Series 1

1. Differentiation in $L^2_{\text{per}}([0,1])$

In class we investigated the inverse problem related to differentiation of continous periodic functions. In this exercise we investigate this challenge on a different pair of Hilbert spaces. Let

$$X := \left\{ f \in L^2_{\text{per}}([0,1]) \,;\, \int_0^1 f\,dx = 0 \right\} \quad \text{and} \quad Y := \left\{ g \in H^1_{\text{per}}([0,1]) \,;\, g(0) = 0 \right\}$$

be equipped with the L^2– and the H^1–norm respectively, and consider the direct operator $T : X \to Y$ defined by $(Tf)' = f$.

a) Show that T belongs to $\mathcal{L}(X,Y)$. Moreover, show that T^{-1} is in $\mathcal{L}(Y,X)$.

b) Show that the inverse problem associated to T is ill-posed, if the data g is perturbed with some noise, which can be controlled in the L^2–norm only.
`Hint:` Let $\mathcal{F} : X \to \ell^2(\mathbb{C})$ be the operator which, when evaluated on a function $v \in X$, returns the coefficient sequence $(\hat{v}_\ell)_{\ell\in\mathbb{Z}}$ of the Fourier series of v, that is,

$$\mathcal{F} = (\hat{v}_\ell)_{\ell\in\mathbb{Z}} = \left(\int_0^1 v(x) e^{-2\pi i \ell x}\,dx \right)_{\ell\in\mathbb{Z}} .$$

With the Parseval's identity it can be shown that $\mathcal{F}$ is an isometry. Characterize the operator $M : \ell^2(\mathbb{C}) \to \mathcal{F}(Y)$, which is the counterpart of T for Fourier series sequences, that is,

$$M := \mathcal{F} \circ T \circ \mathcal{F}^{-1} .$$

Then, consider a noisy data of the form

$$g^\delta := g + \delta e^{2\pi i k x} , \quad \text{for some } k \in \mathbb{N} ,$$

and, with the help of $\mathcal{F}$ and M, show that,

$$\|f - f^\delta\|_X = \Omega(k) , \text{ despite } \|g - g^\delta\|_X = \delta .$$

c) Show that the difference quotient operator

$$(R_h g)(x) := \frac{g(x+h) - g(x-h)}{2h} , \quad h > 0 ,$$

belongs to $\mathcal{L}(Z,X)$ for $Z = L^2_{\text{per}}(]0,1[)$ and that $\|R_h\|_{Z\to X} \to \infty$ for $h \to 0$.

Bitte wenden!

Solution:

Differentiation in $L^2_{per}([0,1])$

a) $T \in \mathcal{L}(X,Y)$

- linearity is trivial
- continuity: $\|Tf\|^2_{H^1} = \|f\|^2_{L^2} + \|Tf\|^2_{L^2} \leq C\|f\|^2_{L^2}$
  $\|Tf\|^2_{L^2} \leq C\|f\|^2_{L^2}$ Poincaré ineq

□

b)
$$\begin{array}{ccc} X & \xrightarrow{F} & \ell^2(\mathbb{C}) \\ T\downarrow & & \downarrow \Pi \\ Y & \xrightarrow[F]{} & F(Y) \end{array}$$
commuting diagram

What is Π?

$w := Tv \quad \Rightarrow \quad w' = v$ in L^2

$\Rightarrow F(w') = F(v)$ in ℓ^2

$$\Rightarrow \int_0^1 w' e^{-2\pi i \ell x}\,dx = \underbrace{\int_0^1 v e^{-2\pi i \ell x}\,dx}_{=\hat{v}_\ell} \qquad \forall \ell \in \mathbb{Z}$$

$$\overset{\text{IbP}}{=} \underbrace{w e^{-2\pi i \ell x}\Big|_0^1}_{=0 \text{ by periodicity}} + 2\pi i \ell \underbrace{\int_0^1 w e^{-2\pi i \ell x}\,dx}_{\hat{w}_\ell} = 2\pi i \ell\, \hat{w}_\ell$$

$$\Rightarrow \quad \hat{w}_\ell = \frac{1}{2\pi i \ell}\hat{v}_\ell \qquad \forall \ell \in \mathbb{Z}$$

$$\Rightarrow \boxed{\Pi\left((\hat{v}_\ell)_{\ell\in\mathbb{Z}}\right) = \left(\frac{\hat{v}_\ell}{2\pi i \ell}\right)_{\ell\in\mathbb{Z}}}, \text{ and } \Pi^{-1}\left((\hat{w}_\ell)_{\ell\in\mathbb{Z}}\right) = \left(2\pi i \ell \hat{w}_\ell\right)_{\ell\in\mathbb{Z}}$$

