Effect of the control variate is shown
- Bagging possible and may improve results in $n$-dimensions significantly - not yet done!
- Values have to be compared with MC for $10,000$ paths! If the accuracy is not enough this can at least be used as a control variate with the estimate for $\beta$ which is challenging in $n$ dimensions.
- Other choices of control variates might be considered, e.g. $\max(X_t)$ or $\min(X_t)$. For instance using the max, resp. min for the rainbow option case leads to improvements!

# Different Control Variates

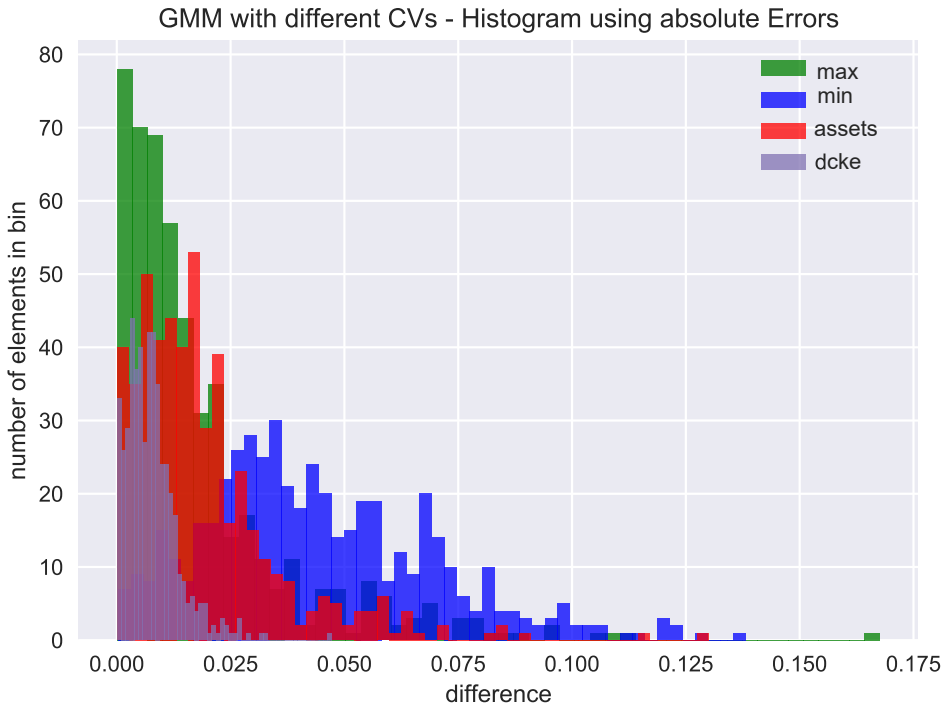



Figure: Different control variates - assets, the max and the min for GMM-DCKE and DCKE.

# Stochastic Local Volatility Model

Stochastic Local Volatility model:

$$\begin{aligned} \frac{dS_t}{S_t} &= rdt + \lambda(S_t, t)\psi(V_t)dW_t \\ dV_t &= \mu(t, V_t)dt + \sigma_V(t, V_t)dZ_t \end{aligned} \tag{11}$$

where $\mu, \sigma$ are the drift, resp. volatility of the instantaneous variance, $\lambda : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ is the leverage function, $W$ and $Z$ are one dimensional correlated Brownian motions. The function $\psi$ represents the stochastic volatility component[2], see Henry-Labordere P. (2009); Guyon J., Henry-Labordere P. (2009); Muguruza A. (2020); van der Stoep A., Grzelak L. A., Oosterlee, C. W. (2014)

# Stochastic Local Volatility Models

Special choices of $\mu$, $\sigma_V$ and $\psi$ lead to well known models, for instance $\mu(x) = \kappa(\theta - x)$, $\sigma_V(x) = \sigma\sqrt{x}$, $\psi(x) := \sqrt{x}$ lead to the Heston model.
Starting with a previously calibrated local volatility $\sigma_D$, Dupire, B. (1994), and a calibrated stochastic volatility model. To derive the leverage function from (11) for each time point we consider:

$$\lambda^2(t, K) = \frac{\sigma_D(t, K)^2}{\mathbb{E}[V_t | S_t = K]} \tag{12}$$

The cond. expectation in the denominator of (12) has to be calculated. GMM-DCKE provides an efficient tool!
We compare our results with the methods from van der Stoep A., Grzelak L. A., Oosterlee, C. W. (2014) demonstrated to be accurate.

