Formulation and Stochastic Galerkin Methods for Stochastic Partial Differential Equations III

Hermann G. Matthies

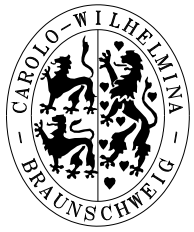
with Andreas Keese

Institute of Scientific Computing

Technische Universität Braunschweig, Germany

wire@tu-bs.de

http://www.wire.tu-bs.de



Repetition of Second Summary

Stochstic Galerkin methods work.

Galerkin procedure is numerically stable $\Rightarrow$ convergence.

Convergence rates seemingly only with regularity.

Stochastic calculations produce huge amounts of data, which is expensive to operate on and to store.

Results a priori live in very high dimensional spaces.

They have a natural tensor product structure.

Overview III

1. Approximation Theory
2. Computations for Stochastic Galerkin (linear)
3. Computations for Stochastic Galerkin (non-linear)
4. Stochastic Collocation for PCE
5. Time Evolution Problems
6. Sparse Tensor Product Approximation

Approximation Theory

- Stability of discrete approximation under truncated KLE and PCE. Matrix stays uniformly positive definite.
- Convergence follows from Céa's Lemma.
- Convergence rates under stochastic regularity in stochastic Hilbert spaces—stochastic regularity theory?.
- Error estimation via dual weighted residuals possible.

Theorem: Let $p > 0$, $r > 1$, $|\rho| \leq 1$. Let $P_{k,m}$ be projection onto $\mathscr{S}_{k,m}$.

$$\forall R \in (\mathcal{S})^{\rho,-p} \quad \|R - P_{k,m}(R)\|^2_{\rho,-p} \leq \|R\|^2_{\rho,-p+r}\, c(m,k,r)^2,$$

where $c(m,k,r)^2 = c_1(r)m^{1-r} + c_2(r)2^{-kr}$. But $\dim \mathscr{S}_{k,m}$ grows too quickly with k and m. Sparser spaces and error estimates needed.

Equations for Linear Stochastic Galerkin

In what follows we use orthonormal $\hat{H}_\alpha(\boldsymbol{\theta}) = H_\alpha(\boldsymbol{\theta})/\sqrt{\alpha!}$ instead of $H_\alpha(\boldsymbol{\theta})$. With $\boldsymbol{f}^{(\gamma)} := \mathbb{E}\left(\hat{H}_\gamma f(\boldsymbol{\theta})\right)$ and $\mathbf{f} = [\boldsymbol{f}^{(0)}, \ldots, \boldsymbol{f}^{(\gamma)}, \ldots, \boldsymbol{f}^{(\lambda)}]$:

$$\sum_\alpha \sum_\beta \sum_\jmath \xi_\jmath^{(\alpha)} \underbrace{\mathbb{E}\left(\hat{H}_\alpha \hat{H}_\beta \hat{H}_\gamma\right)}_{=:\Delta_{\beta,\gamma}^{(\alpha)}} \underbrace{\int \nabla \boldsymbol{N}(x) \varkappa_\jmath \, g_\jmath(x) \nabla \boldsymbol{N}(x)^T \, dx}_{\boldsymbol{K}_\jmath} \boldsymbol{u}^{(\beta)} = \boldsymbol{f}^{(\gamma)}$$

$$\mathbb{E}\left(\hat{H}_\alpha \hat{H}_\beta \hat{H}_\gamma\right) = \Delta_{\beta,\gamma}^{(\alpha)} = C_{\beta\gamma}^{(\alpha)} \sqrt{\frac{\alpha!}{\beta!\gamma!}}$$

Equations and $\mathbf{u}, \mathbf{f}$ have tensor product structure (storage and use).

$$\mathbf{K}\,\mathbf{u} = \sum_\jmath \sum_\alpha \xi_\jmath^{(\alpha)} \Delta^{(\alpha)} \otimes \boldsymbol{K}_\jmath \, \mathbf{u} = \mathbf{f}$$

