Technische Universität Braunschweig

INSTITUT FÜR AKUSTIK

A Statistical Approach for the Fusion of Data and Finite Element Analysis in Vibroacoustics

Masterarbeit

Lucas Hermann

Matr.-Nr.: 4990987

October 2021

Erstprüferin: Prof. Dr.-Ing. Sabine C. Langer

Zweitprüfer: Prof. Dr.-Ing. Ulrich Römer

Declaration

Hiermit versichere ich, Lucas Hermann, durch meine Unterschrift, dass ich die vorliegende Masterarbeit mit dem Titel "<<Titel>>" selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinn- gemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Insbesondere sind auch solche Inhalte gekennzeichnet, die von betreuenden wissenschaftlichen Mitarbeiterinnen und Mitarbeitern des Instituts für Akustik eingebracht wurden.

Die Arbeit oder Auszüge daraus haben noch nicht in gleicher oder ähnlicher Form dieser oder einer anderen Prüfungsbehörde vorgelegen.

Mir ist bewusst, dass Verstöße gegen die Grundsätze der Selbstständigkeit als Täuschung betrachtet und entsprechend der Prüfungsordnung geahndet werden.

Braunschweig, May 21, 2021

Lucas Hermann

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction

Dies ist ein Text in tubsSecondary. Dies ist ein Text in tubsViolet. Dies ist ein Text in tubsGreenDark.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

- Aufzählungspunkt Eins
- Aufzählungspunkt Zwei
  - Unter-Aufzählungspunkt Eins
  - Unter-Aufzählungspunkt Zwei
- Aufzählungspunkt Drei

2 Acoustics in the Frequency Domain

3 The Classical Finite Element Method

4 Probability Theroy and Bayesian Inference

$posterior = \frac{likelihood x prior}{marginal likelihood}$ (Rasmussen p.9)

5 Gaussian Process Regression

The following closely follows [Rasmussen]. Gaussian Processes are a class of Bayesian non-parametric models. Non-parametric doesn't mean that there are no parameters involved but rather that there is an infinite number of them. Every realization of a Gaussian process doesn't yield a scalar or vector but a function. One can think of a Gaussian Process as a collection of infinitely many normally distributed random variables, i.e. a generalization of a Gaussian distribution: A vector with infinitely many entries is basically a function. By picking out a finite set of those random variables when discretizing e.g. on an FEM mesh, one obtains a multivariate distribution which is determined by a mean and a covariance matrix. [Rasmussen p.2] Hence, the GP assigns a confidence band to a function where a usual parametric regression wouldn't.

A GP is described by its mean function and the covariance function, also called kernel.

MArginalization Property: It means that out of a multivariate Gaussian the single variables can directly be taken out of the mean vector and the covariance matrix. E.g. for a point on the axis describved by the GP, a single value can be picked out. A univariate Gaussian comes out by simply using the mean vectors value at that point and the i,i th index of the covariance matrix. This means that the underlying distributions for single variables do not change if a GP is used to describe an arbitrary set of variables [p13].

-Regression in general -differences of a GP regression to e.g. a polynomial regression: every point is assgined an uncertainty

-a distribution over functions

-Herleitung der Kovarianzmatrix, evtl. auch für das standard linear regression modell

-parameter variations

5.1 Kernel Functions

p14:

If a kernel function is evaluated at a finite number of points, one gets a covariance matrix. This can be used as tthe covariance of a Gaussian vector from which samples can be drawn. The values of each entry are a function over the input points of the kernel function.

p79: in a covariance function all the assumptions of the function to be modeled are contained. The kernel defines or rather assesses how near or similar a value of one point is to the one of another. A training point which is close to a test point therefore gives information on how the value at the test point should be.

A covariance function has to be positive semidefinite, i.e. all eigenvalues are greater or equal zero.

stationary kernels : a function of $x - x'$. Stationary because it doesnt matter what value x and x_prime have as a starting value, the difference is always the same. Example: $1100mm - 1000mm = 100mm, 100100mm - 100000mm = 100mm$

squared exponential erklären. infinitely differentiable, is stationary.

p80: A radial basis function is a function which is !only! a function of $r = x - x'$ and whioch is therefore isotropic, i.e. there are no rigid motions. r is like a radius so these functions are called radial basis functions.