Now, $\|g - g^\delta\|^2_{L^2} = \|\delta e^{2\pi i k \cdot}\|^2_{L^2} = |\delta|^2 \|F(e^{2\pi i k \cdot})\|^2_{\ell^2(\mathbb{C})}$

$$= \delta^2 \|(\ldots,0,\ldots,0,1,0,\ldots 0,\ldots)\|^2_{\ell^2(\mathbb{C})} = \delta^2,$$

(1 in the k^{th} pos.)

but $\|\underbrace{f - f^\delta}_{= T^{-1}(g-g^\delta) = F^{-1}\circ\Pi^{-1}\circ F(g-g^\delta)}\|^2_{L^2} = \delta^2 \|(\ldots,0,\ldots,0,2\pi i k,0,\ldots,0,\ldots)\|^2_{\ell^2(\mathbb{C})}$

$$= \delta^2 4\pi^2 k^2 = \mathcal{O}(k^2)$$

$\Rightarrow$ The problem is ill-posed

Siehe nächstes Blatt!

c) $(R_h g)(x) := \dfrac{g(x+h) - g(x-h)}{2h}$

- linearity is trivial
- continuity

$$\left\| \frac{g(x+h) - g(x-h)}{2h} \right\|_{L^2} \leq \frac{1}{4h^2} \left(\| g(x+h) \|_{L^2} + \| g(x-h) \|_{L^2} \right)$$

$$= \frac{1}{2h^2} \| g \|_{L^2} \leq \frac{1}{2h^2} \| g \|_{H^1}$$

(= by periodicity) □

Moreover

$$\| R_h(\cos) \|_{L^2}^2 = \left\| \frac{\cos(x+h) - \cos(x-h)}{2h} \right\|_{L^2}^2 =$$

$$= \frac{1}{4h^2} \left(2 \| \cos \|_{L^2}^2 - 2 \int_0^{\cdot} \cos(x+h) \cos(x-h) \, dx \right)$$

$$= \frac{1}{4h^2} \left(2 \| \cos \|_{L^2}^2 - \int_0^{\cdot} \cos(2x) + \cos(2h) \, dx \right)$$

$2 \| \cos \|_{L^2}^2 = \frac{1}{4}(2 + \sin(2)) - \sin(1)\cos(1) \approx 0.27$

$$\geq \frac{0.27 + \cos(2h)}{4h^2} \xrightarrow{h \to 0} \infty$$

Bitte wenden!

2. Pressure Gauges

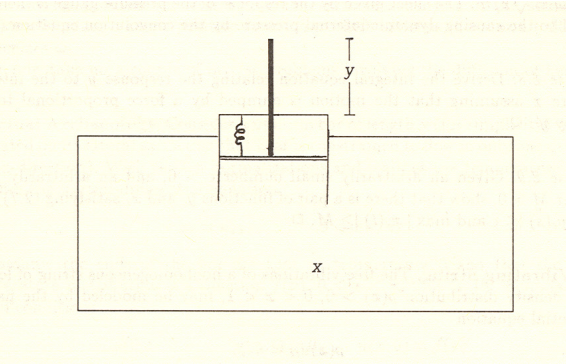

Abbildung 1: Sealed vessel with an attached gauge.

Consider a sealed vessel with an attached gauge, see Fig. 2. The pressure in the vessel changes due to some external heating/cooling. The gauge is an undamped spring-loaded piston. The piston mass is m and the spring constant k. The cross-sectional area of the piston is 1 so that force and and pressure are equal. The gauge displacement $y(t)$ above the equilibrium and the internal dynamic pressure $x(t)$ then satisfy

$$my''(t) + ky(t) = x(t), \quad y(0) = 0, y'(0) = 0$$

with the initial conditions chosen for simplicity.

a) Using the Laplace transform, show that

$$y(t) = \frac{1}{\omega m} \int_0^t \sin(\omega(t-\tau))x(\tau)\, d\tau \quad \text{with} \quad \omega = \sqrt{k/m}. \tag{1}$$

b) Consider the inverse problem of determining the pressure $x(t)$ from the gauge displacement. Given an arbitrarily small $\varepsilon > 0$ and arbitrarily large $M > 0$, show that a pair of functions $y_\varepsilon(t), x_\varepsilon(t)$ exists satisfying (1) with $\max|y_\varepsilon(t)| \leq \varepsilon$ and $\max|x_\varepsilon(t)| \geq M$ for $t \in [0, T]$. One example suffices.