Computations for Linear Iterative Solution

$$\mathbf{K}\,\mathbf{u} = \left[\sum_{\jmath}\sum_{\alpha} \xi_{\jmath}^{(\alpha)} \Delta^{(\alpha)} \otimes \boldsymbol{K}_{\jmath}\right] \mathbf{u} = \mathbf{f}$$

With preconditioner $\mathbf{P} = \boldsymbol{I} \otimes \boldsymbol{K}_0$ a simple iteration is

$$^{k+1}\mathbf{u} = {}^{k}\mathbf{u} + \mathbf{P}^{-1}(\mathbf{f} - \mathbf{K}\,{}^{k}\mathbf{u}).$$

Of course $\mathbf{P}^{-1}$ is only a symbolic notation.

Which means for 'material' $\varkappa_{\jmath} g_{\jmath}(x)$ and each β compute the 'residuum' $^{k}_{\jmath}\boldsymbol{w}^{(\beta)} := \boldsymbol{K}_{\jmath}\,{}^{k}\boldsymbol{u}^{(\beta)}$ and then $^{k+1}\mathbf{u} = {}^{k}\mathbf{u} + {}^{k}\mathbf{v}$ with

$$^{k}\boldsymbol{v}^{(\gamma)} = \boldsymbol{K}_0^{-1}(\boldsymbol{f}^{(\gamma)} - \sum_{\alpha,\beta,\jmath} {}^{k}_{\jmath}\boldsymbol{w}^{(\beta)} \xi_{\jmath}^{(\alpha)} \Delta_{\beta,\gamma}^{(\alpha)})$$

This is one 'preconditioner solve' of the deterministic code.

Interlude: Computation of Moments from PCE

Discrete version of PCE of $\boldsymbol{u}(\boldsymbol{\theta})$:

$$\boldsymbol{u}(\boldsymbol{\theta}) = \bar{\boldsymbol{u}} + \tilde{\boldsymbol{u}}(\boldsymbol{\theta}) = \bar{\boldsymbol{u}} + \sum_{\gamma \neq 0} \boldsymbol{u}^{(\gamma)} \hat{H}_\gamma(\boldsymbol{\theta})$$

Let $\boldsymbol{M}_u^{(k)} = \mathbb{E}\left(\overbrace{\tilde{\boldsymbol{u}}(\boldsymbol{\theta}) \otimes \ldots \otimes \tilde{\boldsymbol{u}}(\boldsymbol{\theta})}^{k \text{ times}} \right) = \mathbb{E}\left(\tilde{\boldsymbol{u}}^{\otimes k} \right)$, totally symmetric,

especially $\boldsymbol{M}_u^{(1)} = \bar{\boldsymbol{u}}$, $\boldsymbol{M}_u^{(2)} = \boldsymbol{C}_u = \sum_{\gamma \neq 0} \boldsymbol{u}^{(\gamma)} \otimes \boldsymbol{u}^{(\gamma)}$.

It results that multi-point correlation

$$\boldsymbol{M}_u^{(k)} = \mathbb{E}\left(\tilde{\boldsymbol{u}}^{\otimes k} \right) = \sum_{\gamma_{(1)}, \ldots, \gamma_{(k)} \neq 0} \mathbb{E}\left(\prod_{j=1}^{k} \hat{H}_{\gamma_{(j)}}(\boldsymbol{\theta}) \right) \boldsymbol{u}^{(\gamma_{(1)})} \otimes \ldots \otimes \boldsymbol{u}^{(\gamma_{(k)})}.$$

Non-Linear Equations

Example: Use $\kappa(x,u,\omega) = a(x,\omega) + b(x,\omega)u^2$, and a, b random.
Space discretisation generates a non-linear equation
$\boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{u}(\boldsymbol{\theta})) = \boldsymbol{A}(\boldsymbol{\theta}, \sum_\beta \boldsymbol{u}^{(\beta)} \hat{H}_\beta(\boldsymbol{\theta})) = \boldsymbol{f}(\boldsymbol{\theta})$. Projection onto PCE:

$$\mathbf{a}[\mathbf{u}] = [\ldots, \mathbb{E}\left(\hat{H}_\alpha(\boldsymbol{\theta}) \boldsymbol{A}(\boldsymbol{\theta}, \sum_\beta \boldsymbol{u}^{(\beta)} \hat{H}_\beta(\boldsymbol{\theta}))\right), \ldots] = \mathbf{f} = \boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{\Phi}^T$$