Covariance matrices always have to be symmetric, i.e. $k(x - x') = k(x' - x)$, since everything else wouldnt make sense

The so-called Gram matrix is the function containing $x - x'$ evaluated at a set of points x_i. if the function is a covariance funtion, the gram matrix is called covariance matrix.

[Code von Murphy]

```
def squared_exponential(xa, xb, lf, sigf):
    sq_norm = -0.5 * scipy.spatial.distance.cdist(xa, xb, 'sqeuclidean') \
      * (1/lf**2)
    return sigf**2 * np.exp(sq_norm) .
```

positive definiteness : A covariance function has to be positive semidefinite, i.e. all eigenvalues are greater or equal zero. A covariance matrix K is positive semidefinite when $Q(v) = v^T K v \geq 0$. Q is the quadratic form. A kernel function is positive semidefinite, if it only outputs matrices which are positive semidefinite.

characteristic length scale : can be thought of as the number of "upcrossings" in the interval, i.e. the number of crossings of the x axis. can be calculated with eq 4.3. rasmussen. the more upcrossings, the shorter the length scale.

matern kernel

hyperparameters

hyperparameter learning

marginal likelihood: integral of the likelihood times the prior marginal means marginalization over the function values so these dont appear in the equation anymore. in this case it's marginalizing over the test points X so those dont appear anymore.

5.2 Prior before observation

explain, how a sample from the multivariate gaussian canbe drawn (A2 Rasmussen).

5.3 Posterior after observation

p16: The prior is conditioned on the observations. It would also be possible, abut computationally demanding, to sample many many functions out of the GP and only take those which go through the training points.

explain conditioning

6 The Statistical Finite Element Method

The Statistical Generating Model Measuring data always involves a measurement error: One can never measure any physical quantity without some measurement noise. Therefore, the so-called statistical generating model (Cirak) for a vector of n_y measurements $\boldsymbol{y} \in \mathbb{R}^{n_y}$,

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{e} = \rho \boldsymbol{P}\boldsymbol{u} + \boldsymbol{d} + \boldsymbol{e}\,, \tag{6.1}$$

consists of response of the true underlying process $\boldsymbol{z} \in \mathbb{R}^{n_y}$ at the measurement points and the measurement noise $\boldsymbol{e} \in \mathbb{R}^{n_y}$. The true process $\boldsymbol{z}$ can further be expressed as a combination of the used model $\boldsymbol{u} \in \mathbb{R}^{n_u}$, which tries to describe the true process, and a model discrepancy error d which accounts for the model not being a completely accurate representation of the true physics. The model is scaled using the scaling parameter $\rho \in \mathbb{R}$. It can be evaluated at n_u points which are determined by the FEM mesh but only its evaluations at the measurement points are taken into account by using the projection matrix $\boldsymbol{P} \in \mathbb{R}^{n_y \times n_u}$. e is modeled as a multivariate Gaussian

$$\boldsymbol{e} \sim p(\boldsymbol{e}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_e}) \tag{6.2}$$

with zero mean and the covariance matrix $\boldsymbol{C_e} = \sigma_e^2 \boldsymbol{I}$ which adds the measurement noise to each entry of $\boldsymbol{z}$. The model discrepancy error is, in order to account for a possibly complex behavior, modeled as a GP

$$\boldsymbol{d} \sim p(\boldsymbol{d}|\sigma_d, l_d) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_d}) \tag{6.3}$$

which σ_d, l_d the unknown parameters of a squared exponential kernel function. Evaluating it at the measurement positions, the GP is represented by a multivariate Gaussian with a covariance matrix $\boldsymbol{C_d} \in \mathbb{R}^{n_y \times n_y}$.

Basic statFEM procedure statFEM can basically be broken down into three major steps[Cirak, abstract]. The first one is applying Bayesian inversion to the FEM part of the solution. The prior GP is constructed and evaluated before data is introduced. Equations are derived which incorporate the data in order to compute a posterior density for the FEM solution GP. The equations for that solution are dependent on certain hyperparameters. These are estimated in the second step: For each of the hyperparameters a posterior density is calculated using a prior and the marginal likelihood. The third and last step is in principle finding a proper mesh size which is able to describe data the best. This last step may appear rather unimportant since usually in FEM the consensus is that a finer mesh leads to a more accurate result. However, it will be demonstrated that the optimum element size strongly depends on the value of certain hyperparameters estimated in step 2 and that there is a correlation between relatively inaccurate statFEM results and a too small element size.