# Example - Heston

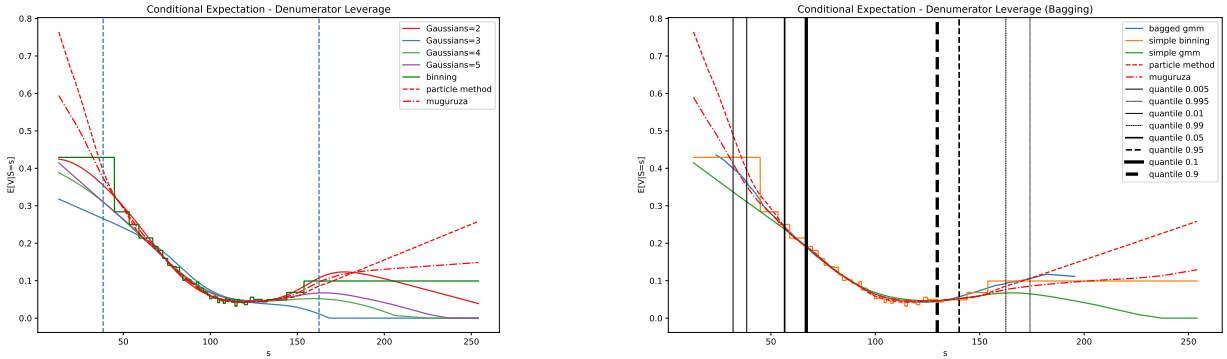



Figure: (top) GMM-DCKE for calculating $\mathbb{E}[V_t|S_t = s]$ for the Heston model with different numbers of Gaussians vs binning without bagging (left) and with bagging (right).

# Example - rough Bergomi

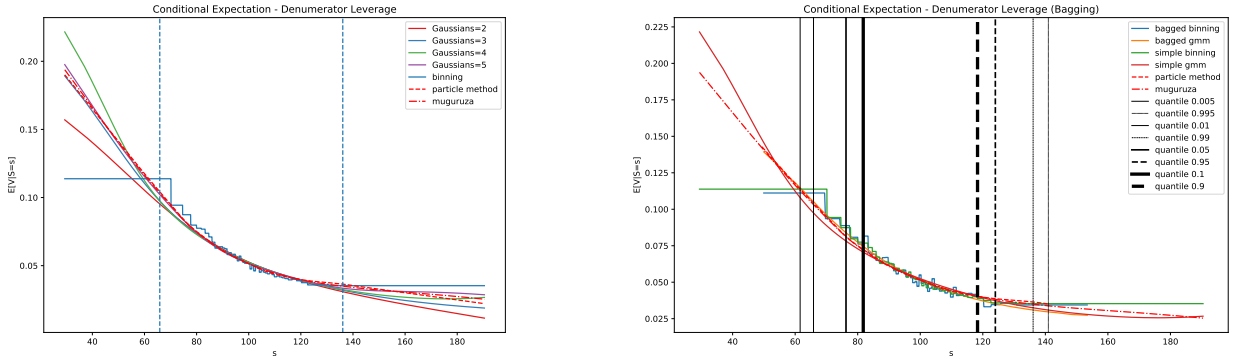



Figure: (top) GMM-DCKE for calculating $\mathbb{E}[V_t|S_t = s]$ for the rough Bergomi model with different numbers of Gaussians vs binning without bagging (left) and with bagging (right).

# Example - Pricing

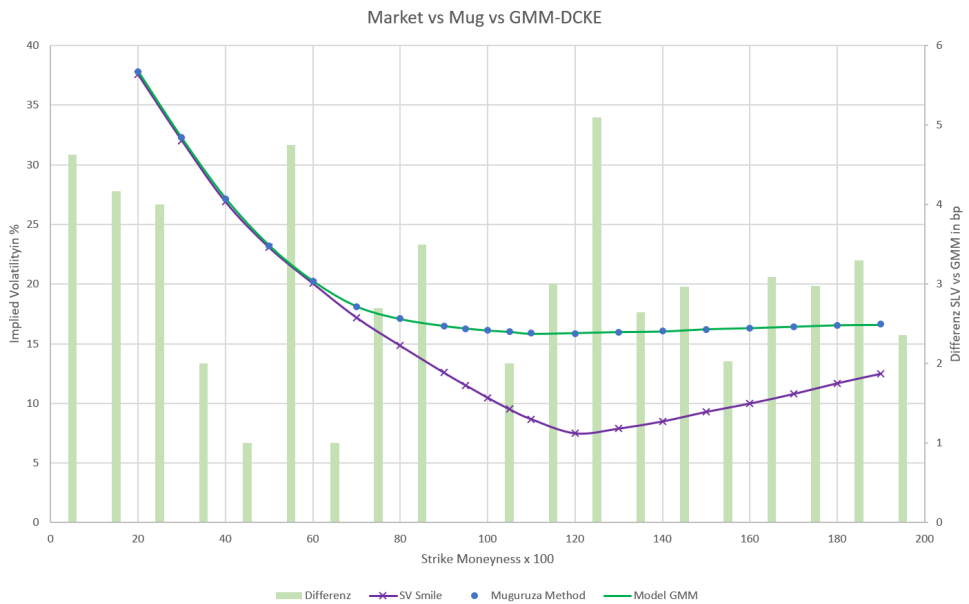



Figure: Implied volatilities for a Heston model serving as the basis smile and calibrated local stochastic volatility models using the method from Muguruza A. (2020), resp. the GMM-DCKE method. On the secondary axis we plotted the differences in bp.