Expressions in $\mathbf{a}$ need high-dimensional integration (in each iteration),
e.g. Monte Carlo or Smolyak (sparse grid) quadrature:
$\mathbf{a}^{(\alpha)} = \mathbb{E}\left(\hat{H}_\alpha(\boldsymbol{\theta}) \boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{u}(\boldsymbol{\theta}))\right) \approx \sum_{z=1}^{Z} w_z \hat{H}_\alpha(\theta_z) \boldsymbol{A}(\theta_z, \boldsymbol{u}(\theta_z))$

The residual equation to be solved is

$$\mathbf{r}(\mathbf{u}) := \mathbf{f} - \mathbf{a}[\mathbf{u}] = 0.$$

Solution of Non-Linear Equations

Assume solver for deterministic problem: $\boldsymbol{r}(\boldsymbol{u}) = 0$:

$$^{k+1}\boldsymbol{u} = {}^{k}\boldsymbol{u} + {}^{k}\boldsymbol{w} = {}^{k}\boldsymbol{u} + S({}^{k}\boldsymbol{u}, \boldsymbol{r}({}^{k}\boldsymbol{u})) =: T({}^{k}\boldsymbol{u}, \boldsymbol{r}({}^{k}\boldsymbol{u}))$$

Then stochastic Galerkin non-linear iteration may be

$$^{k+1}\mathbf{u} = {}^{k}\mathbf{u} + {}^{k}\mathbf{w} = {}^{k}\mathbf{u} + \mathbf{S}({}^{k}\mathbf{u}, \mathbf{r}({}^{k}\mathbf{u})) =: \mathbf{T}({}^{k}\mathbf{u}, \mathbf{r}({}^{k}\mathbf{u}))$$

with

$$\mathbf{S}({}^{k}\mathbf{u}, \mathbf{r}({}^{k}\mathbf{u})) = [S({}^{k}\boldsymbol{u}^{(0)}, \boldsymbol{r}^{(0)}({}^{k}\mathbf{u})) \ldots, S({}^{k}\boldsymbol{u}^{(\beta)}, \boldsymbol{r}^{(\beta)}({}^{k}\mathbf{u})), \ldots]$$

The only interaction with the deterministic solver is

- computing residua for realisations θ_z
- using iteration for those residua.

Solution of Non-Linear Equations II

A quasi-Newton method may accelerate convergence

$$ {}^{k+1}\mathbf{u} = {}^{k}\mathbf{u} + {}^{k}\mathbf{w}, \quad {}^{k}\mathbf{w} = \mathbf{H}_k(\mathbf{r}({}^{k}\mathbf{u})) $$

$$ \mathbf{H}_k(\mathbf{r}) = \mathbf{S}({}^{k}\mathbf{u}, \mathbf{r}) + \sum_{j=1}^{k} (a_j \mathbf{p}_j \mathbf{p}_j^T \mathbf{r} + b_j \mathbf{q}_j \mathbf{q}_j^T \mathbf{r}) $$

Tensors $\mathbf{p}$ and $\mathbf{q}$ computed from residuum and last increment. Notice tensor products of (hopefully sparse) tensors.

Needs pre-conditioner $\mathbf{S}$ for good convergence: May use linear solver as described before, i.e.

$$ \mathbf{S} = \boldsymbol{I} \otimes \boldsymbol{K}_0. $$

Stochastic Collocation

As the $\{\hat{H}_\alpha(\boldsymbol{\theta})\}$ are an orthonormal basis, the PCE of

$$\boldsymbol{u}(\boldsymbol{\theta}) = \sum_\beta \boldsymbol{u}^{(\beta)} \hat{H}_\beta(\boldsymbol{\theta})$$

may be formally computed from simple projections:

$$\boldsymbol{u}^{(\beta)} = \langle \hat{H}_\beta(\boldsymbol{\theta}), \boldsymbol{u}(\boldsymbol{\theta}) \rangle = \mathbb{E}\left(\hat{H}_\beta(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{\theta})\right) =$$