Siehe nächstes Blatt!

Solution:

Pressure Gauges

a) Set $Y(s) := \mathcal{L}(y)(s)$, then

$$m s^2 Y(s) + k Y(s) = \underbrace{X(s)}_{= \mathcal{L}(x)(s)}$$

because $\mathcal{L}(f') = s\,\mathcal{L}(f) - f(0)$

Thus $$Y(s) = \frac{X(s)}{m s^2 + k} = \frac{1}{m} \frac{X(s)}{s^2 + k/s} =$$

$$= \frac{1}{mw} \frac{w}{s^2 + w^2} X(s)$$

$\Rightarrow$ by linearity of $\mathcal{L}$ and the convolution thm, and since $\mathcal{L}(\sin(at)) = \frac{a}{s^2 + a^2}$,

$$y(t) = \frac{1}{wm} \int_0^t \sin\left(w(t-\tau)\right) x(\tau)\, d\tau \qquad \square$$

b) set $y(t) := \delta \sin^2(\ell t) \rightarrow y(0) = 0$

$y'(t) = 2\delta\ell \sin(\ell t)\cos(\ell t) \rightarrow y'(0) = 0$

} initial conditions satisfied

$x(t)$ can be computed with the ODE

$$x(t) = m y''(t) + k y(t)$$

$$= 2\delta\ell^2 \left(1 - 2\sin^2(\ell t)\right) + k\delta \sin^2(\ell t)$$

$\Rightarrow \max_t |x| \approx O(\ell)$ while $\max_t |y(t)| = \delta$

Bitte wenden!

3. Deconvolution

In this exercise, we wish to implement a deconvolution similar to the gravimetry example from the lecture. We begin by theoretically deriving a method and then implement it and conduct some numerical experiments. Templates and some basic functions (e.g. for the generation of quadrature points) are available on the course website.

Consider the Hilbert spaces $X = Y = L^2([-1,1])$, the elements $f \in X$ and $g \in Y$ and the continuous linear operator $T : X \to Y$ defined by

$$T : f \mapsto \int_{-1}^{1} \mathcal{K}(\xi - x) f(x)\,\mathrm{d}x. \tag{2}$$

a) Show that T is linear and continuous for $\mathcal{K} \in L^1([-1,1])$.

Solution: Linearity follows trivially from the linearity of the integral. The result from the lecture concerning convolution can be used if f is replaced by the function $\tilde{f} : \mathbb{R} \to \mathbb{R}$:

$$\tilde{f}(x) = \begin{cases} f(x) & , \text{ if } -1 \le x \le 1 \\ 0 & , \text{ else} \end{cases},$$

giving us the result $||Tf||_{L^2} \le ||\mathcal{K}||_{L^1} ||f||_{L^2}$.

b) Given $f(x) = \exp(x^2)$ and $\mathcal{K}(\sigma) = \exp(-\sigma^2)$, compute the analytical solution $g(\xi)$ to the *forward problem* $g = Tf$:

$$g(\xi) = \int_{-1}^{1} \mathcal{K}(\xi - x) f(x)\,\mathrm{d}x. \tag{3}$$

Solution:

$$\begin{aligned} \int_{-1}^{1} \mathcal{K}(\xi - x) f(x)\,\mathrm{d}x &= \int_{-1}^{1} e^{-(\xi - x)^2} e^{x^2}\,\mathrm{d}x = e^{-\xi^2} \int_{-1}^{1} e^{2x\xi - x^2} e^{x^2}\,\mathrm{d}x \\ &= e^{-\xi^2} \int_{-1}^{1} e^{2x\xi}\,\mathrm{d}x = \frac{e^{-\xi^2}}{\xi} \frac{\left(e^{2\xi} - e^{-2\xi}\right)}{2} \\ &= \frac{e^{-\xi^2}}{\xi} \sinh(2\xi) =: g(\xi). \end{aligned}$$