# Observations - Stochastic Local Volatility

- Analytic approximation for conditional expectations for any Stochastic Volatility/rough Stochastic Volatility model
- Model free and data driven - as explained before
- Excellent performance compared to the existing methods which are fully numeric

# xVA/Exposure

- GMM is also a generative model and it can be used to sample realizations from the approximated distribution.
- We consider xVA/Exposure of an Interest Rate Swap (IRS).
- xVA calculation uses a model calibrated to current market observations (spot yield curves, observed implied volatilities), thus, the risk neutral measure.
- For exposure calculation many institutions may wish to use the real world or physical measure, see Stein H. (2016).
- This may lead to two different systems and, thus, increases maintenance and development, thus, costs.
- Main difference: The drift! Risk managers claim that the real world drift is used and mention backtesting as one example.

# xVA/Exposure

- Assume a calibrated model and drift estimates given.
- Calculate the differences (risk neutral and real world forwards).
- Fit some GMM to the risk neutral values.
- Add the differences to the means while keeping the covariance structure and the weights.
- We have chosen the Trolle-Schwartz model with constant parameters and compute the positive and the negative exposure for an 55y IRS with a 6m floating rate.
- For each step we fitted the a GMM and applied GMM-DCKE method to the observed swap rate distribution.
- We shift the mean parameters and calculate the exposure again. The results are displayed in 11.

# xVA/Exposure

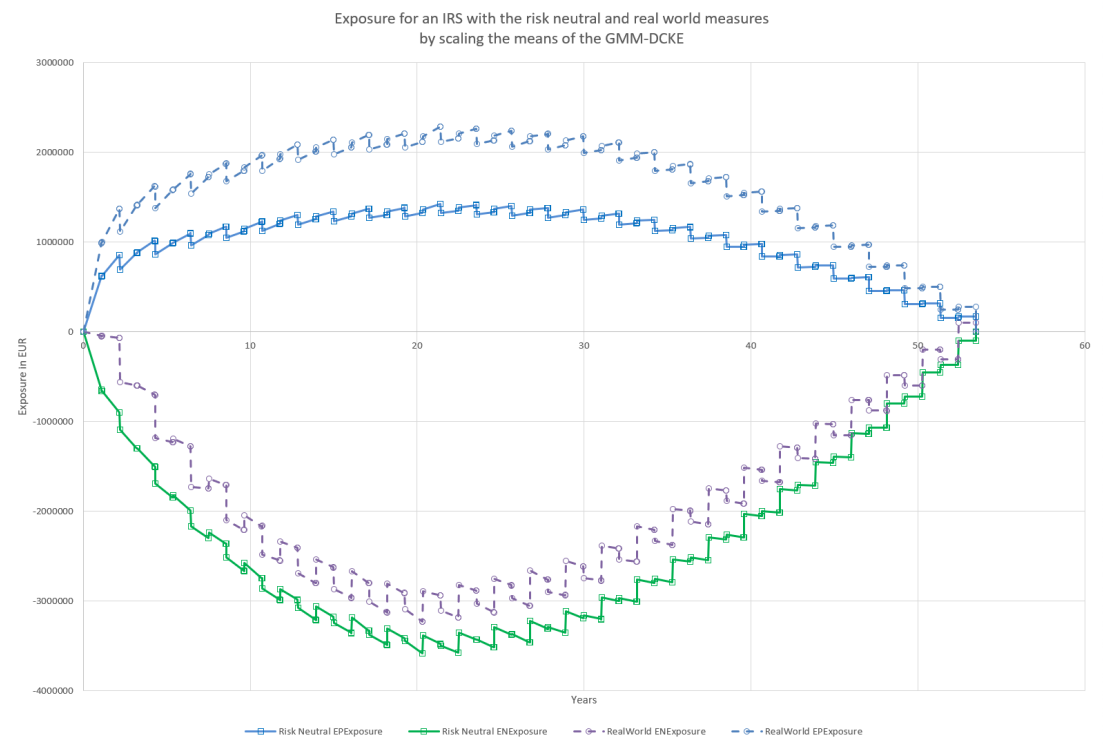



Figure: Positive, resp. negative exposures for an IRS (55y, data from the TS model) compared to the GMM-DCKE at each observed step. Shifting means to reflect differences in drifts (fixed percentage of the risk neutral forwards) but for real cases estimates via time series analysis may be used.

