Institute of Mechanics
and Computational Mechanics

Leibniz
Universität
Hannover

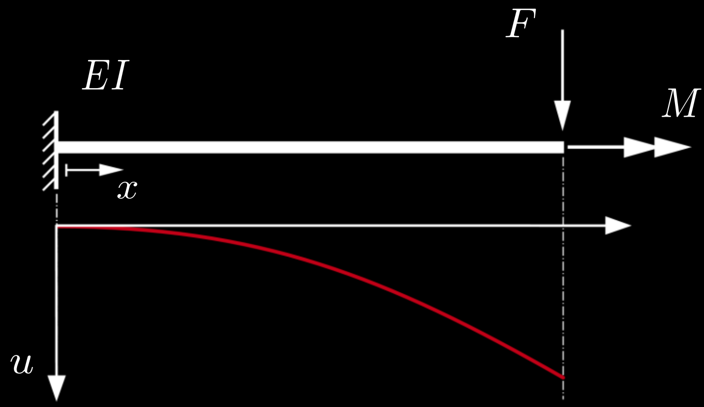
M. Mika, T.J.R. Hughes, D. Schillinger, P. Wriggers, R.R. Hiemstra

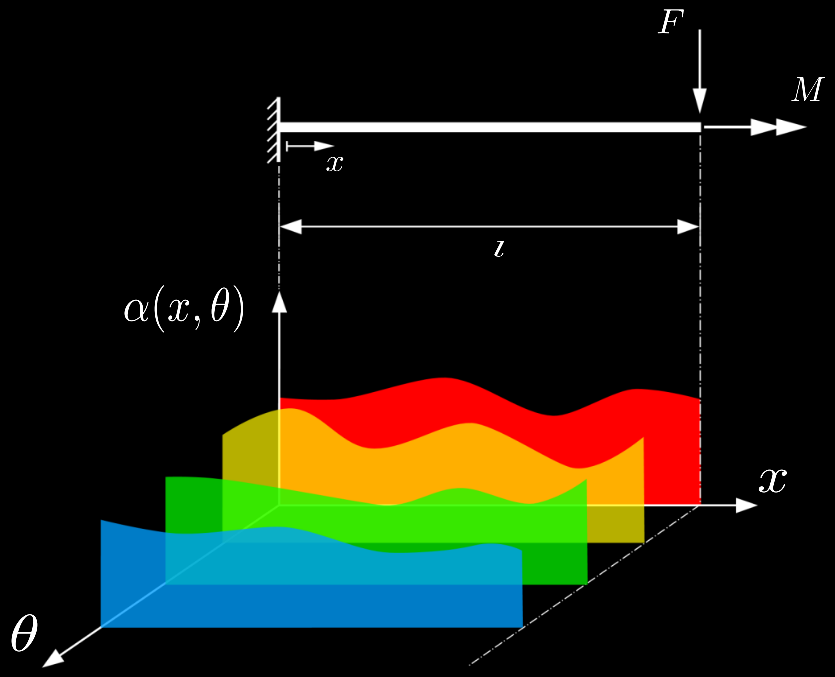
Matrix-free isogeometric Galerkin method Karhunen-Loève approximation of random fields

Using tensor product splines, tensor contraction and interpolation based quadrature

Nondeterminism of (physical) systems

$$\mathcal{L}(x)u(x) = f(x)$$ *deterministic*





stochastic $$\mathcal{L}(x,\theta)u(x,\theta) = f(x,\theta)$$

Most physical systems exhibit **randomness**, which, because of its lack of pattern or regularity, can not be explicitly captured by deterministic mathematical models

Random fields

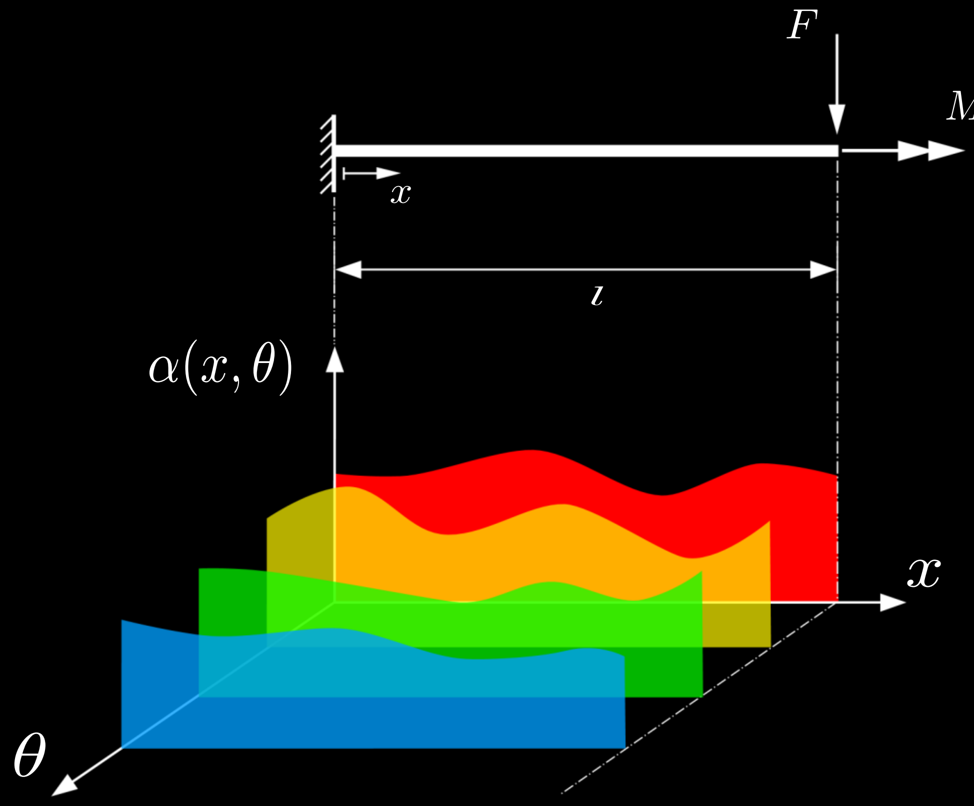
$$\alpha(\cdot,\theta) \;:\; \Theta \mapsto L^2(\mathcal{D})$$

Mean value

$$\mu(x) := \mathbb{E}\left[\alpha(x,\theta)\right]$$

Covariance function

$$\Gamma(x,x') := \mathbb{E}\left[(\alpha(x,\theta)-\mu(x))(\alpha(x',\theta)-\mu(x'))\right]$$



> A **random field** is a collection of deterministic functions on a bounded domain, called realizations, which are indexed by events in some sample set