6.1 Step 1: Posterior of the FEM Forward Model

Assembling the FEM System Matrix The FEM assembly and solving is handled by Fenics. To compute the system matrix and to later solve for the vector of unknowns, the PDE has to be defined in a variational formulation as a boundary value problem. At first, the FEM solver is initiated by defining the domain, the wished number of elements and the kind of elements to be used. In this example, a 1 dimensional mesh with Lagrange basis functions is used.

```
def __init__(self, nMC = 200):
  self.dom_a = 0.0 #domain boundaries
```

```
  self.dom_b = 1.0
  self.ne = 100 #number of elements
  self.mesh = IntervalMesh(self.ne,self.dom_a,self.dom_b) #define mesh
  self.bbt = self.mesh.bounding_box_tree()
  self.coordinates = self.mesh.coordinates() # vertices vector
  self.V = FunctionSpace(self.mesh, "Lagrange", 1) # Function space
```

The variational formulation for the chosen PDE consists of a linear form L and a bilinear form a. These need to be specified individually for Fenics. The boundary condition is set to $u_0 = 0.0$ for both boundaries in the 1D mesh. The system matrix A can be returned by calling `A = assemble(a)`. The boundary conditions are applied to A after assambly.

```
def doFEM(self):
        # Define Dirichlet boundary (x = 0 or x = 1)
        def boundary(x):
                return x[0] < (self.dom_a + DOLFIN_EPS) or x[0] > self.dom_b - DOLFIN_EPS
        # Define boundary condition
        u0 = Constant(0.0)
        self.bc = DirichletBC(self.V, u0, boundary)
        # Define variational problem
        self.u = TrialFunction(self.V)
        self.v = TestFunction(self.V)
        # find index of cell which contains point.
        class DoGP(UserExpression):
                def eval(self, value, x):
                        collisions1st = bbt.compute_first_entity_collision(Point(x[0]))
                        value[0] = f_vals[collisions1st] # f is constant in a cell.
                def value_shape(self):
                        return ()
        f_vals = self.draw_FEM_samples(coordinates)
        f_vals = f_vals[0]
        f = DoGP()

        a = dot(grad(self.u), grad(self.v))*dx #variational Problem
        L = f*self.v*dx

        A = assemble(a)
        b = assemble(L)
        self.bc.apply(A, b)

        U = self.u.vector()
        solve(A, U, b)
        return U, A
```

Generate and sample from source term GP The differential equation is treated from a Bayesian viewpoint: all parameters are random variables. For dependent parameters, such as a source term $f(x)$ dependent on the spatial coordinate, not a single distribution but a Gaussian Process is applied. Therefore prior to solving the FEM linear system, the parameter $\mathcal{GP}(\bar{f}, C_f)$ is sampled. That sample is evaluated for each cell in the FEM mesh which makes it necessary to assemble the system matrix with that in mind.

$\bar{f}$ is found by

solve for C_f

Generate and sample from the diffusion coefficient GP If the diffusion coefficient κ is assumed to be a random variable and dependent on x, it can be modeled as a GP as well and there holds

$$\kappa \sim \mathcal{GP}(\bar{\kappa}, C_\kappa) \tag{6.4}$$

Computing the FEM Prior Having the GPs for the input parameters defined, the FEM system of equations can be solved. With the input parameters defined as a GP, also the FEM solution is going to be a GP. Therefore,

$$u \sim \mathcal{GP}(\bar{u}, C_u) \tag{6.5}$$

is to be calculated. Expressed in terms of the FEM system matrix, there holds

$$\boldsymbol{u} = \boldsymbol{A}^{-1}\boldsymbol{f} \tag{6.6}$$

Because of discretizing, the multivariate Gaussian

$$\boldsymbol{u} \sim p(u) = \mathcal{N}\left(A^{-1}\bar{f}, A^{-1}C_f A^{-T}\right) \tag{6.7}$$

arises. As visible in (6.7), there holds for $\bar{u}$

$$\bar{u} = A^{-1}\bar{f}\,, \tag{6.8}$$

which can be computed using Python by calling

```
def get_U_mean(self):
  u_mean = Function(self.V)
  U_mean = u_mean.vector()
  b_mean = assemble(f_mean*self.v*dx)
  self.bc.apply(b_mean)
  solstd,A = self.doFEM()  # solstd is unused here.
  solve(A,U_mean,b_mean)
  return U_mean .
```