# Observations - xVA/Exposure

- GMM-DCKE is able to fit the exposure observed for simple instruments, need to do this for derivatives like swaptions next
- With GMM-DCKE one can accomodate measure changes by not running another full simulation
- The example only serves for illustration. For practical applicability this approach needs to be tested on a portfolio level including different risk factors and contracts.

# Conclusion, Summary, Future Research

- We specified and applied a model free and data driven method for computing conditional expectations based on GMM.
- Up to fitting a GMM the method is analytic!
- Several examples covering important fields of applications were considered (Bermudans, SLV, xVA/Exposure) but also multi-modal distributions may be considered
- We outlined the problems that may arise (e.g. curse of dimensionality) and considered how to mitigate this
- Strategies for future research:
  - Examination of Transformations,
  - GMM-DCKE for FBSDEs or latent space GMM-DCKE using (Variational) Autoencoders,
  - Efficient calculation of the mixtures.

# Appendix

# Gaussian Distribution

# Gaussian Distribution

Let $d \in \mathbb{N}$, $m \in \mathbb{R}^d$ and $\Sigma$ be a symmetric, positive definite $d \times d$ matrix. The $d$-dimensional multivariate Gaussian distribution has a joint probability density given by

$$p(x|m,\Sigma) = \frac{1}{(2\pi)^{-d/2}}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(x-m)^\top\Sigma(x-m)\right) \tag{13}$$

We call $m$ the *mean* and $\Sigma$ the *covariance* of the Gaussian distribution. A short hand notation for $x$ having a Gaussian distribution with mean $m$ and covariance $\Sigma$ is

$$x \sim \mathbb{N}(m,\Sigma) \tag{14}$$

# Conditional Distribution

Let $Z = (Z_1, \ldots, Z_m, Z_{m+1}, \ldots, Z_{m+n=d})^\top$ be a $d$-dimensional random variable with $d$-dimensional Gaussian distribution, ie. $Z \sim N(\mu, \Sigma)$. Let $X = (Z_1, \ldots, Z_m)^\top$ and $Y = (Z_{m+1}, \ldots, Z_{m+n})^\top$ being $m$, resp. $n$ dimensional random variable we have

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix}.$$

The conditional and marginal distributions for $X$ and $Y$ can be derived from the joint probability distribution of $Z$. To this end let

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix}, \mu_z = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$$

# Marginal Conditional Distribution

The marginal distribution $p$ is given by

$$p(x) = \int p_z(x, y)dy \sim \mathcal{N}(\mu_X, \Sigma_{XX})$$

and the conditional distribution $p_{X|Y}$ is given by

$$p_{x|y}(x) = \frac{p_z(x, y)}{p_z(y)} \sim \mathcal{N}\left(\mu_{X|Y}, \Sigma_{X|Y}\right)$$

with

$$\mu_{X|Y} = \mu_X - \Sigma_{XY}\Sigma_{YY}^{-1}(y - \mu_Y)$$

and

$$\Sigma_{X|Y} = \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}$$

# Multi-dimensional Case

Using these results for each single component $k = 1, \ldots, K$, ie. $X = X_k$ and $Y = (Z_1, \ldots, Z_{k-1}, Z_{k+1}, \ldots, Z_d)^\top$ we have

$$p_k(x|y) = \frac{p_k(x,y)}{p_k(y)} \sim \mathcal{N}\left(\mu_{k,X|Y}, \Sigma_{k,X|Y}\right)$$

and for the Gaussian Mean Mixture

$$p(x|y) = \sum_{k=1}^{K} \pi_k p_k(x|y), \quad \pi_k \sim \frac{\pi_k \mathcal{N}(\mu_{k,Y}, \Sigma_{k,YY})}{\sum_l \pi_l \mathcal{N}(\mu_{l,Y}, \Sigma_{l,YY})}$$

# Another View on Mixtures

Let $U = (U_1, \ldots, U_K)^\top$ with $U_k$, $k = 1, \ldots, K$ denote some random variables taking values in $\{0, 1\}$ and $\sum_{k=1}^K U_k = 1$. The $K$ possible states of $U$ corresponding to $K$ components. $Z$ is the observed and $U$ is the hidden variable.
Denote $\pi_k := \mathbb{P}(U_k = 1)$ with the parameters we used for the Gaussian Mixture model. Since $U$ uses 1 out of $K$ it follows that

$$\mathbb{P}(U) = \prod_{k=1}^K \underbrace{\pi_k \delta_1(u_k)}_{=:\pi_k^{u_k}}$$

We have

$$\mathbb{P}(z|u_k = 1) = N(z|\mu_k, \Sigma_k), \quad \mathbb{P}(z|u) = \prod_{k=1}^K N(z|\mu_k, \Sigma_k)^{u_k}$$