$$\mathbb{E}\left(\hat{H}_\beta(\boldsymbol{\theta}) \sum_\alpha \boldsymbol{u}^{(\alpha)} \hat{H}_\alpha(\boldsymbol{\theta})\right) = \sum_\alpha \boldsymbol{u}^{(\alpha)} \mathbb{E}\left(\hat{H}_\beta(\boldsymbol{\theta})\hat{H}_\alpha(\boldsymbol{\theta})\right) = \sum_\alpha \boldsymbol{u}^{(\alpha)} \delta_{\alpha,\beta}.$$

Hence $\boldsymbol{u}^{(\beta)} = \mathbb{E}\left(\hat{H}_\beta(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{\theta})\right) \approx \sum_{z=1}^{Z} w_z \hat{H}_\beta(\theta_z)\boldsymbol{u}(\theta_z).$

This only needs sample solutions $\boldsymbol{u}(\theta_z)$ and works both for linear and non-linear problems.

Work Count

Stochastic collocation faces same
stability problems as was previously explained for Monte Carlo.

Assume N deterministic and M stochastic variables,
i.e. $\mathbf{u}$ is $N \times M$.

Assume that for the stochastic Galerkin method we need J iterations,
then the total work is $Z \cdot J$ residua $+M \cdot J$ iterations.

Assume that for the stochastic collocation method we need I iterations
on average, then the total work is $Z \cdot I$ residua $+Z \cdot I$ iterations.

Often J slightly larger than I, but Z is much larger than M.
Often iterations cost more than residua.

Time Dependent Problems

For a time dependent problem with $u \in \mathcal{U}$ and $f \in \mathcal{F}$ (usually $= \mathcal{U}^*$)

$$\dot{u} + A[u] = f(t),$$

if now either the (possibly non-linear) operator or the rhs f are stochastic, the stochastic solution may be sought in a space $\mathcal{U} \otimes \mathcal{S}$

If this evolution problem is normally space-discretised via $u(x,t) = \sum_k u_k(t) N_k(x)$ to give ODEs for the $[\ldots, u_k(t), \ldots]^T = \boldsymbol{u}(t)$:

$$\dot{\boldsymbol{u}} + \boldsymbol{A}[\boldsymbol{u}] = \boldsymbol{f}(t),$$

the stochastic ansatz $\boldsymbol{u}(t,\boldsymbol{\theta}) = \sum_\alpha \boldsymbol{u}^{(\alpha)}(t) \hat{H}_\alpha(\boldsymbol{\theta})$ gives for $\mathbf{u}(t) = [\ldots, \boldsymbol{u}^{(\alpha)}(t), \ldots]$ many more ODEs:

$$\dot{\mathbf{u}} + \mathbf{a}[\mathbf{u}] = \mathbf{f}(t),$$

What is Wrong with the pure PCE?

Remember already for data field $\kappa(x, \omega)$
a KLE was additionally used to have fewer terms.

For a solution $\mathbf{u} = [\ldots, \boldsymbol{u}^{(\alpha)}, \ldots]$ there are too many PCE terms,
all stochastically relevant information is encoded optimally in
Karhunen-Loève expansion (KLE).

When only a few vectors $\boldsymbol{u}_\ell$ are needed in the end,
why compute all the $\boldsymbol{u}^{(\alpha)}$ in between?

Stochastic Galerkin may be adapted (this is a kind of model reduction)
to work on much less information.