C_u is obtained by calculating

$$A^{-1}C_f A^{-T}\,. \tag{6.9}$$

In Python code there holds

```
def get_C_u(self):
  C_f = self.get_C_f()
  solstd,A = self.doFEM()
  ident = np.identity(len(self.coordinates))
  self.bc.apply(A)
  A = A.array()
  A_inv = np.linalg.solve(A,ident)
```

```
  thresh = 1e-16
  C_u = np.dot( np.dot(A_inv,C_f), np.transpose(A_inv))
  C_u = C_u + 1e-16*(self.sigf**2)*ident
  c_u = np.transpose(self.integratedTestF) * C_u * self.integratedTestF
  return C_u .
```

With $\bar{u}$ and C_u the GP for the prior is completely defined.

find Prior mean and variance

The Projection Matrix P The projection matrix P is constructed by evaluating all FEM ansatz functions of the mesh at the n_y measurement positions. The result is a sparse matrix with most entries zero and only those entries non zero that correspond to the ansatz functions $\Phi_i(\boldsymbol{x})$ associated with the finite element in which the measurement positions lie. This computation for linear ansatz functions can easily be implemented in Python using Fenics:

```
def getP(self,y_points):
  ny = len(y_points)
  P = np.zeros((ny, self.ne+1), dtype = float) #with ne the number of elements
  for j,point in enumerate(y_points):
    x = np.array([point])
    x_point = Point(x)
    bbt = self.mesh.bounding_box_tree()
    cell_id = bbt.compute_first_entity_collision(x_point)
    cell = Cell(self.mesh, cell_id)
    coordinate_dofs = cell.get_vertex_coordinates()
    values = np.ones(1, dtype=float)
    for i in range(self.el.space_dimension()):
      phi_x = self.el.evaluate_basis(np.array(i),x
        ,np.array(coordinate_dofs),cell.orientation())
      P[j,cell_id+i] = phi_x
  self.Pu = np.dot(P,self.U_mean)
  return P
```

calculate C_e

derive marginal likelihood

estimate Hyperparameters for C_d and scaling factor

calculate C_d

6.1.1 Inference of the Posterior Density

To infer the posterior density, observed data is necessary. These can be obtained by either actually measuring a physical process or by generating synthetic data which allows developing and improving the statFEM model already before having to set up an experiment and take real measurements beforehand. Sampling from a synthetic source or from the real physical response yields the observation vector

$$\boldsymbol{y} = z + e \tag{6.10}$$

with z the real response and e the measurement error which is also a part of $\boldsymbol{y}$ for the synthetic observations. It can be seen that the model mismatch error $\boldsymbol{d}$ is, as opposed to eq. (), not part of the equation because it is only part of the FEM modeling error. The data is observed on multiple measurement points n_y throughout the domain. It can be chosen if the data is assumed to be deterministic or to be random. For the deterministic case only one value is measured for each measurement point. In the random case many measurements are taken per point what leads to a probability distribution for each point. Now, Bayes' rule can be applied to update the prior with the observations. The result of this operation is

$$p(u|y) = \mathcal{N}(\bar{u_{|y}}, C_{u|y}) \tag{6.11}$$

with

$$\bar{u_{|y}} = C_{u|y}\left(\rho P^T (C_d + C_e)^{-1} y + C_u^{-1}\bar{u}\right) \tag{6.12}$$

and

$$C_{u|y} = \left(\rho^2 P^T (C_d + C_e)^{-1} P + C_u^{-1}\right)^{-1} \tag{6.13}$$