# Another View on Mixtures

The standard form of a GMM can then be obtained by

$$
\begin{aligned}
p(z) &= \sum_u p(u)p(z|u) = \sum_z \prod_{k=1}^{K} \pi_k^{u_k} N(x|\mu_k, \Sigma_k)^{u_k} \\
&= \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)
\end{aligned}
$$

# Another View on Mixtures

Also for hyperparameter estimation the quantities $\gamma_k$, $k = 1, \ldots, K$ are useful:

$$\begin{aligned} \gamma_k &:= \mathbb{P}(z_k = 1|x) = \frac{\mathbb{P}(u_k = 1)\mathbb{P}(x|u_k = 1)}{\sum_{l=1}^{K} \mathbb{P}(u_l = 1)\mathbb{P}(x|u_l = 1)} \\ &= \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l N(x|\mu_l, \Sigma_l)} \end{aligned}$$

The representation considered is the *latent variable representation* of GMM.
$\pi_k = \mathbb{P}(u_k = 1)$ can be viewed as the prior probability of component $k$ and $\gamma(u_k) = \mathbb{P}(u_k = 1|z)$ as the posterior probability. This can directly translated of the probability that component $k$ takes for explaining the observation.

# Covariance

Let $X_i = (X_{i,1}, \ldots, X_{i,d})^\top$ be $i = 1, \ldots, n$ with $X_i \sim \mathcal{N}(\mu_i, \Sigma_i)$ and $w_i > 0$ such that $\sum_{i=1}^n \omega_i = 1$.
For $Z = \sum_{i=1}^K \omega_i X_i$ the covariance of the components of $Z$ can be calculated using $\mathbb{E}[X_{i,k} X_{j,l}]$ with $i, j = 1, \ldots, k$ and $k, l = 1, \ldots, n$.
Let $\sigma_{i,k}$ be the std deviations and $\rho_{i,j,k,l}$ the correlations.
The covariance of $Z_i$ and $Z_j$ is given by:

$$\begin{aligned}
\mathbb{E}[(Z_i - \mu_{Z_i})(Z_j - \mu_{Z_j})] &= \mathbb{E}[Z_i Z_j] - \mathbb{E}[Z_i]\mathbb{E}[Z_j] \\
&= \mathbb{E}\left[\sum_{k=1}^K \omega_k X_{i,k} \sum_{l=1}^K \omega_l X_{j,l}\right] - \mu_{Z_i}\mu_{Z_j} \\
&= \mathbb{E}[\sum_{k,l=1}^K \omega_k \omega_l X_{i,k} X_{j,l}] - \mu_{Z_i}\mu_{Z_j}
\end{aligned}$$

# Covariance

We have

$$\begin{aligned}
\mu_{Z_i} &:= \mathbb{E}[Z_i] = \mathbb{E}\left[\sum_{k=1}^{K} \omega_k X_{i,k}\right] = \sum_{k=1}^{K} \omega_k \mu_{i,k} \\
\mathbb{E}[Z_i Z_j] &= \sum_{k,l=1}^{K} \omega_k \omega_l \mathbb{E}[X_{i,k} X_{j,l}] \\
&= \sum_{k,l=1}^{K} \omega_k \omega_l \sigma_{i,k} \sigma_{j,l} \rho_{i,j,k,l}
\end{aligned}$$

# Local Regression

# Local Regression - LR vs GR

- Basis function regression performs a global fit to the data $(x_i, y_i)$, $i = 1, \ldots, N$
- The resulting function $\tilde{f}$ however, can be evaluated very cheaply for any $x$.
- The drawback is that for some $x$, the fit might be bad.
- In local polynomial regression we specify the $x$ value at which to make predictions and fit to the data that is particularly suited for that $x$.
- If the unknown function $f : \mathbf{R}^d \to \mathbf{R}$ is differentiable consider its Taylor series of order $p$ at $x$ we get for any $x_i$:

$$f(x_i) = \sum_{|\alpha| \leq p} \frac{\partial^\alpha f(x)}{\alpha!}(x - x_i)^\alpha + \mathcal{O}(|x - x_i|^p)$$

# Local Regression - Kernels

- Thus, we replace $f(x_i)$ in (**??**) by its Taylor polynomial.
- The coefficients of the polynomial are not known, but can be obtained by minimizing the weighted cost function

$$J(\beta) := \sum_{i=1}^{N} \left( y_i - \beta^0 - \sum_{1 \leq |\alpha| \leq p} \beta^\alpha (x - x_i)^\alpha \right)^2 w_i. \qquad (15)$$

- Weighted average (Taylor is off for points far from $x$).
- Standard way to choose weights $w_i$ is using a *kernel function* $K : \mathbf{R}^d \to \mathbf{R}$, e.g. the *exponential kernel* $K(z) := \exp(-\frac{1}{2}z^2)$.
- Given *bandwidth* $h \in \mathbf{R}^d$ define $K_h(z) := (\prod_{i=1}^d h_i)^{-1} K(\frac{z_1}{h_1}, \ldots, \frac{z_d}{h_d})$ and set $w_i := K_h(x - x_i)$.