Simpler Case of Additive Noise

Assume first that $\boldsymbol{K}$ is not random: $\forall \gamma \in \mathcal{J}_{k,m}$ satisfy:

$$\sum_{\beta \in \mathcal{J}_{k,m}} \mathbb{E}\left(\hat{H}_\gamma(\boldsymbol{\theta})\hat{H}_\beta(\boldsymbol{\theta})\right)\boldsymbol{K}\boldsymbol{u}^{(\beta)} = \boldsymbol{K}\boldsymbol{u}^{(\gamma)} = \mathbb{E}\left(\boldsymbol{f}(\boldsymbol{\theta})H_\gamma(\boldsymbol{\theta})\right) =: \boldsymbol{f}^{(\gamma)}$$

There are too many $\boldsymbol{f}^{(\gamma)}$. Use discrete version of KLE of $\boldsymbol{f}(\boldsymbol{\theta})$:

$$\boldsymbol{f}(\boldsymbol{\theta}) = \bar{\boldsymbol{f}} + \sum_\ell \varphi_\ell\, \phi_\ell(\boldsymbol{\theta})\boldsymbol{f}_\ell = \bar{\boldsymbol{f}} + \sum_\ell \sum_\gamma \varphi_\ell\, \phi_\ell^{(\gamma)} \hat{H}_\gamma(\boldsymbol{\theta})\boldsymbol{f}_\ell.$$

In particular $\boldsymbol{f}^{(\gamma)} = \sum_\ell \varphi_\ell\, \phi_\ell^{(\gamma)} \boldsymbol{f}_\ell$. The SVD of $\boldsymbol{f}(\boldsymbol{\theta})$ is with $\boldsymbol{\Phi} = (\phi_\ell^{(\gamma)})$, $\boldsymbol{\varphi} = \mathrm{diag}(\varphi_\ell)$, and $\boldsymbol{F} = [\ldots, \boldsymbol{f}_\ell, \ldots]$;

$$\mathbf{f} = [\ldots, \boldsymbol{f}^{(\gamma)}, \ldots] = \boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{\Phi}^T = \sum_\ell \varphi_\ell\, \boldsymbol{v}_\ell \boldsymbol{\phi}_\ell^T.$$

Solution for Additive Noise

Observe $\boldsymbol{K}\bar{\boldsymbol{u}} = \bar{\boldsymbol{f}}$, and set $\boldsymbol{V} := \boldsymbol{K}^{-1}\boldsymbol{F}$ (i.e. $\forall \ell$ solve $\boldsymbol{K}\boldsymbol{v}_\ell = \boldsymbol{f}_\ell$),
then $\boldsymbol{u}(\boldsymbol{\theta}) = \bar{\boldsymbol{u}} + \boldsymbol{V}\boldsymbol{\varphi}\boldsymbol{\Phi}^T\boldsymbol{H} = \bar{\boldsymbol{u}} + \sum_\ell \varphi_\ell \boldsymbol{v}_\ell \boldsymbol{\phi}_\ell^T \boldsymbol{H}$

But this is not the SVD of $\boldsymbol{u}(\boldsymbol{\theta})$! This via eigenproblem:
Covariance $\boldsymbol{C}_f := \mathbb{E}\left(\tilde{\boldsymbol{f}}(\boldsymbol{\theta}) \otimes \tilde{\boldsymbol{f}}(\boldsymbol{\theta})\right) = \boldsymbol{F}\boldsymbol{\varphi}^2\boldsymbol{F}^T$. Hence
$\boldsymbol{C}_u := \mathbb{E}\left(\tilde{\boldsymbol{u}}(\boldsymbol{\theta}) \otimes \tilde{\boldsymbol{u}}(\boldsymbol{\theta})\right) = \boldsymbol{K}^{-1}\boldsymbol{F}\boldsymbol{\varphi}^2(\boldsymbol{K}^{-1}\boldsymbol{F})^T = \boldsymbol{V}\boldsymbol{\varphi}^2\boldsymbol{V}^T$

Even fewer terms needed with SVD of $\boldsymbol{u}(\boldsymbol{\theta})$ from
$\boldsymbol{C}_u\boldsymbol{U} = (\boldsymbol{V}\boldsymbol{\varphi}^2\boldsymbol{V}^T)\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\upsilon}^2$

Sparsification achieved for $\mathbf{u}$ via SVD with small m:

$$\mathbf{u} = \bar{\boldsymbol{u}} + [\ldots, \boldsymbol{u}_m]\,\mathrm{diag}(\upsilon_m)(y_m^{(\beta)})^T\boldsymbol{H} = \bar{\boldsymbol{u}} + \boldsymbol{U}\boldsymbol{\upsilon}\boldsymbol{Y}^T\boldsymbol{H},$$

with $\boldsymbol{Y}^T :=$ truncate $(\boldsymbol{\upsilon}\boldsymbol{U}^T\boldsymbol{V}\boldsymbol{\varphi}^{-1}\boldsymbol{\Phi}^T)$.