6.1.2 find Posterior mean and variance

6.1.3 Modeling Parameters of the PDE

6.1.4 Probabilistic FEM Prior

The prior before observation is obtained by computing the FEM response u_h of the PDE given all parameters. The random parameters are modeled as GPs. Thereby result a mean $\bar{u_h}$ and a covariance matrix C_u. For this being the prior, the GP hyperparameters are chosen deterministically. The covariance matrix is obtained by calculating $C_u = A^{-1} C_{par} A^{-T}$ with A the FEM system matrix and C_{par} the covariance matrix of the respective random parameter's GP. The mean can easily be determined by calculating the deterministic mean of the FEM by using only the GP's mean. The resulting response u_h is, since it is discretized and not continuous, a multivariate Gaussian distribution. To check convergence of the FEM model, it can be compared to the exact analytically determined system response u. It is assumed that there is a difference between the exact and the calculated FEM response to the true system response. That difference will be reduced by updating the prior with observations. With the basic FEM code set up for a random source term and a deterministic diffusion coefficient, $\bar{u}$ can be obtained by calling

C_u can be obtained by calling

For numerical reasons a small nugget is added to the diagonal of C_u. This does not significantly change the result but allows using the Cholesky decomposition. With $\bar{u}$ and C_u at hand the prior GP is defined as $u \sim \mathcal{GP}(\bar{u}, C_u)$. To check if the covariance matrix is correct, a MC approximation $\bar{u}_{MC}$ of the mean can be computed by evaluating the GP n_{MC} times and taking the average. As visible in Figure 6.1.4, the error of $\bar{u}_{MC}$ converges to zero with a slope of 2 on a logarithmic scale. Therefore, C_u indeed has a mean of $\bar{u}$. The MC samples can be drawn via

```
def samplePrior(self):
  U_mean = self.get_U_mean()
  C_u = self.get_C_u()
  priorGP = np.random.multivariate_normal(
    mean = U_mean, cov=C_u,
    size=self.nMC)
  return priorGP
```

Code for calculating u direct calculation MC approximation from GP samples -error bands

Code for calulating Cu direct calucaltion MC approximation error bands

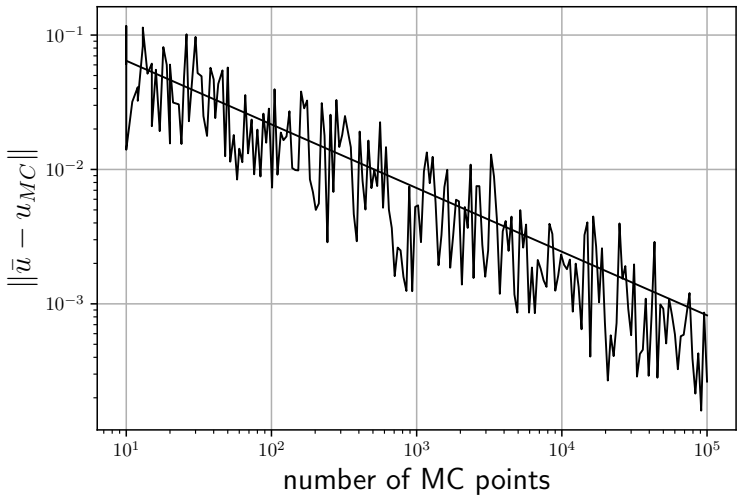

Figure 6.1: Convergence of the error norm for the FEM prior mean

6.2 Step 2: Estimation of Hyperparameters

Below it was already stated that the hyperparameters for computing the model discrepancy covariance matrix $\boldsymbol{C_d}$ are not known. The scaling factor ρ is not known, as well. The measurement data can be used to estimate these parameters, collected in the hyperparameter vector $\boldsymbol{w}$, with Bayes' rule

$$p(w|y) = \frac{P(y|w)p(w)}{\int P(y|w)p(w)dw} \tag{6.14}$$

where the denominator, which only serves as a normalization constant, can be dropped if only a point estimate of the hyperparameters is needed. This approach is indeed non-bayesian because no probability density of the hyperparameters is computed. For multiple measurements y_i with a total number of n_o there holds, because it is assumed that individual measurements are independent [Rasmussen p.9],

$$P(Y|w) = \prod_{i=1}^{n_o} p(\boldsymbol{y_i}) \tag{6.15}$$

which replaces $P(y|w)$. If no prior knowledge of the parameters is available, it is possible to use an uninformative prior $p(w) = 1$. The equation then basically states that $p(w|Y) \propto P(Y|w)$ which means that maximizing $P(Y|w)$, i.e. the likelihood to measure Y given a set of hyperparameters, yields the optimal vector of hyperparameters for the given data. The marginal likelihood $P(y|w)$, marginal because ?!?!?!, can be expressed as