# Local Regression - Special Cases

For $d = 1$, the problem (15) is the solution of

$$\beta^0 = e_0^\top (X^\top W X)^{-1} X^\top y, \text{ where } X_{ij} = (x - x_i)^j, W = \text{diag}(w_1, \ldots, w_N)$$

For $p = 0$ (*Nadaraya-Watson estimator*) or $p = 1$ (*locally linear estimator*), this equation can be explicitly solved. The resulting estimators for $\mathbb{E}[Y|X = x]$ are

$$\hat{\mu}_Y(x; p, h) = \sum_{i=1}^{N} W_i^p(x) y_i, \tag{16}$$

where

$$W_i^0(x) := \frac{K_h(x - x_i)}{\sum_{j=1}^N K_h(x - x_j)}, \quad W_i^1(x) := \frac{1}{N} \frac{\hat{s}_2(x) - \hat{s}_1(x)(x - x_i)}{\hat{s}_2(x) s_0(x) - \hat{s}_1(x)^2} K_h(x - x_i).$$

with $\hat{s}_r(x) := \frac{1}{N} \sum_{i=1}^N (x - x_i)^r K_h(x - x_i)$.

# Gaussian Process Regression

# Bayesian Regression

Let $x \in \mathbb{R}^d$ be an input vector and $w \in \mathbb{R}^d$ a vector of weights. The basis function regression model with Gaussian noise is

$$f(x) = \varphi(x)^\top w \tag{17}$$

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{18}$$

We call $y$ in (18) the observed target values. To the approximation $f(\cdot)$ we add a Gaussian noise $\epsilon$ accounting for the fact that $y$ differ from the approximation values $f(x)$ in (17).
**Remark:** $\varphi = (\varphi_1, \ldots, \varphi_d)$ is a collection of basis functions.
In *frequentist* regression we aim to minimize $\epsilon$ and not assuming a distribution for the error.
For the considerations here see Rasmussen, C. and William C. (2006).

# The Prior and Posterior

Let us assume some uncertainty on the parameters $w$, e.g.:

$$w \sim \mathcal{N}(0, \Sigma_p)$$

We call this distribution the *Prior Distribution*.

The *posterior distribution* is now the likelihood times the prior divided by the marginal likelihood (Bayes Theorem)

$$\text{Posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

or, formally,

$$p(w|Y, X) = \frac{p(Y|X, w)p(w)}{p(Y|X)}$$

# A Gaussian Process

A *Gaussian Process* is a collection of random variables for which any finite number of these have a joint Gaussian distribution. For the regression we are interested in $f(x)$ for a random variable $x$. Let us assume

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)] && (19) \\ \mathrm{CoV}(x,y) &:= k(x,y) && (20) \\ &= \mathbb{E}[(f(x)-m(x))(f(y)-m(y))] \end{aligned}$$

We write

$$f(x) \sim \mathcal{GP}(m,k) \qquad (21)$$

**Remark:** Often $m = 0$ but this assumption can be relaxed!

# How does it work

A Gaussian process helps to formulate the regression we considered earlier in a different way. This is in terms of *function spaces*.

- Choose prior distribution (on functions) and covariance/kernel
- Choose/Optimize wrt log-likelihood the hyperparameters values
- Combine prior with observation to create the posterior distribution (on functions)
- Calculate notions of interest quantiles, MAP (mean of posterior is called *maximum a posterior-MAP* estimate!)