Example: Computation of Moments

Let $\boldsymbol{M}_f^{(k)} = \mathbb{E}\left(\overbrace{\tilde{\boldsymbol{f}}(\omega) \otimes \ldots \otimes \tilde{\boldsymbol{f}}(\omega)}^{k \text{ times}}\right) = \mathbb{E}\left(\tilde{\boldsymbol{f}}^{\otimes k}\right)$, totally symmetric,

and $\boldsymbol{M}_f^{(1)} = \bar{\boldsymbol{f}}$, $\boldsymbol{M}_f^{(2)} = \boldsymbol{C}_f$.

KLE of $\boldsymbol{C}_f$ is $\boldsymbol{C}_f = \sum_\ell \varphi_\ell \, \boldsymbol{f}_\ell \boldsymbol{f}_\ell^T = \sum_\ell \varphi_\ell \, \boldsymbol{f}_\ell \otimes \boldsymbol{f}_\ell$.

For deterministic operator $\boldsymbol{K}$, just compute $\boldsymbol{K}\boldsymbol{v}_\ell = \boldsymbol{f}_\ell$, and then

$\boldsymbol{K}\bar{\boldsymbol{u}} = \bar{\boldsymbol{f}}$, and $\boldsymbol{M}_u^{(2)} = \boldsymbol{C}_u = \mathbb{E}\left(\tilde{\boldsymbol{u}}^{\otimes k}\right) = \sum_\ell \varphi_\ell \, \boldsymbol{v}_\ell \otimes \boldsymbol{v}_\ell$.

As $\tilde{\boldsymbol{u}}(\boldsymbol{\theta}) = \sum_\ell \sum_\alpha \varphi_\ell \, \phi_\ell^{(\alpha)} H_\alpha(\boldsymbol{\theta}) \boldsymbol{v}_\ell$, it results that multi-point correlation

$$\boldsymbol{M}_u^{(k)} = \sum_{\ell_1 \le \ldots \le \ell_k} \prod_{m=1}^{k} \varphi_{\ell_m} \sum_{\alpha_{(1)},\ldots,\alpha_{(k)}} \prod_{n=1}^{k} \phi_{\ell_m}^{(\alpha_{(n)})} \mathbb{E}\left(H_{\alpha_{(1)}} \cdots H_{\alpha_{(k)}}\right) \boldsymbol{v}_{\ell_1} \otimes \ldots \otimes \boldsymbol{v}_{\ell_k}.$$

Sparsification

Goal: As with additive noise, compute only with $\boldsymbol{f}_\ell$, $\boldsymbol{u}_\ell$ from SVD.

Start iteration with tensor product of low rank L.

$$\mathbf{f} = [\ldots, \boldsymbol{f}^{(\gamma)}, \ldots] = \boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{\Phi}^T = \sum_\ell \varphi_\ell \, \boldsymbol{v}_\ell \boldsymbol{\phi}_\ell^T.$$

At each iteration k, rank of iterative approximation $\mathbf{u}_k$ will increase by number of terms in matrix sum.

In each iteration, perform a SVD of $\mathbf{u}_k$ and reduce rank again to L.

Resulting iteration converges to SVD = discrete KLE of $\mathbf{u}$.

Operation Count

$$\mathbf{K}\,\mathbf{u} = \sum_{\jmath} \sum_{\alpha} \xi_{\jmath}^{(\alpha)} \Delta^{(\alpha)} \otimes \boldsymbol{K}_{\jmath}\,\mathbf{u} = \mathbf{f}$$

Assume sum in $\mathbf{K}$ has K terms, $\mathbf{u}$ and $\mathbf{f}$ have size $N \times M$, and $\mathbf{f} = [\ldots, \boldsymbol{f}^{(\gamma)}, \ldots] = \boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{\phi}^T = \sum_\ell \varphi_\ell \boldsymbol{v}_\ell \boldsymbol{\phi}_\ell^T$ has $L \ll N, L \ll M$ terms.