$$p(y|w) = \mathcal{N}(\rho \boldsymbol{P}\bar{u}, \boldsymbol{C_d} + \boldsymbol{C_e} + \rho^2 \boldsymbol{P C_u P}^T) \tag{6.16}$$

because, as stated above, all components of the statistical generating model are Gaussian. The equation for the corresponding PDF reads

$$p(y|w) = \frac{1}{(2\pi)^{n/2}|K|^{1/2}} \exp\left(-\frac{1}{2}(y - \rho \boldsymbol{P}\bar{u})^T \boldsymbol{K}^{-1}(y - \rho \boldsymbol{P}\bar{u})\right) \tag{6.17}$$

with

$$K = \boldsymbol{C_d} + \boldsymbol{C_e} + \rho^2 \boldsymbol{P C_u P}^T \ . \tag{6.18}$$

To make further calculations easier the exponential function can be removed by taking the negative log of the function (Rasmussen):

$$-\log p(y|w) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{K}| + \frac{1}{2}(\rho \boldsymbol{P}\bar{u} - y)^T \boldsymbol{K}^{-1}(\rho \boldsymbol{P}\bar{u} - y) \tag{6.19}$$

This also improves numerical stability because products of possibly very small factors can become smaller than machine accuracy. The log replaces products with sums. Because of taking the negative log, (6.19) now needs to be minimized instead of maximized.

Cholesky Decompositon [Gentle 1998 p.93-95] For numerical stability reasons $\boldsymbol{K}^{-1}$ should not be calculated directly. Instead, the linear system is solved. Additionally, if K is symmetric and positive-semidefinite, computing the lower triangular matrix using the Cholesky decomposition

$$K = LL^T \tag{6.20}$$

makes the calculation quicker and yields the also needed determinant of K as a by-product. A generic linear system is defined as $Ax = b$. It is solved for x, so the inverse of A is needed. In this case for (6.19) there holds $A = K$ and $b = y$, since the inverse of K is meant to be found. Defining $x = K^{-1}y$ there holds

$$K \times K^{-1} \times y = y \tag{6.21}$$

With L the lower triangular matrix of K obtained by

```
L = scipy.linalg.cho_factor(K)
```

the linear system can now be solved for $x = K^{-1}y$ with

```
K_inv_y = scipy.linalg.cho_solve(L, y) .
```

The determinant of any triangular matrix is defined as the product of its diagonal entries [http://www.math.lsa.umich.edu/ ho
Therefore the lower triangular matrix L can be used to efficiently calculate the determinant of K. There holds

$$detK = detL detL^T = detL^2 \tag{6.22}$$

and therefore

$$detK = \left(\prod_{i=1}^{n_y} L_{i,i}\right)^2 \tag{6.23}$$

Since in (6.19) the log likelihood is used, (6.23) can be simplified to a sum:

$$\begin{aligned}\log detK &= \log detL^2 \\ &= 2\sum_{i=0}^{n_y} \log L_{i,i}\end{aligned}$$

6.2.1 Minimization of the Negative log-Likelihood

The best possible set of hyperparameters w for (6.19) to be as small as possible has to be found. That can be achieved by using a gradient based optimizer or by sampling the parameter space with, e.g., a MCMC approach.

MCMC Markov Chain Monte Carlo (MCMC) is a very efficient sampling method. Just as standard MC, it can be used to draw samples from a given distribution. The advantage of MCMC is now that an algorithm finds a relatively small set of samples which is able to describe the distribution very accurately. The result is a distribution for the hyperparameters of which e.g. the mean can be deduced. One common algorithm is the metropolis algorithm. The metropolis algorithm is a special case of MCMC for a symmetric proposal distribution. Metropolis algorithm as code.

L-BGFS L-BGFS is a gradient based optimizer. It doesn't deliver a distribution for the hyperparameters but only a point estimate. For the optimization with L-BGFS the derivative of (6.19) is needed. It can be computed as follows.