# Example - Noise Free / Noisy Observations

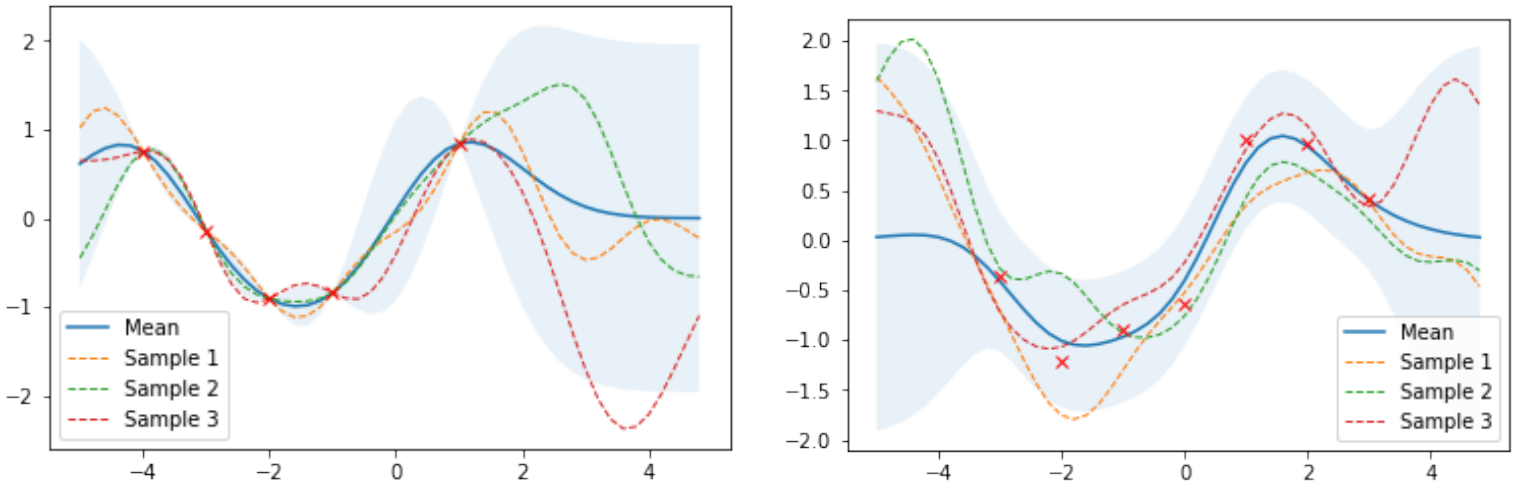



Figure: Posterior Samples without (left)/with (right) uncertainty.

# Example: Using the matrices

The MAP estimator can be calculated since we can use

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K & K_*^\top \\ K_* & K_{**} \end{pmatrix}\right)$$

Thus, we have for $K = k(x_i, x_j)_{i,j}$, $K^* := k(x_{*,i}, x_j)_{i,j}$ and $K^{**} = k(x_{*,i}, x_{*,j})_{i,j}$

$$f_*|f \sim \mathcal{N}(\underbrace{K_* K^{-1} f}_{\bar{f}_* := \mathbb{E}[f_*|f]}, \underbrace{K_{**} - K_* K^{-1} K_*^\top}_{CoV(f^*) := CoV[f_*|f]})$$

The predictive distribution in the setting of GP is

$$\begin{aligned} f_*|X, Y, X_* &\sim \mathcal{N}(\bar{f}_*, \mathrm{CoV}(f_*)) && (22) \\ \bar{f}_* &= \mathbb{E}[f_*|X, Y, X_*] = K_*(K + \sigma_n^2 I)^{-1} Y && (23) \\ \mathrm{CoV}(f_*) &= K_{**} - K_*(K + \sigma_n^2 I)^{-1} K_*^\top && (24) \end{aligned}$$

# Example: Squared Exponential (SE) and ARDSE

The *squared exponential* or *radial basis* covariance is istotropic, stationary, i.e. $r = |x - y|$ with hyperparameter $l$ and is given by

$$k_{\text{SE}}(x, y) = \sigma^2 \exp\left(-\frac{r^2}{2l^2}\right) \tag{25}$$

$l$ is called the *(characteristic) length scale*.
The *automatic relevance determination squared exponential* covariance is given by

$$k_{\text{ARDSE}}(x, y) = \sigma_0^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - y_i}{l_i}\right)^2\right) \tag{26}$$

The parameters $l_i$, $i = 1, \ldots, d$ are the individual length scales. $l_i$ determines the relevance of parameter $i$. Large $l_i$ renders the $i$-th parameter irrelevant

# Trolle-Schwartz Model

# The Trolle-Schwartz (TS) Model

We now take the Trolle-Schwartz model. This model is specified in terms of the instantaneous forward rate $f$:

$$f(t,T) = f(0,T) + \sum_{i=1}^{N} B_{x_i}(T-t)x_i(t) + \sum_{i=1}^{N}\sum_{j=1}^{6} B_{\phi_{j,i}}(T-t)\phi_{j,i}(t)$$

where the stochastic drivers $x_i$ and $v_i$ are modeled by

$$\begin{aligned} dx_i &= -\gamma_i x_i(t)dt + \sqrt{v_i(t)}dW_i(t) \\ dv_i(t) &= \kappa_i(\theta_i - v_i)dt + \sigma_i\sqrt{v_i(t)}dB(t) \end{aligned}$$

with $B_i$ correlated to $W_i$ according to $\rho_i$ and $\alpha_i := \frac{\alpha_{1,i}}{\gamma_i}\left(\frac{1}{\gamma_i} + \frac{\alpha_{0,i}}{\alpha_{1,i}}\right)$, $\beta_i := \frac{\alpha_{1,i}}{\gamma_i} + 2\alpha_{0,i}$. For this model the zero bond prices are given by

$$P(t,T) = \frac{P^M(0,T)}{P^M(0,t)} \exp\left(\sum_{i=1}^{N} B_{x_i}(T-t)x_i(t) + \sum_{j}\sum_{i} B_{\phi_{i,j}}(T-t)\phi_{j,i}(t)\right)$$