Each application of $\mathbf{K}$ on full $\mathbf{u}_k$ needs $K \times M$ $\boldsymbol{K}$-multiplications plus $K \times N$ Δ-multiplications.

Each application of $\mathbf{K}$ on low rank tensor product $\mathbf{u}_k$ needs $K \times L$ $\boldsymbol{K}$-multiplications plus $K \times L$ Δ-multiplications, which is much less.

Storage is reduced from $N \times M$ to $L \times (N + M)$.

Integration Rules

For Monte Carlo (MC), the points θ_z are random according to measure Γ, and weights are $w_z = 1/Z$.

For Quasi Monte Carlo (QMC), the points θ_z are non-random according to number-theoretic low discrepancy series , and weights are still $w_z = 1/Z$.

For Product Gauss (full tensor product) rules the points θ_z and weights w_z come from one dimensional rules.

For sparse grid Smolyak rules, the points θ_z and weights w_z are known, they come from combination of different rules in different dimensions.

High-Dimensional Integration

We want $\mathbb{E}\left(\psi(\boldsymbol{\theta})\right) = \int_{\Theta_m} \psi(\boldsymbol{\theta}_m)\, d\Gamma_m(\boldsymbol{\theta}_m) =$

$$\int_{\theta_1} \cdots \int_{\theta_m} \psi(\theta_1, \ldots, \theta_m) d\Gamma_1(\theta_1) \cdots d\Gamma_1(\theta_m)$$

Expected error ϵ:

Pure Monte Carlo (MC) has $\epsilon = O(\|\psi\|_2 Z^{-1/2})$

Quasi Monte Carlo (QMC) has $\epsilon = O\big(\|\psi\|_{BV} Z^{-1} (\log Z)^m\big)$

Quadrature-formulas (integrand in $C^r(\Theta_m)$:

Full Product k-point Gauss $\epsilon = O(Z^{-(2r-1)/m})$ with $Z = O\Big(k^m\Big)$

Sparse Grid Smolyak $\epsilon = O(Z^{-r}(\log Z)^{(m-1)(r+1)})$ with $Z = O\big(\frac{2^k}{k!} m^k\big)$

Evaluation of Residuum through Integration

Model problem, evaluation of $\mathbb{E}\left((\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{H}(\boldsymbol{\theta})\mathbf{u}) - \boldsymbol{A}(\boldsymbol{\theta})[\boldsymbol{H}(\boldsymbol{\theta})\mathbf{u}])H_\gamma(\boldsymbol{\theta})\right)$

Polynomial-degree	σ of component	Monte Carlo $Z = 10^6$ abs. error $\cdot 10^3$	Quadrature $k = 10$ $Z = 36$ abs. error $\cdot 10^3$
0	0.26	0.5	≈ 0
1	0.27	0.2	0.008
2	0.61	1.2	≈ 0
3	0.77	1.5	0.07
4	2.29	4.5	≈ 0

For error $1 \cdot 10^{-3}$ in degree 4 ca. $\approx$ 20 million MC evaluations required.
Variance grows with degree.

Third Summary

- Stochastic Galerkin methods work.
- They are computationally possible on todays hardware.
- They are numerically stable, and have variational convergence theory behind them.
- They can use existing software efficiently.
- They can be sparsified via sparse tensor products.
- Software framework is being built for easy integration of existing software.

Important Features of Stochastic Galerkin

- For efficency try and use sparse representation throughout: ansatz in tensor products, as well as storage of solution and residuum—and matrix in tensor products, sparse grids for integration.
- In contrast to MCS, they are stable and have only cheap integrands.
- Can be coupled to existing software, only marginally more complicated than with MCS.

Outlook

- Stochastic problems at very beginning (like FEM in the 1960's), when to choose which stochastic discretisation?
- Nonlinear (and instationary) problems possible (but much more work).
- Development of framework for stochastic coupling and parallelisation.
- Computational algorithms have to be further developed.
- Hierarchical parallelisation well possible.