6.3 Step 3: Estimating the Optimum Mesh Parameters

7 Application of statFEM in Vibroacoustics

7.1 Simple 1D example

Choice of PDE The Poisson equation

$$-\nabla \cdot \mu(x)\nabla u(x) = f(x) \tag{7.1}$$

is chosen as the governing equation. It is an elliptic partial differential equation (PDE). In this work it is used as a simple 1D example to illustrate how the statistical FEM and especially the Gaussian Process Regression works. The most standard form of it does not, contrary to this example, include $\mu(x) \in \mathbb{R}^+$ which is the diffusion coefficient dependent on the spatial variable x. The right-hand side consists of the source term $f(x) \in \mathbb{R}$. Both $\mu(x)$ and $f(x)$ are the free parameters in this case. The equation is solved for the unknown $u(x)$ in the domain $\Omega = (0, 1)$ with the boundary condition $u(x) = 0$ on $x = 0$ and $x = 1$. For a first example there holds $\mu(x) = 1$. $f(x)$ is modeled as a Gaussian Process [Cirak]

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(\bar{f}(\boldsymbol{x}), c_f(\boldsymbol{x}, \boldsymbol{x}')\right) . \tag{7.2}$$

Construction of the GP The mean function of the GP is set to $\bar{f}(x) = 1.0$. For the covariance function at first a squared exponential kernel

$$c_f(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2l_f^2}\right) \tag{7.3}$$

is used with the standard deviation $\sigma_f = 0.1$ and the length scale $l_f = 0.4$. In Python the kernel is directly implemented as a function which takes two lists and the parameters as input variables [Murphy]:

Here, the parameters are fixed but in a later example a method on how to infer the optimum position for these will be studied.

Having prepared the mean function and the kernel, which can be considered the prior in a GP regression setting, a sample of the GP can be drawn. For this points have to be chosen on which the kernel is evaluated. These are the test points and, according to [Cirak], correspond to the center of the FEM cells. This implies that there are as many test points as FEM cells and therefore the coordinates of the FEM mesh can directly be used to compute the covariance matrix. For that (7.3) is evaluated at $\boldsymbol{x} = \boldsymbol{x}'$ with $\boldsymbol{x}$ the vector of test points.

A GP has the marginalization property: if you sample it at a finite number of points it yields a multivariate Gaussian distribution $\mathcal{N}(\bar{\boldsymbol{f}}, \boldsymbol{C}_f)$ with $\bar{\boldsymbol{f}}$ the mean vector and $\boldsymbol{C}_f$ the covariance matrix while still describing the underlying continuous sample. According to [Rasmussen] sampling from a multivariate Gaussian distribution works as follows: To obtain n samples from the prior, at first n samples of a standard normal distribution, also called Gaussian white noise, $e \sim \mathcal{N}(0, 1)$ have to be drawn. Computing the Cholesky decomposition of the covariance matrix $\boldsymbol{C}_f = LL^T$, which can also be thought of as taking the square root of a matrix, yields the lower triangular matrix L. Samples can now easily be drawn from $\boldsymbol{f} = \bar{\boldsymbol{f}} + Le$ which is a multivariate Gaussian distribution with a mean $\boldsymbol{f}$ and a covariance $\boldsymbol{C}_f$.

Sample: Normal distribution times std dev. GP with n points is basically a multivariate gaussian with n dimensions. therefore we need the std dev. for a univariate gaussian thats .

For the observations made in a later step it is assumed that there is some measurement noise. This is modeled as an added variance σ_n^2 on the diagonal of the prior covariance matrix. An additional effect of this added noise is the improved numerical stability of the covariance matrix which is important for computing the Cholesky decomposition.

7.2 1D Vibroacoustics Example

Helmholtz Equation. Looks similar to the Poisson equation but the right side differs. It's the eigenvalue problem for the laplace operator.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

8 Conclusion and Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

8.1 Section 1

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non

odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Bibliography

[1] Campolina, B.: Vibroacoustic modelling of aircraft double-walls with structural links using Statistical Energy Analysis (SEA). Acoustics [physics.class-ph]. Université de Sherbrooke; Université Pierre et Marie Curie - Paris VI (2012)

[2] Langer, S.C., Blech, C.: Cabin noise prediction using wave-resolving aircraft models. Proc. Appl. Math. Mech., Vol. 12 (1) (2019): e201900388. doi:10.1002/pamm.201900388

[3] Yaghoubi, V., Marelli, S., Sudret, B., Abrahamsson, T.: Sparse polynomial chaos expansions of frequency response functions using stochastic frequency transformation, Probabilistic Engineering Mechanics, volume 48, pages 39-58 (2017)