# The Trolle-Schwartz (TS) Model

We use (with the coeffs via solving an ODE system)

$$
\begin{aligned}
B_{x_i}(t) &= (\alpha_{0,i} + \alpha_{1,i} t) e^{-\gamma_i t} \\
B_{\phi_{1,i}}(t) &= \alpha_{1,i} e^{-\gamma_i t} \\
B_{\phi_{2,i}}(t) &= \alpha_i (\alpha_{0,i} + \alpha_{1,i} t) e^{-\gamma_i t} \\
B_{\phi_{3,i}}(t) &= -\left( \alpha_{0,i}\alpha_i + \frac{\alpha_{1,i}}{\gamma_i}\beta_i t + \frac{\alpha_{1,i}^2}{\gamma_i} t^2 \right) e^{-2\gamma_i t} \\
B_{\phi_{4,i}}(t) &= \alpha_{1,i}\alpha_i e^{-\gamma_i t} \\
B_{\phi_{5,i}}(t) &= -\frac{\alpha_{1,i}}{\gamma_i} \left( \beta_i + 2\alpha_{1,i} t \right) e^{-2\gamma_i t} \\
B_{\phi_{6,i}}(t) &= -\frac{\alpha_{1,i}^2}{\gamma_i} e^{-2\gamma_i t}
\end{aligned}
$$

$$
\begin{aligned}
d\phi_{1,t} &= \left( x_i(t) - \gamma_i \phi_{1,i}(t) \right) dt; \\
d\phi_{2,t} &= \left( v_i(t) - \gamma_i \phi_{2,i}(t) \right) dt; \\
d\phi_{3,t} &= \left( v_i(t) - 2\gamma_i \phi_{3,i}(t) \right) dt; \\
d\phi_{4,t} &= \left( \phi_{2,i}(t) - \gamma_i \phi_{4,i}(t) \right) dt; \\
d\phi_{5,t} &= \left( \phi_{3,i}(t) - 2\gamma_i \phi_{5,i}(t) \right) dt; \\
d\phi_{6,t} &= \left( 2\phi_{5,i}(t) - 2\gamma_i \phi_{6,i}(t) \right) dt
\end{aligned}
$$

Bates, D. Jumps and stochastic volatility, exchange rate processes implicit in Deutsche mark options. *Review of Financial Studies*, 9:69–107, 1996.

Bayer, C., Friz, P., and Gatheral, J. Pricing under rough volatility. *Quantitative Finance*, 16(6):887–904, 2016.

Bishop C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Dempster A. P., Larid N. M., Rubin D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B*, 39 (1):1–38, 1977.

Dupire, B. Pricing with a Smile. *RISK*, 7:18–20, 1994.

Geng Q., Kienitz J., Lee G. T., Nowaczyk N. Dynamically controlled kernel estimation. *RISK*, 2, 2022.

Guyon J., Henry-Labordere P. Being particular about calibration. *RISK*, 25(1), 2009.

Halperin, I. Qlbs: Q-learner in the black-scholes (-merton) worlds. *SSRN*, 12 2017. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3087076`.

Henry-Labordere P. Calibration of local stochastic volatility models to market smiles: A monte carlo approach. *RISK*, No. 9, 2009.

Heston, S. A closed form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Fin. Studies*, 6:327–343, 1993.

Huge, B. N., Savine, A. Differential machine learning: The shape of things to come. *RISK*, 2020.

Madan, D. and Seneta, E. The V.G. model for share market returns. *J. Business*, 63:511–524, 1990.

Muguruza A. Not so particular abut calibration: Smile problem resolved. *SSRN*, 2020. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3461545`.

Marc Potters, Jean-Philippe Bouchaud, and Dragan Sestovic. Hedged monte-carlo: low variance derivative pricing with objective probabilities. *Physica A: Statistical Mechanics and its Applications*, 289(3-4):517–525, 2001.

Potters, M., Sestovic, D., Bouchaud, J.-P. Hedge your monte carlo. *RISK*, 14:3:133–136, 2001.

Rasmussen, C. and William C. *Gaussian Process for Machine Learning*. MIT Press, 2006.

Stein H. Fixing risk neutral risk measures. *International Journal of Theoretical and Applied Finance*, 19 (3), 2016.

van der Stoep A., Grzelak L. A., Oosterlee, C. W. . The heston stochastic-local volatility model: Efficient monte carlo simulation. *International Journal of Theoretical and Applied Finance)*, 17, No. 7, 2014.