Covariance functions

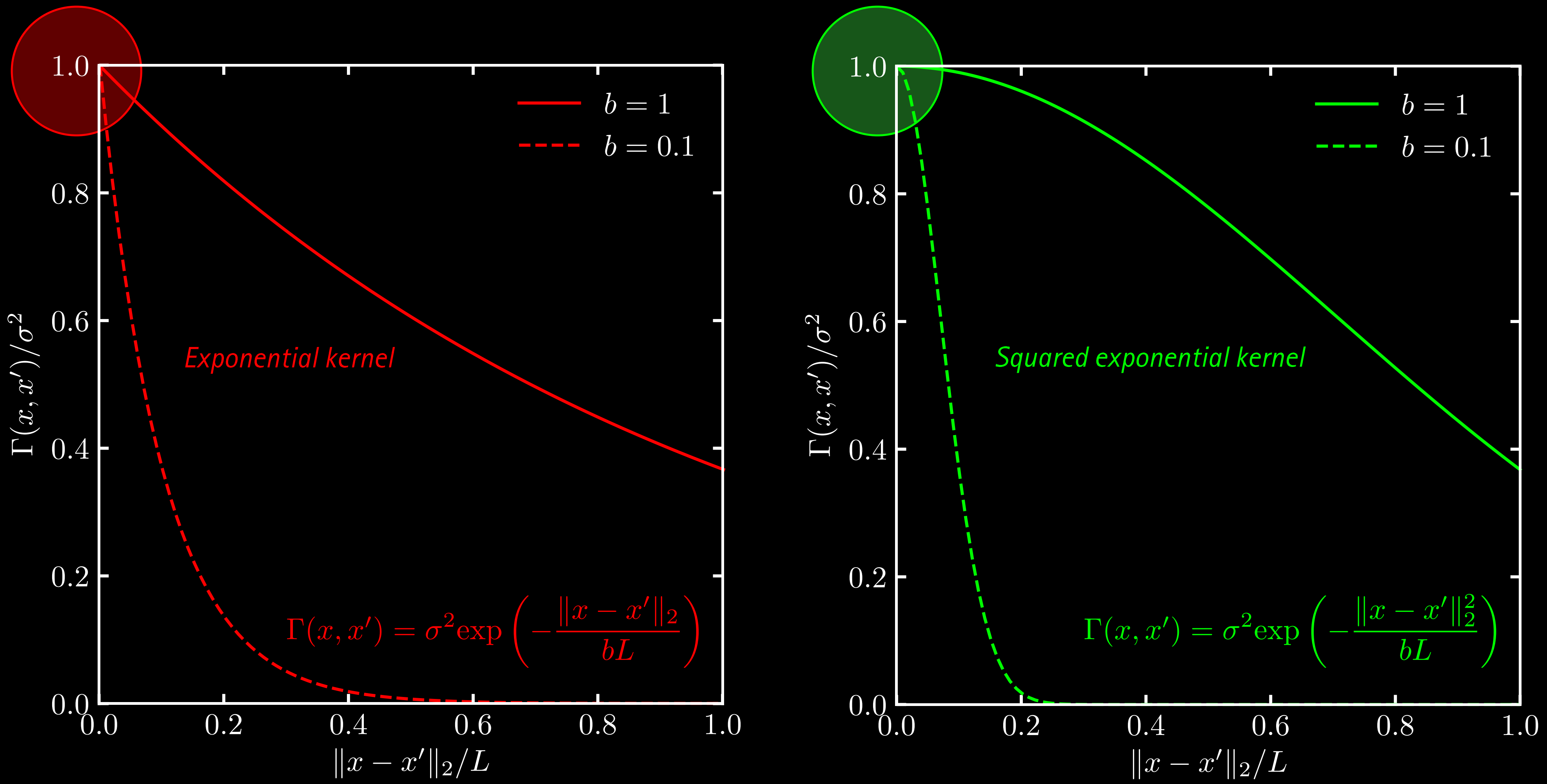


Figure 1: Common covariance functions (kernels), i.e. exponential covariance function (left) and squared exponential covariance function (right)

Random field discretization

Numerical treatment of a **continuous** random field requires discretization in the stochastic space!

Decompose the random field into a sum of the mean and a finite linear combination of L^2 orthogonal functions weighted by uncorrelated stochastic random variables

$$\tilde{\alpha}_M(x,\theta) = \mu(x) + \sum_{i=1}^{M} f_i(x)\xi_i(\theta)$$

Karhunen-Loève expansion

$$T\phi_i = \lambda_i \phi_i, \qquad (T\phi)(x) = \int_{\mathcal{D}} \Gamma(x, x')\phi(x')\,\mathrm{d}x'$$

Hilbert-Schmidt operator

Karhunen (1947) and Loève (1948)

$$f_i(x) = \sqrt{\lambda_i}\phi_i(x)$$

$$\tilde{\alpha}_M(x, \theta) = \mu(x) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(x)\xi_i(\theta)$$

The Karhunen-Loève series expansion yields the **best** *M*-term linear approximation of the random field, in the sense that the total mean squared error is minimized

Solution of the integral eigenvalue problem

- A range of different methods for IEVP has been proposed in the literature[1]
- Recently an isogeometric Galerkin method has been proposed [2]

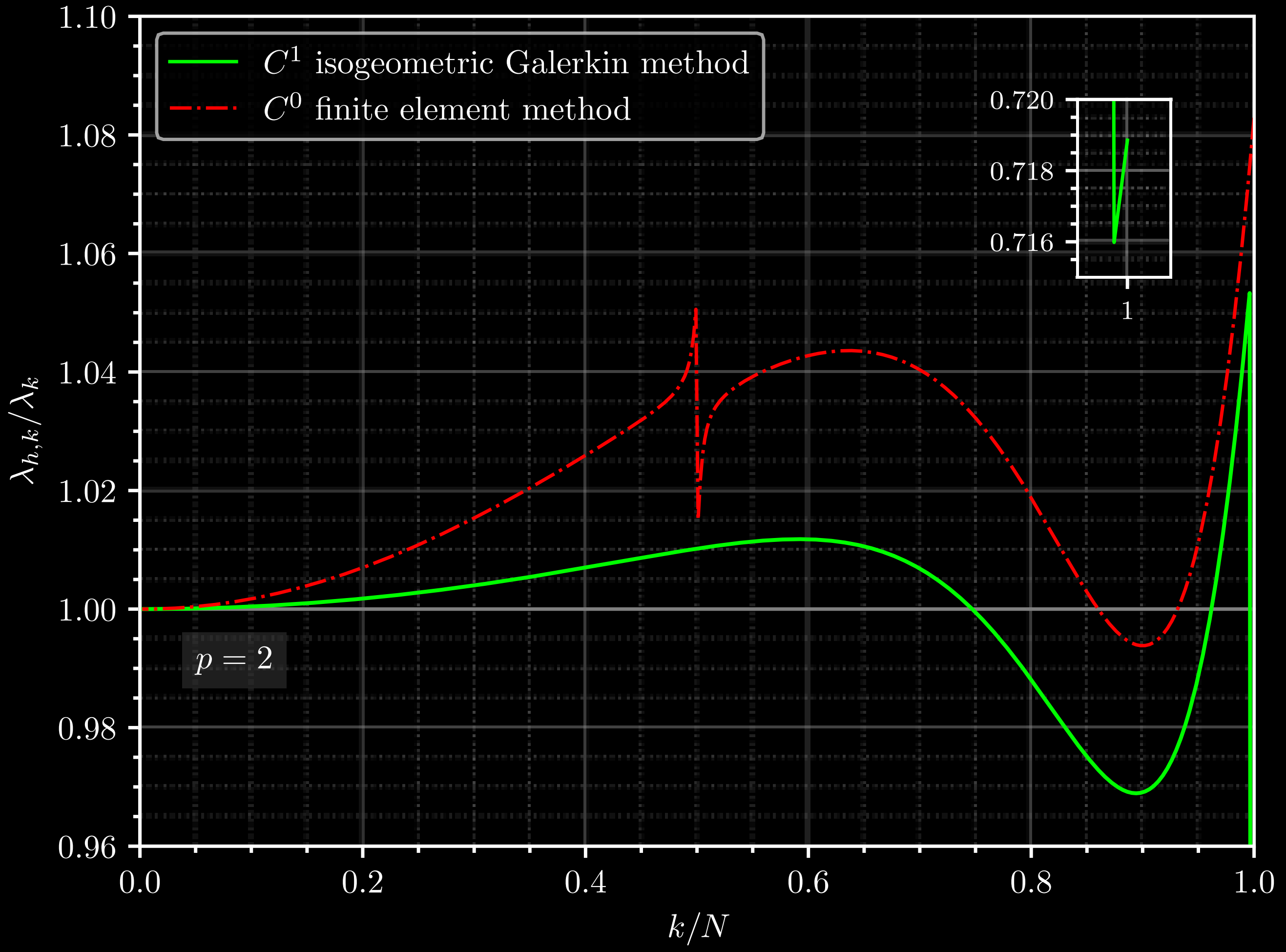
Find $\{\lambda_h, \phi_h\} \in \mathbb{R}_0^+ \times \mathcal{R}_h$ such that

$$\int_{\mathcal{D}} \left(\int_{\mathcal{D}'} \Gamma(x, x')\phi_h(x')\,\mathrm{d}x' - \lambda\phi_h(x) \right) \psi_h(x)\,\mathrm{d}x = 0 \qquad \forall \psi_h \in \underset{\text{Non-uniform rational B-splines}}{\mathcal{R}_h} \subset L^2(\mathcal{D})$$

Non-uniform rational B-splines

[1] For an overview see the review paper by Betz et al. *Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion* (2014)
[2] Rahman, S., *A Galerkin isogeometric method for Karhunen–Loève approximation of random fields* (2018)

Why an **isogeometric** Galerkin method?