c) We now consider the *inverse problem* of determining $f \in X$ given a datum $g \in Y$. To this end, we introduce a reconstruction R_N with discretization parameter N based on a spectral Galerkin approach. This consists of approximating the unknown solution f by an element f_N of a finite-dimensional trial space V_N spanned by the $N+1$ basis functions $\{\varphi_j\}_{j=0}^{N}$

$$f_N(\xi) = \sum_{j=0}^{N} \mu_j \varphi_j(\xi). \tag{4}$$

Derive the weak formulation of (3) using V_N as the test space and the expansion of f from (4). Reformulate the problem as a linear system of equations $A\vec{\mu} = \vec{\psi}$.

Siehe nächstes Blatt!

Solution: Variational formulation yields $\vec{\psi} = (\psi_k)_{k=0,\dots,N}$ and $A = (A_{kj})_{k,j=0,\dots,N}$:

$$\underbrace{\int_{-1}^{1} g(\xi)\varphi_k(\xi)\,\mathrm{d}\xi}_{\psi_k} = \sum_{j=0}^{N} \underbrace{\left[\int_{-1}^{1}\int_{-1}^{1} \mathcal{K}(\xi - x)\varphi_j(x)\,\mathrm{d}x\,\varphi_k(\xi)\,\mathrm{d}\xi\right]}_{A_{kj}} \mu_j.$$

d) As our choice for a concrete basis $\{\varphi_j\}$ we will use the $N+1$ Legendre polynomials $\{L_0(x), \dots, L_N(x)\}$ on $[-1, 1]$, defined by the recursion

$$(n+1)L_{n+1}(x) = (2n+1)xL_n(x) - nL_{n-1}(x)$$

with first two polynomials

$$L_0(x) = 1 \qquad L_1(x) = x.$$

Read up on the Legendre polynomials and their properties, eg. on Wikipedia.

e) Formulate (analytically) the quadrature method to approximate the entries A_{kj} with Gauss-Legendre quadrature using M points. For the trivial kernel $\mathcal{K} = 1$, what do you expect the matrix entries to be?

Solution: Replacing the integrals by sums and weights yields

$$\begin{aligned} A_{kj} &= \int_{-1}^{1}\int_{-1}^{1} \mathcal{K}(x-\xi)\,L_j(\xi)\,L_k(x)\,\mathrm{d}\xi\,\mathrm{d}x \\ &\approx \sum_{m=0}^{M}\sum_{n=0}^{M} \mathcal{K}(\xi_m - \xi_n)\,L_j(\xi_n)\,L_k(\xi_m)\,w_m\,w_n \\ &= \vec{w}\cdot\left[\mathcal{K}(\vec{\xi}\cdot\vec{e}^{\,T} - \vec{e}\cdot\vec{\xi}^{\,T})\,.*\left(L_j(\vec{\xi})\cdot L_k(\vec{\xi})^T\right)\right]\cdot\vec{w}, \end{aligned}$$

where the last line assumes $\vec{\xi}$ to be the column vector of quadrature points, $\vec{w}$ the column vector of quadrature weights and $.*$ component-wise multiplication.

For $\mathcal{K} = 1$, the matrix should be zero everywhere except for the entry $k = j = 1$, corresponding to the constant polynomials. This is because the integration is conducted over two independent variables and $L_j(x)$ integrates to 0 for $j > 0$.

f) Write a Matlab function that accepts a function handle to $g(x)$ and the discretization parameter N and returns the coefficient vector $\vec{\mu}$ that solves the linear system derived above, i.e.:

```
function mu = IP_spectral_galerkin(g,K,N)
  % solve inverse problem using spectral Galerkin approach
  %   g: function handle representing the data
  %   K: function handle representing the kernel
  %   N: discretization param. Number of basis fcts is N+1
  %   mu: coefficient vector of Lagrange basis of length N+1
end
```

Bitte wenden!

You may use the Matlab function `gauleg(a,b,M)` from the file `gauleg.m` to construct the Gauss-Legendre quadrature rule with M points for an integral over the interval $[a, b]$. $M = 20$ is a good value. The evaluation of the Legendre basis at all points in the row vector `x` is implemented in the function `legendre(N,x)` in the file `legendre.m`. See the course website for a download link for these files.