Spectral properties of the method improve due to higher continuity the basis

Figure 2: Full spectrum of eigenvalues normalized with respect to a reference solution. Comparing quadratic C^1 continuous B-splines and standard quadratic C^0 continuous basis functions. One dimensional example with an exponential covariance function.

Standard discretization

$\mathsf{A} \in \mathbb{R}^{N\times N},\ \mathsf{Z} \in \mathbb{R}^{N\times N},\ \mathsf{v}_h \in \mathbb{R}^{N},\ \lambda_h \in \mathbb{R}_0^+,\ (\mathsf{i},\mathsf{j}) \in \mathcal{I},\ N := \#\mathcal{I}$

$$\mathsf{A}\mathsf{v}_h = \lambda_h \mathsf{Z}\mathsf{v}_h$$

Generalized algebraic eigenvalue problem

$$\mathsf{A}_{\mathsf{ij}} = \int_{\hat{\mathcal{D}}} \int_{\hat{\mathcal{D}}'} \hat{\Gamma}(\hat{x},\hat{x}') N_{\mathsf{i}}(\hat{x}) N_{\mathsf{j}}(\hat{x}') \det \mathrm{D}F(\hat{x}) \det \mathrm{D}F(\hat{x}')\, \mathrm{d}\hat{x}\, \mathrm{d}\hat{x}'$$

$$\mathsf{Z}_{\mathsf{ij}} = \int_{\hat{\mathcal{D}}} N_{\mathsf{i}}(\hat{x}) N_{\mathsf{j}}(\hat{x}) \det \mathrm{D}F(\hat{x})\, \mathrm{d}\hat{x}$$

Standard Galerkin methods for this class of integral eigenvalue problems are numerically challenging!

i. Due to numerical integration over a $2d$ dimensional domain the complexity of the assembly is $O(N \quad p^{3d})$

ii. The main system matrix **A** is a dense matrix, which requires $8N^2$ bytes of memory in double precision arithmetic

iii. The **generalized** algebraic eigenvalue problem (usually) requires a reformulation into a **standard** eigenvalue problem, i.e. using Cholesky decomposition of the right-hand-side mass matrix with complexity $O(N \quad)$

Figure 3: Complexity of assembly $O(N \quad p^{3d})$

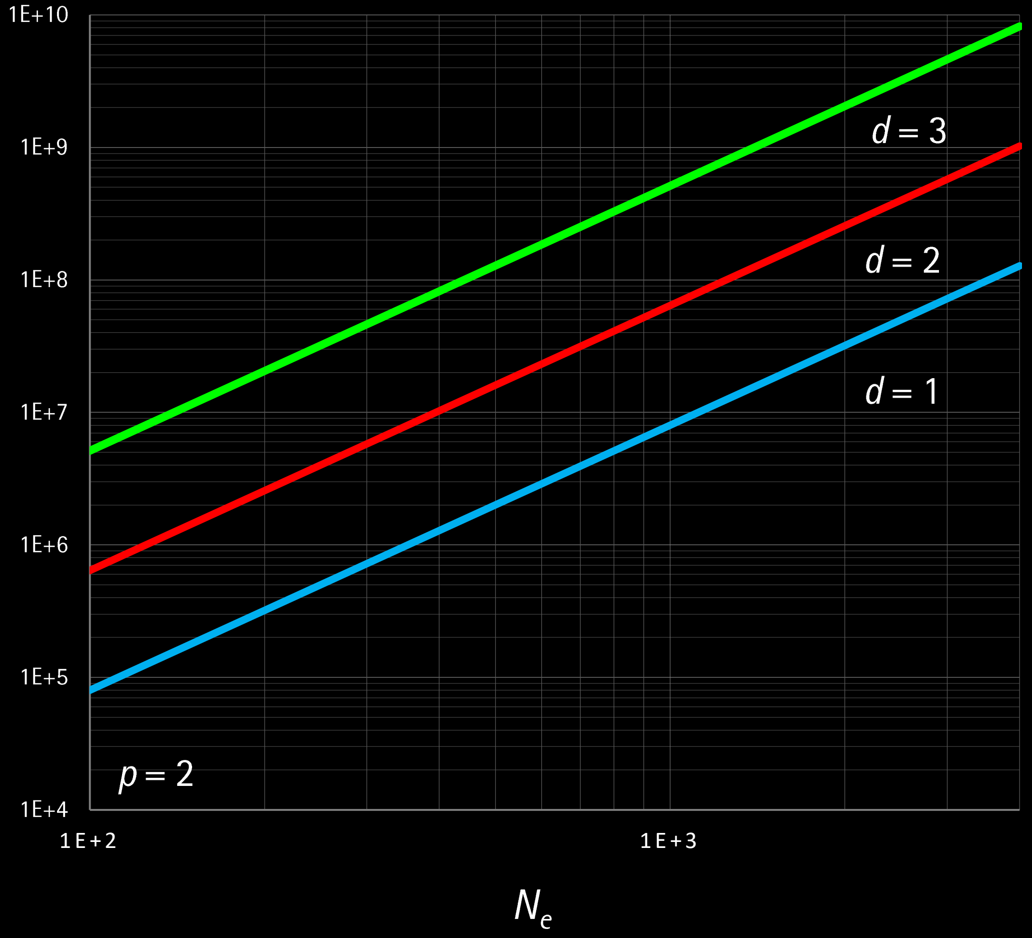


Figure 4: Complexity of Cholesky decomposition $O(N^3)$

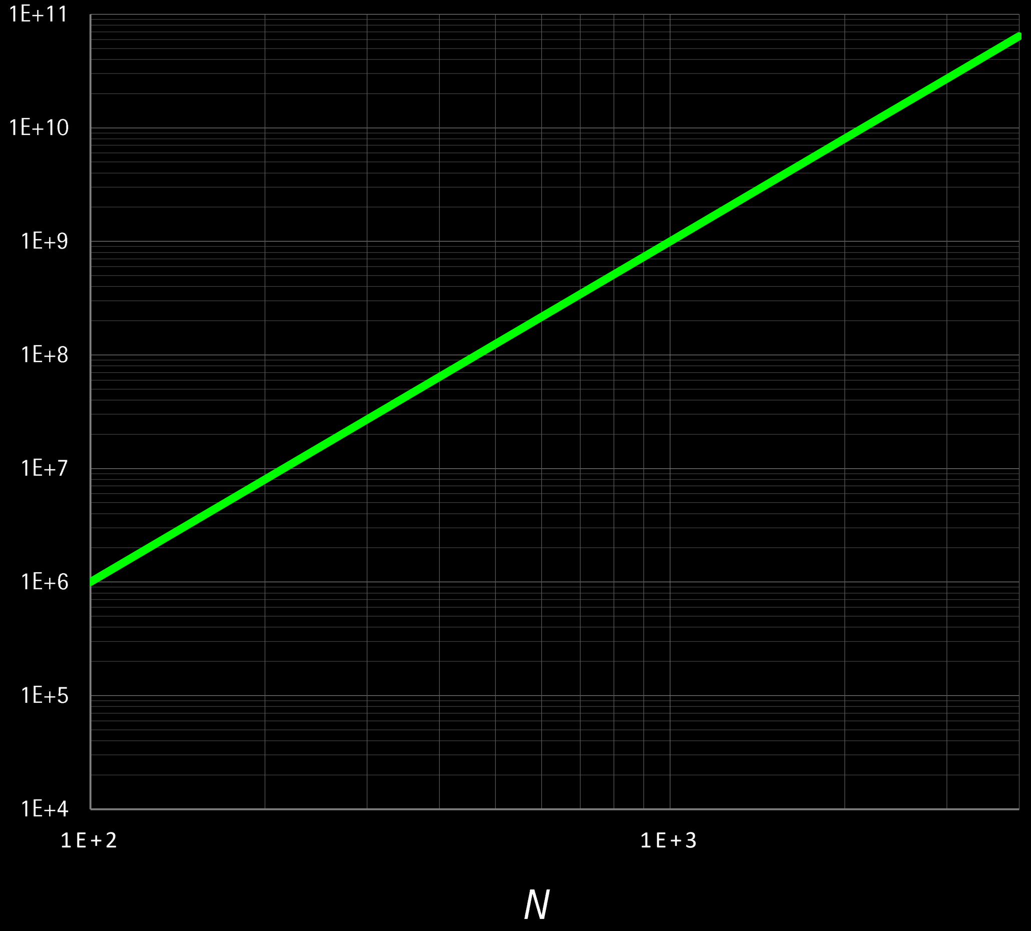


Table 1: Matrix-storage costs in double precision

Number of degrees of freedom	10^3	10^4	10^5	10^6
Matrix storage	8 MB	800 MB	80 GB	8 TB

New trial space

Search in $\mathcal{S}_h \subset L^2(\mathcal{D})$ where $\mathcal{S}_h := \operatorname{span}\left\{ \frac{B_{\mathsf{i}}(\hat{x})}{\sqrt{\det \mathrm{D}F(\hat{x})}} \right\}_{\mathsf{i}\in\mathcal{I}}$

$$\mathsf{A}_{\mathsf{ij}} = \int_{\mathcal{D}}\int_{\mathcal{D}'} \hat{\Gamma}(\hat{x},\hat{x}') \frac{B_{\mathsf{i}}(\hat{x})}{\sqrt{\det \mathrm{D}F(\hat{x})}} \frac{B_{\mathsf{j}}(\hat{x}')}{\sqrt{\det \mathrm{D}F(\hat{x}')}} \det \mathrm{D}F(\hat{x}) \det \mathrm{D}F(\hat{x}')\, \mathrm{d}\hat{x}\, \mathrm{d}\hat{x}'$$

$$\mathsf{Z}_{\mathsf{ij}} = \int_{\mathcal{D}} \frac{B_{\mathsf{i}}(\hat{x})}{\cancel{\sqrt{\det \mathrm{D}F(\hat{x})}}} \frac{B_{\mathsf{j}}(\hat{x})}{\cancel{\sqrt{\det \mathrm{D}F(\hat{x})}}} \cancel{\det \mathrm{D}F(\hat{x})}\, \mathrm{d}\hat{x}$$

New trial space

Search in $\mathcal{S}_h \subset L^2(\mathcal{D})$ where $\mathcal{S}_h := \operatorname{span}\left\{\dfrac{B_{\mathsf{i}}(\hat{x})}{\sqrt{\det \mathrm{D}F(\hat{x})}}\right\}_{\mathsf{i}\in\mathcal{I}}$

$$\mathsf{A}_{\mathsf{ij}} = \int_{\mathcal{D}}\int_{\mathcal{D}'} \hat{\Gamma}(\hat{x},\hat{x}')B_{\mathsf{i}}(\hat{x})B_{\mathsf{j}}(\hat{x}')\sqrt{\det \mathrm{D}F(\hat{x})\det \mathrm{D}F(\hat{x}')}\,\mathrm{d}\hat{x}\,\mathrm{d}\hat{x}'$$

$\mathsf{Z}_{\mathsf{ij}} = \int_{\mathcal{D}} B_{\mathsf{i}}(\hat{x})B_{\mathsf{j}}(\hat{x})\,\mathrm{d}\hat{x} = \mathsf{Z}_d \otimes \cdots \otimes \mathsf{Z}_1$ where $\mathsf{Z}_k = \int_0^1 B_{i_k,p_k}(\hat{x}_k)B_{j_k,p_k}(\hat{x}_k)\,\mathrm{d}\hat{x}_k$

Integrated exactly up machine precision using
Gauss-Legendre quadrature rule with (p+1) quadrature points per element

Standard algebraic eigenvalue problem

$$\mathsf{A}\mathsf{v}_h = \lambda_h \mathsf{Z}\mathsf{v}_h \qquad \Rightarrow \qquad \mathsf{A}'\mathsf{v}_h = \lambda_h \mathsf{v}_h$$

$$\mathsf{A}' = \mathsf{L}^{-1}\mathsf{A}\mathsf{L}^{-\top} \qquad \text{where} \qquad \mathsf{L}\mathsf{L}^{\top} = \mathsf{Z}$$

Using the Kronecker structure of the matrix Z one can reduce the cost of a Cholesky decomposition from $O(N\;\;)$ to $O(n\;\;)$, where n is the number of degrees of freedom in a single parametric direction

$$\begin{aligned}
\mathsf{Z} &= \mathsf{Z}_d \otimes \cdots \otimes \mathsf{Z}_1 \\
&= \mathsf{L}_d\mathsf{L}_d^{\top} \otimes \cdots \otimes \mathsf{L}_1\mathsf{L}_1^{\top} \\
&= (\mathsf{L}_d \otimes \cdots \otimes \mathsf{L}_1)(\mathsf{L}_d \otimes \cdots \otimes \mathsf{L}_1)^{\top} = \mathsf{L}\mathsf{L}^{\top}
\end{aligned}$$

Novel interpolation based quadrature method[1,2]

To reduce assembly costs of the main system matrix, we propose a novel method that

i. is optimal with regard to the number of integration points

ii. has complexity independent of polynomial degree, which enables higher order methods

Computational complexity in comparison

$$\mathcal{O}\left(N_e^2(p+1)^{3d}\right)$$

Standard FEM

$$\mathcal{O}\left(2dN_e^2(p+1)^{2d+1}\right)$$

Sum factorization

$$\mathcal{O}\left(\tilde{N}^2 N_{\text{iter}}/N_{\text{thread}}\right)$$

Interpolation based quadrature

[1] Arthur, D.W., *The Solution of Fredholm Integral Equations Using Spline Functions* (1973)

[2] Mantzaflaris, A., Jüttler, B., *Integration by interpolation and look-up for Galerkin-based isogeometric analysis* (2015)

Interpolation based quadrature

$$\mathsf{A}_{\mathsf{ij}} = \int_{\mathcal{D}}\int_{\mathcal{D}'} \hat{\Gamma}(\hat{x},\hat{x}')\sqrt{\det \mathrm{D}F(\hat{x})\det \mathrm{D}F(\hat{x}')}\, B_{\mathsf{i}}(\hat{x})B_{\mathsf{j}}(\hat{x}')\,\mathrm{d}\hat{x}\,\mathrm{d}\hat{x}'$$

$$G(\hat{x},\hat{x}') := \hat{\Gamma}(\hat{x},\hat{x}')\sqrt{\det \mathrm{D}F(\hat{x})\det \mathrm{D}F(\hat{x}')}$$

Interpolation i.e. at Greville abscissae

$$\tilde{G}(\hat{x}_{\mathsf{m}},\hat{x}'_{\mathsf{n}}) := \sum_{\mathsf{k},\mathsf{l}\in\tilde{\mathcal{I}}} \tilde{\mathsf{G}}_{\mathsf{kl}}\tilde{B}_{\mathsf{k}}(\hat{x}_{\mathsf{m}})\tilde{B}_{\mathsf{l}}(\hat{x}'_{\mathsf{n}}) = \tilde{\mathsf{B}}^{\top}\tilde{\mathsf{G}}\tilde{\mathsf{B}}$$

$$\tilde{\mathsf{A}}_{\mathsf{ij}} = \int_{\mathcal{D}}\int_{\mathcal{D}'} \tilde{G}(\hat{x},\hat{x}')\, B_{\mathsf{i}}(\hat{x})B_{\mathsf{j}}(\hat{x}')\,\mathrm{d}\hat{x}\,\mathrm{d}\hat{x}'$$

$$\tilde{\mathsf{A}}_{\mathsf{ij}} = \sum_{\mathsf{k},\mathsf{l}\in\tilde{\mathcal{I}}} \tilde{\mathsf{G}}_{\mathsf{kl}} \int_{\mathcal{D}} \tilde{B}_{\mathsf{k}}(\hat{x})B_{\mathsf{i}}(\hat{x})\,\mathrm{d}\hat{x} \int_{\mathcal{D}'} \tilde{B}_{\mathsf{l}}(\hat{x}')B_{\mathsf{j}}(\hat{x}')\,\mathrm{d}\hat{x}'$$

$$\tilde{\mathsf{A}}_{\mathsf{ij}} = \sum_{\mathsf{k},\mathsf{l}\in\tilde{\mathcal{I}}} \tilde{\mathsf{G}}_{\mathsf{kl}}\mathsf{M}_{\mathsf{ki}}\mathsf{M}_{\mathsf{lj}}$$

$$\mathsf{M}_{\mathsf{ki}} = \mathsf{M}_{d}\otimes\cdots\otimes\mathsf{M}_{1} \quad \text{where}$$

$$\mathsf{M}_{k} = \int_0^1 \tilde{B}_{i_k,\tilde{p}_k}(\hat{x}_k)B_{j_k,p_k}(\hat{x}_k)\,\mathrm{d}\hat{x}_k$$