Solution: The following Matlab code is called in the main function `run_specgal.m`:

```
function mu = IP_spectral_galerkin_legendre(g,K,N)
  % solve inverse problem using spectral Galerkin approach with a
      Legendre basis.
  % The matrix is computed using full quadrature.
  %   g:  function handle accepting and returning a scalar value
  %         (represents the data)
  %   K:  function handle accepting and returning a scalar value
  %         (represents the kernel)
  %   N:  discretization param. Number of basis fcts is N+1
  %   mu: coefficient vector of Lagrange basis of length N+1

  % N+1: number of basis functions

  %% M: number of quadrature points
  M = 20;

  %% Gauss-Legendre quadrature rule
  quadrule = gauleg(-1,1,M);
  %quadrule = clenshawcurtis(-1,1,M);

  %% Legendre polynomial basis (up to L_{N+1_} evaluated in [-1,1]
  L = legendre(N,quadrule.x); % goes from L_0 to L_N (N+1 elements)
  L = diag(sqrt(((2*(0:N)+1)/2)))' * L; % normalization

  %% construct right-hand side vector
  phi = zeros(N+1,1);
  for i=1:N+1
    phi(i) = quadrule.w' * (g(quadrule.x).*L(i,:)');
  end

  %% construct matrix
  A = zeros(N+1);
  [X,Y] = meshgrid(quadrule.x,quadrule.x);
  for k=1:N+1
    for j=1:N+1
      % quadrature
      A(k,j) = (quadrule.w .* L(k,:)')' * K(Y-X) * (quadrule.w .* L(j,:)
          ');
    end
  end

  %% solve linear system
  mu = A\phi;

end
```

To run this code, execute

```
%% some definitions
K = @(sigma) exp(-sigma.^2);
g = @(x) iff(x,x==0,@(x)1,@(x) exp(-x.^2) ./x .* sinh(2*x));
f = @(xi) exp(xi.^2);

%% number of basis functions
```

Siehe nächstes Blatt!

```
N_basis = 2;
x_plot = linspace(-1,1,100)';

mu = IP_spectral_galerkin(g,K,N_basis);
fx = eval_legendre(mu,x_plot);

%% f) plot solution for N=2
figure(1)
plot(x_plot,g(x_plot),'k'); hold on;
plot(x_plot,f(x_plot),'--r');
plot(x_plot,fx,'b');
legend('g (data)', 'exact sol f', ['numerical sol f, N=', int2str(N_basis
    )])
print -painters -depsc -r600 sol_specgal.eps
```

g) Write a Matlab function that accepts the computed coefficient vector of length $N+1$ and a vector of (many) evaluation points and computes the solution f_N at these points, i.e.:

```
function f = eval_legendre(mu,x)
  % mu: coefficient vector of Legendre basis
  % x: vector of evaluation points of length
  % f: value of f_N at all points in the vector x
end
```

`Hint:` The solution should look like the following:

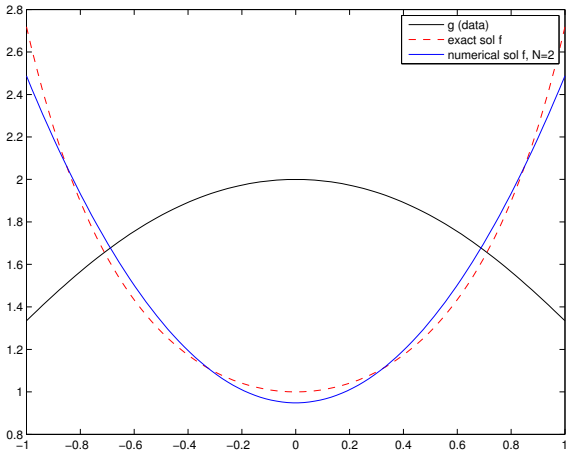

Abbildung 2: Solution of the inverse problem with $N = 2$ (3 basis functions L_0, L_1, L_2).