Integrated exactly up machine precision using
Gauss-Legendre quadrature rule with (p+1) quadrature points per element

The approximation error is entirely due to the interpolation error

Interpolation error analysis

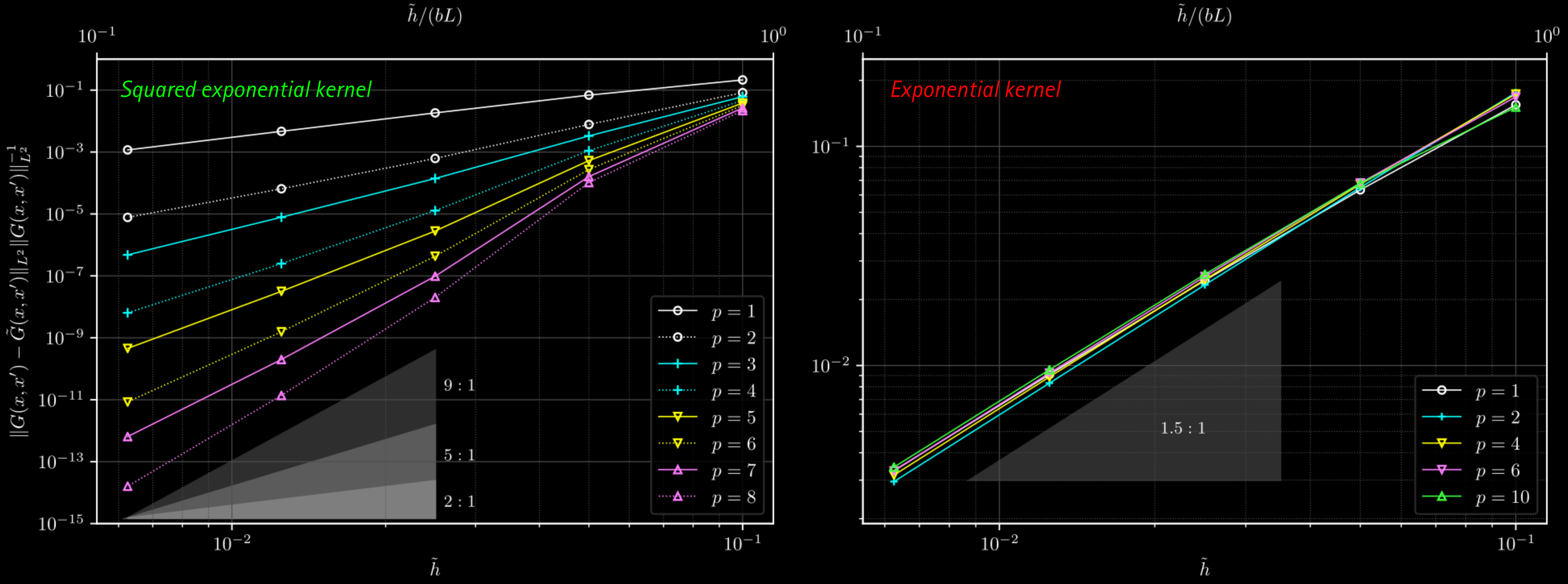
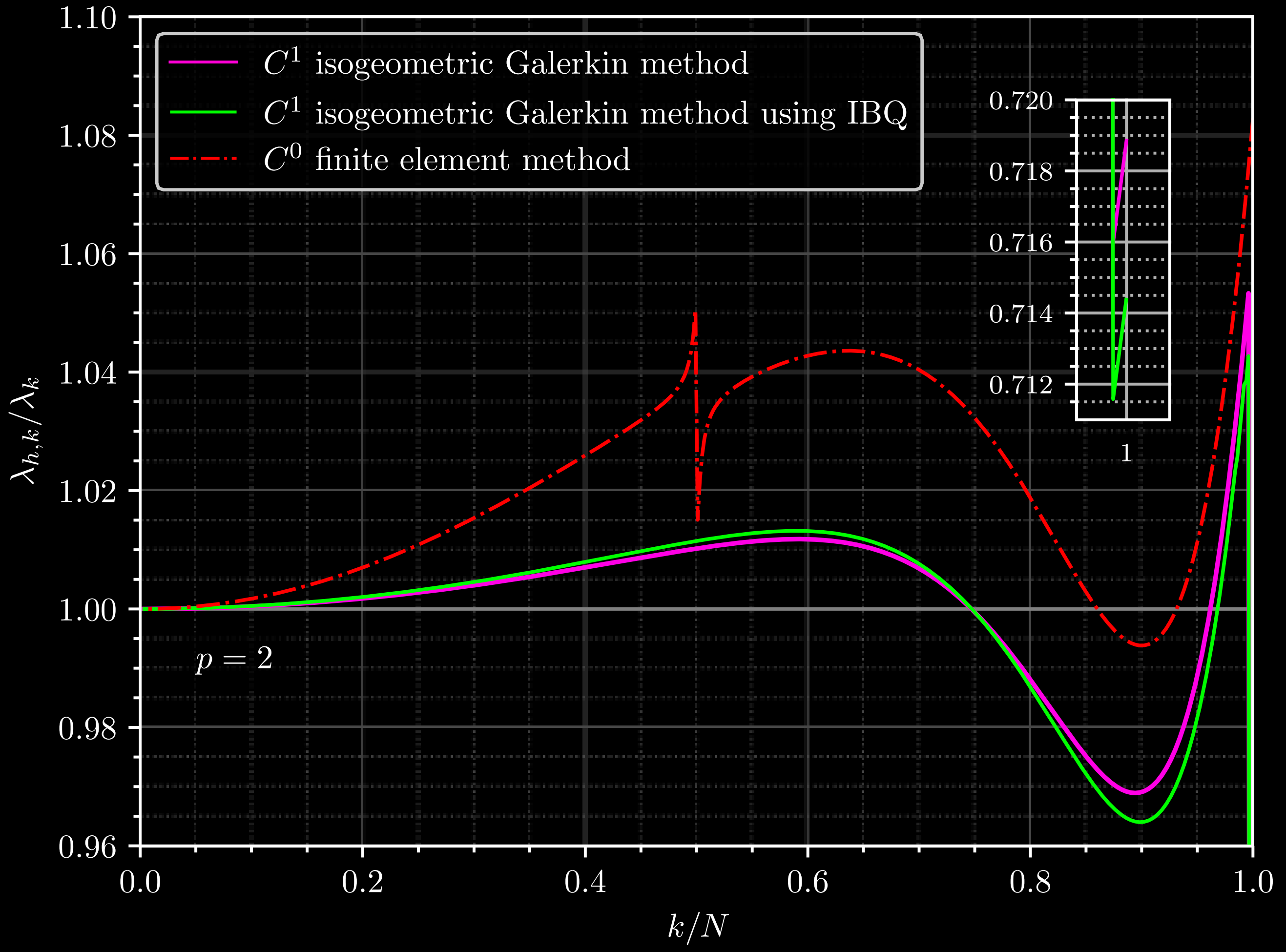


Figure 5: Normalized L^2 interpolation error in a one-dimensional study case for the squared exponential kernel (left) and the exponential kernel (right) with convergence rates $O(\tilde{l}^{p+1})$ and $O(\tilde{l}^{3/2})$, respectively.

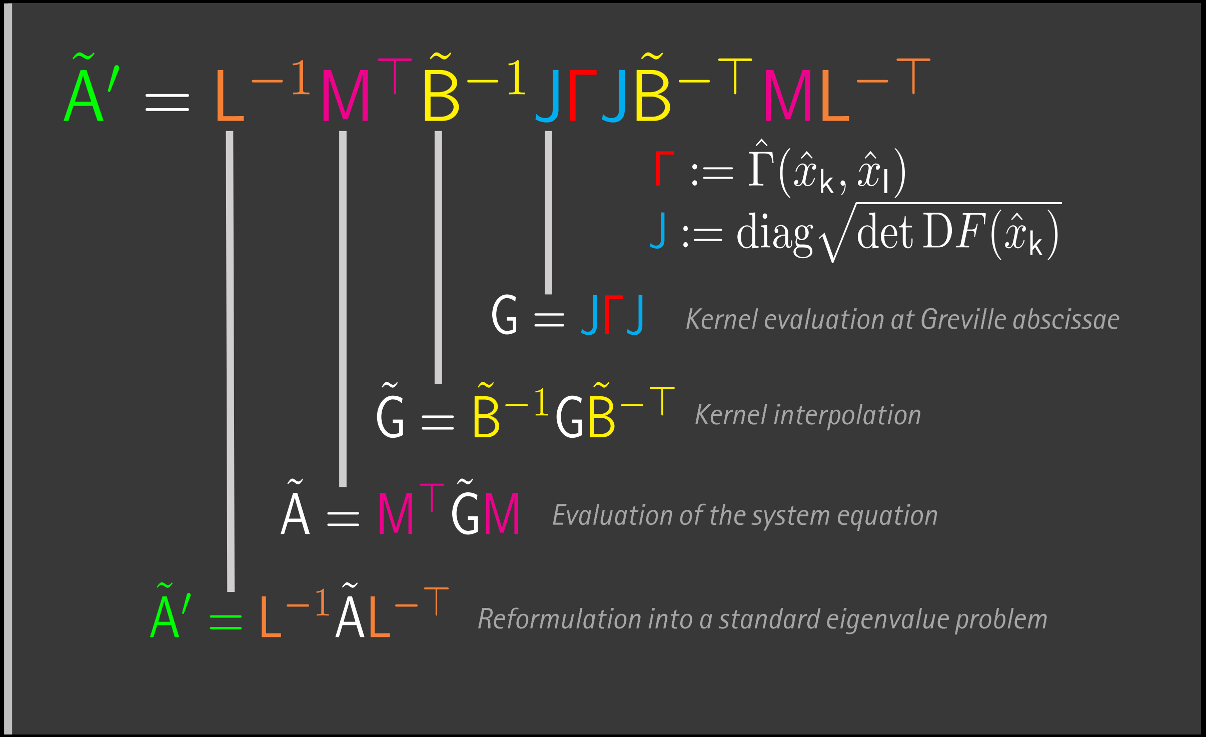
Reproduction of spectral properties



Spectral properties of the standard isogeometric Galerkin method remain preserved while applying the interpolation based quadrature

Figure 6: Full spectrum of eigenvalues normalized with respect to a reference solution. Comparing quadratic C^1 continuous B-splines, C^1 continuous B-splines using interpolation based quadrature (IBQ) and standard quadratic C^0 continuous basis functions. One dimensional example with an exponential covariance function.

Proposed matrix-free matrix-vector product $\tilde{\mathsf{A}}'\mathsf{v} = \mathsf{v}'$



$$\tilde{\mathsf{A}}' = \mathsf{L}^{-1}\mathsf{M}^{\top}\tilde{\mathsf{B}}^{-1}\mathsf{J}\Gamma\mathsf{J}\tilde{\mathsf{B}}^{-\top}\mathsf{M}\mathsf{L}^{-\top}$$

$$\Gamma := \hat{\Gamma}(\hat{x}_{\mathsf{k}}, \hat{x}_{\mathsf{l}})$$

$$\mathsf{J} := \operatorname{diag}\sqrt{\det \mathrm{D}F(\hat{x}_{\mathsf{k}})}$$

$\mathsf{G} = \mathsf{J}\Gamma\mathsf{J}$ *Kernel evaluation at Greville abscissae*

$\tilde{\mathsf{G}} = \tilde{\mathsf{B}}^{-1}\mathsf{G}\tilde{\mathsf{B}}^{-\top}$ *Kernel interpolation*

$\tilde{\mathsf{A}} = \mathsf{M}^{\top}\tilde{\mathsf{G}}\mathsf{M}$ *Evaluation of the system equation*

$\tilde{\mathsf{A}}' = \mathsf{L}^{-1}\tilde{\mathsf{A}}\mathsf{L}^{-\top}$ *Reformulation into a standard eigenvalue problem*

Besides the kernel evaluation, all matrices have a **Kronecker product structure** – we can utilize it to efficiently implement the matrix-vector product for iterative eigenvalue solvers!

Input: $v_{i_1\ldots i_d} \in \mathbb{R}^{n_1\times\cdots\times n_d}$, $J_{l_1\ldots l_d} \in \mathbb{R}^{\tilde{n}_1\times\cdots\times\tilde{n}_d}$, $B_{i_k j_k} \in \mathbb{R}^{\tilde{n}_k\times\tilde{n}_k}$ and $M_{l_k j_k} \in \mathbb{R}^{\tilde{n}_k\times n_k}$

Output: $v'_{i_1\ldots i_d} \in \mathbb{R}^{n_1\times\cdots\times n_d}$

```
$V_{j_1\ldots j_d} \leftarrow L^{-1}_{i_1 j_1}\cdots L^{-1}_{i_d j_d} v_{i_1\ldots i_d}$
$X_{k_1\ldots k_d} \leftarrow M_{k_1 j_1}\cdots M_{k_d j_d} V_{j_1\ldots j_d}$
$Y_{l_1\ldots l_d} \leftarrow B^{-1}_{k_1 l_1}\cdots B^{-1}_{k_d l_d} X_{k_1\ldots k_d}$
$Y'_{l_1\ldots l_d} \leftarrow J_{l_1\ldots l_d} \odot Y_{l_1\ldots l_d}$
$Z'_{k_1\ldots k_d} \leftarrow \hat{\Gamma}_{k_1\ldots k_d l_1\ldots l_d} Y'_{l_1\ldots l_d}$   $\Leftarrow \mathcal{O}(\tilde{N}^2 N_{\text{iter}}/N_{\text{threads}})$
$Z_{k_1\ldots k_d} \leftarrow J_{k_1\ldots k_d} \odot Z'_{k_1\ldots k_d}$
$Y_{j_1\ldots j_d} \leftarrow B^{-1}_{j_1 k_1}\cdots B^{-1}_{j_d k_d} Z_{k_1\ldots k_d}$
$V_{l_1\ldots l_d} \leftarrow M_{j_1 l_1}\cdots M_{j_d l_d} Y_{j_1\ldots j_d}$
$v'_{i_1\ldots i_d} \leftarrow L^{-1}_{i_1 l_1}\cdots L^{-1}_{i_d l_d} V_{l_1\ldots l_d}$
```

Algorithm 1: Matrix-free matrix-vector product using the Kronecker structure and tensor contraction[1] for efficient evaluation

[1] Bressan, A., Takacs, S., *Sum factorization techniques in Isogeometric Analysis* (2019)

Benchmark case

In this benchmark we compare the performance of a solution obtained by Rahman (2018) using the standard isogeometric Galerkin method and our own fast IBQ method, while keeping the estimated eigenvalues comparable

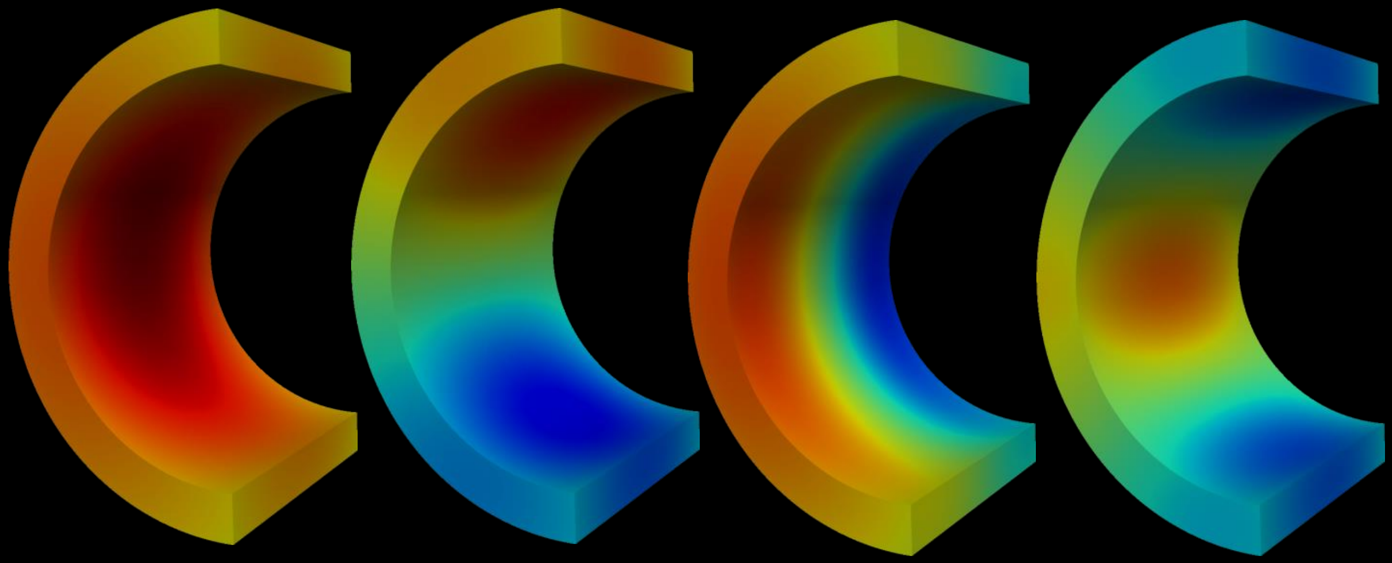
Kernel under consideration is the exponential kernel

Mode #	IBQ	Benchmark
1	162.7993245	162.7993688
2	91.43092739	91.43079317
3	57.56741771	57.56765769
4	51.09028325	51.09029418
5	38.79796382	38.79808752
6	27.90416545	27.90392169
7	25.05648777	25.05681566

Table 2: Comparison of eigenvalues with the analogous test case presented by Rahman (2018)

	IBQ	Benchmark
Solution time	12.91 s	> 24 h
NDOF Solution space	59940	1050
NDOF Interpolation	8525	–
Memory	0.587 GB	–
Number of iterations	63	–

Table 3: Performance comparison between IBQ and the standard isogeometric Galerkin method presented by Rahman (2018)



$p = \{2, 1, 1\}$

High-order test case

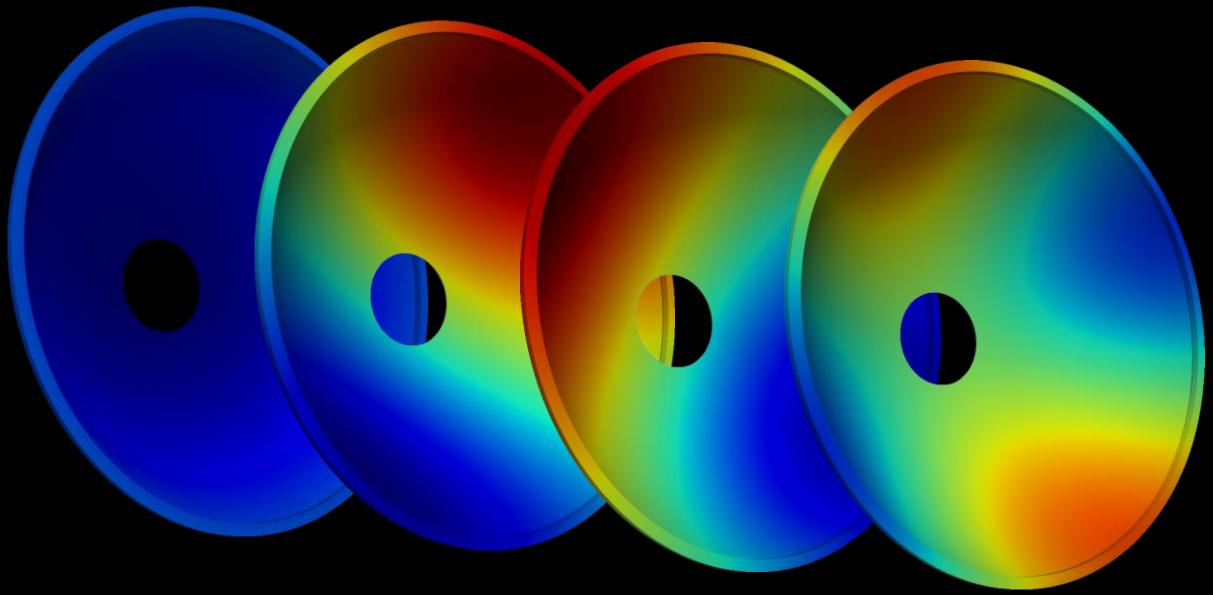
> In this benchmark we use a high-order interpolation and solution space and compare performance and eigenvalues for two computations under *h*-refinement of the solution space
>
> Kernel under consideration is the squared exponential kernel

Mode #	Case 1	Case 2
1	14476.27783	14476.27783
2	6531.770729	6531.770729
3	6531.770728	6531.770725
4	2091.877012	2091.877015
5	2091.877003	2091.877012
6	1971.747135	1971.747136
7	552.7086269	552.7086262

Table 4: Comparison of eigenvalues in both test cases

	Case 1	Case 2
Polynomial order	16	16
Solution time	1.65 h	1.63 h
NDOF Solution space	1.80E5	4.21E5
NDOF Interpolation	1.80E5	1.80E5
Memory	1.34 GB	7.48 GB
Number of iterations	41	41

Table 5: Performance comparison between both test cases

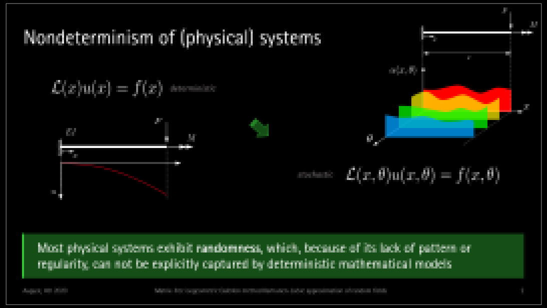


In conclusion

- Novel projection based quadrature of the weak form is used for efficient formation of high-dimensional integrands of high polynomial order
- Inexpensive reformulation of the generalized eigenvalue problem into a standard eigenvalue problem using Kronecker product of univariate mass matrices
- Reduced memory usage by implementing a fast multithreaded matrix–free matrix–vector product for iterative eigenvalue solvers by using tensor contraction of Kronecker structured matrices

Further research effort is required towards analysis of the error, as well as the spectral properties of the method

Discussion

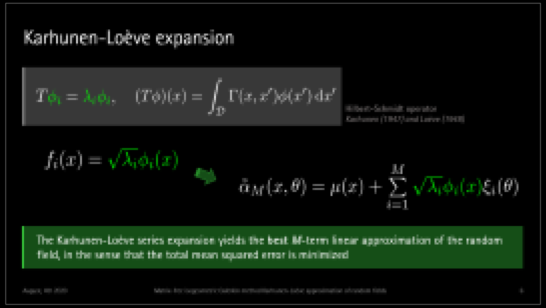


Nondeterminism of (physical) systems

$\mathcal{L}(x)u(x) = f(x)$ deterministic

stochastic $\mathcal{L}(x,\theta)u(x,\theta) = f(x,\theta)$

Most physical systems exhibit randomness, which, because of its lack of pattern or regularity, can not be explicitly captured by deterministic mathematical models



Karhunen-Loève expansion

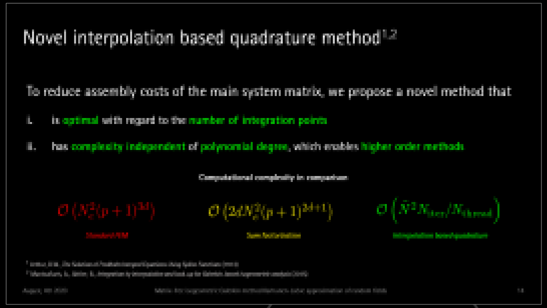
$\mathcal{T}\phi_i = \lambda_i\phi_i, \quad (\mathcal{T}\phi)(x) = \int_D \Gamma(x,x')\phi(x')\,\mathrm{d}x'$

Hilbert-Schmidt operator
Karhunen (1947) and Loève (1948)

$f_i(x) = \sqrt{\lambda_i}\phi_i(x)$

$\hat{\alpha}_M(x,\theta) = \mu(x) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(x)\xi_i(\theta)$

The Karhunen-Loève series expansion yields the best M-term linear approximation of the random field, in the sense that the total mean squared error is minimized



Novel interpolation based quadrature method[1,2]

To reduce assembly costs of the main system matrix, we propose a novel method that

i. is optimal with regard to the number of integration points
ii. has complexity independent of polynomial degree, which enables higher order methods

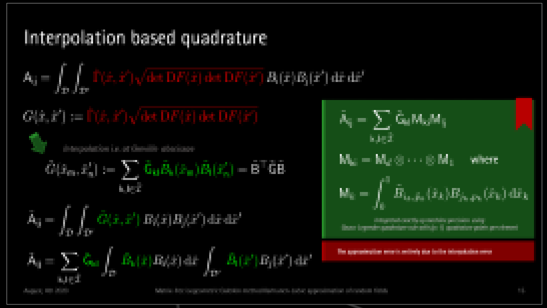
Computational complexity in comparison

$\mathcal{O}\left(N_e^2(p+1)^{3d}\right)$ Standard FEM

$\mathcal{O}\left(2dN_e^2(p+1)^{2d+1}\right)$ Sum factorization

$\mathcal{O}\left(\tilde{N}^2 N_{\text{iter}}/N_{\text{thread}}\right)$ Interpolation based quadrature

[1] [illegible]
[2] [illegible]



Interpolation based quadrature

$A_{ij} = \int_D\int_{D'} \Gamma(\hat{x},\hat{x}')\sqrt{\det DF(\hat{x})\det DF(\hat{x}')}\,B_i(\hat{x})B_j(\hat{x}')\,\mathrm{d}\hat{x}\,\mathrm{d}\hat{x}'$

$G(\hat{x},\hat{x}') := \Gamma(\hat{x},\hat{x}')\sqrt{\det DF(\hat{x})\det DF(\hat{x}')}$

Interpolation i.e. at Greville abscissae

$\tilde{G}(\hat{x}_m,\hat{x}'_n) := \sum_{k,l\in\mathcal{I}} \tilde{G}_{kl}\tilde{B}_k(\hat{x}_m)\tilde{B}_l(\hat{x}'_n) = \tilde{B}^{\top}\tilde{G}\tilde{B}$

$\tilde{A}_{ij} = \int_D\int_{D'} \tilde{G}(\hat{x},\hat{x}')\,B_i(\hat{x})B_j(\hat{x}')\,\mathrm{d}\hat{x}\,\mathrm{d}\hat{x}'$

$\tilde{A}_{ij} = \sum_{k,l\in\mathcal{I}} \tilde{G}_{kl}\int_{D} \tilde{B}_k(\hat{x})B_i(\hat{x})\,\mathrm{d}\hat{x}\int_{D'} \tilde{B}_l(\hat{x}')B_j(\hat{x}')\,\mathrm{d}\hat{x}'$

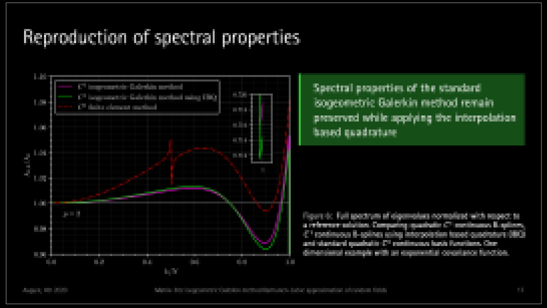
$\tilde{A}_{ij} = \sum_{k,l\in\mathcal{I}} \tilde{G}_{kl}M_{ki}M_{lj}$

$M_{ki} = M_d \otimes \cdots \otimes M_1$ where

$M_{k} = \int_0^1 \tilde{B}_{i_k,\tilde{p}_k}(\hat{x}_k)B_{j_k,p_k}(\hat{x}_k)\,\mathrm{d}\hat{x}_k$

[illegible]

[illegible]

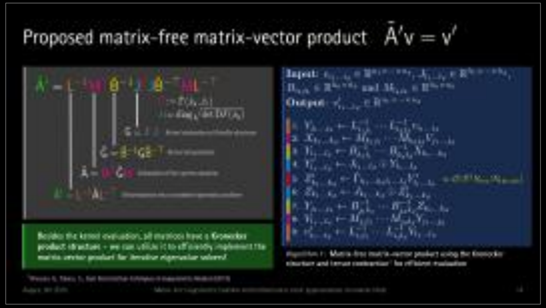


Reproduction of spectral properties

C^1 isogeometric Galerkin method
C^1 isogeometric Galerkin method using IBQ
C^0 finite element method
$p = 2$
l/N
0.0
0.2
0.4
0.6
0.8
1.0

Spectral properties of the standard isogeometric Galerkin method remain preserved while applying the interpolation based quadrature

Figure 6: Full spectrum of eigenvalues normalized with respect to a reference solution. Comparing quadratic C^1 continuous B-splines, C^1 continuous B-splines using interpolation based quadrature (IBQ) and standard quadratic C^0 continuous basis functions. One dimensional example with an exponential covariance function.



Proposed matrix-free matrix-vector product $\tilde{A}'v = v'$

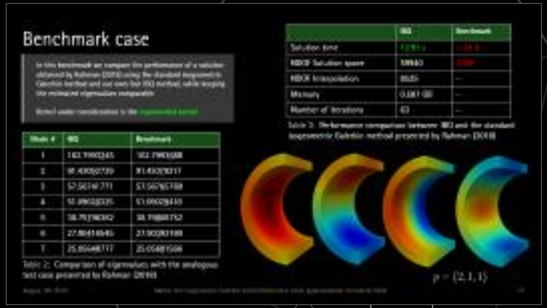
Input: [illegible]

Output: [illegible]

Besides the kernel evaluation, all matrices have a Kronecker product structure – we can utilize it to efficiently implement the matrix-vector product for iterative eigenvalue solvers!

Algorithm 1: Matrix-free matrix-vector product using the Kronecker structure and tensor contraction[1] for efficient evaluation

[1] [illegible]



Benchmark case

In this benchmark we compare the performance of a solution obtained by Rahman (2018) using the standard isogeometric Galerkin method and our own fast IBQ method, while keeping the estimated eigenvalues comparable

Kernel under consideration is the exponential kernel

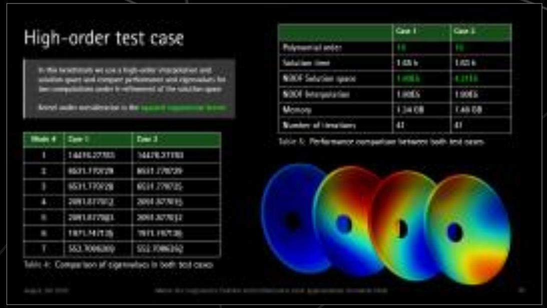
Mode #	IBQ	Benchmark
1	[illegible]	[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]
4	[illegible]	[illegible]
5	[illegible]	[illegible]
6	[illegible]	[illegible]
7	[illegible]	[illegible]

Table 2: Comparison of eigenvalues with the analogous test case presented by Rahman (2018)

	IBQ	Benchmark
Solution time	[illegible]	[illegible]
NDOF Solution space	[illegible]	[illegible]
NDOF Interpolation	[illegible]	–
Memory	[illegible]	–
Number of iterations	63	–

Table 3: Performance comparison between IBQ and the standard isogeometric Galerkin method presented by Rahman (2018)

$p = (2,1,1)$



High-order test case

In this benchmark we use a high-order interpolation and solution space and compare performance and eigenvalues for two computations under k-refinement of the solution space

Kernel under consideration is the squared exponential kernel

Mode #	Case 1	Case 2
1	[illegible]	[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]
4	[illegible]	[illegible]
5	[illegible]	[illegible]
6	[illegible]	[illegible]
7	[illegible]	[illegible]

Table 4: Comparison of eigenvalues in both test cases

	Case 1	Case 2
Polynomial order	[illegible]	[illegible]
Solution time	1.03 h	1.03 h
NDOF Solution space	[illegible]	[illegible]
NDOF Interpolation	1.00E6	1.00E6
Memory	1.34 GB	1.46 GB
Number of iterations	41	41

Table 5: Performance comparison between both test cases

Michal Mika, M. Sc.

Institute of Mechanics and Computational Mechanics
Leibniz Universität Hannover
Phone: +49 511.762-14095
Mail: mika@ibnm.uni-hannover.de
Homepage: https://mika.sh/