Solution: The following script does this:

```
function f = eval_legendre(mu,x)
  % g) computes linear combination of legendre basis functions
  % mu: coefficient vector of Legendre basis
  % x: vector of evaluation points of length
  % f: value of f_N at all points in the vector x
  x = x(:)';
  mu = mu(:);
  N = size(mu,1)-1;
  L = legendre(N,x);
  L = diag(sqrt(((2*(0:N)+1)/2)))' * L; % normalization
```

Bitte wenden!

```
11
   % linear combination
13   f = mu'*L;
end
```

h) Use overkill quadrature (eg. Gaussian quadrature with 10^4 points) to compute the L^2-norm of the discretization error $e_N(x) = f(x) - R_N(g)$ caused by the use of a finite-dimensional trial space V_N. Plot this error against N for $N = 1, 2, \ldots, 15$. What kind of convergence do you observe? Is this expected? Measure the convergence rate.

Solution: The error computation can be done as follows:

```
function err = err_discretization(mu,exactfct)
2   % mu:        coefficient vector
   % exactfct: function handle that can evaluate the exact solution
4
   % overkill quadrature rule
6   quadrule = gauleg(-1,1,1e4);

8   % evaluate numerical solution
   f_num = eval_legendre(mu,quadrule.x)';
10
   % evaluate exact solution
12   f_ex = exactfct(quadrule.x);

14   % L2 error: square of differences
   err = sqrt( quadrule.w' * (f_num-f_ex).^2 );
16 end
```

To run this code, execute

```
%% h) compute discretization error for various values for N
2 N_vals = [1:15];
discerr = [];
4 for N_basis=N_vals
   fprintf('N_basis = %d\n',N_basis);
6   mu = IP_spectral_galerkin(g,K,N_basis);
   discerr = [discerr, err_L2(mu,f)];
8 end

10 %% measure rate
maxind=10;
12 p = polyfit(N_vals(1:maxind),log(discerr(1:maxind)),1);
fprintf('Convergence rate of discretization error is: %f\n', p(1));
14 p(2) = 0.5;
y = exp(polyval(p,N_vals(1:maxind)));
16
%% plot discretization error
18 figure(2)
semilogy(N_vals,discerr,'o-'); hold on;
20 semilogy(N_vals(1:maxind),y,'--r');
legend('discretization error vs N', ['rate:' num2str(p(1))]);
22 ylabel('||disc err||_{L^2}');
xlabel('N');
24 print -painters -depsc -r600 err_specgal.eps
```

In Figure 3 we observe exponential convergence, since the error decays linearly in the `semilogy` plot. Exponential convergence is typical for spectral methods. The

Siehe nächstes Blatt!

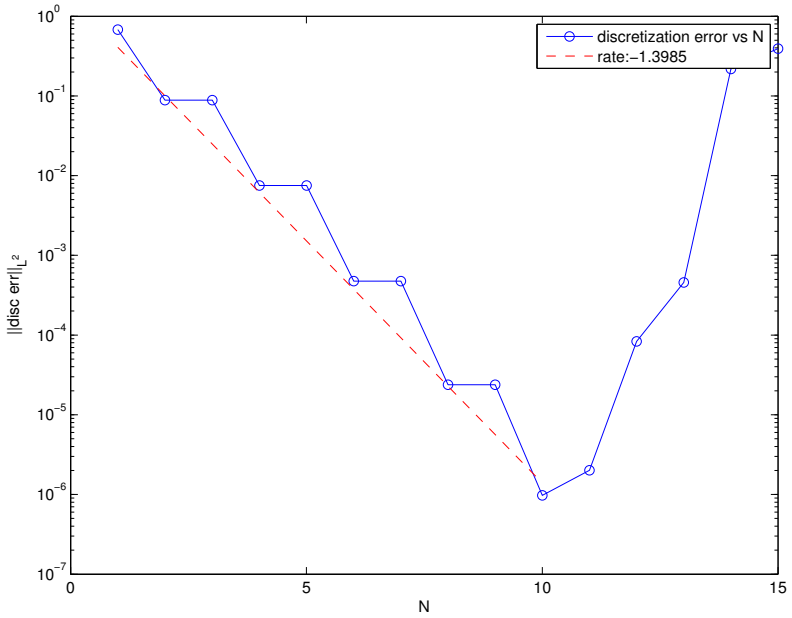

Abbildung 3: L^2 norm of $e_N(x) = f(x) - R_N(g)$.

divergence for $N > 10$ is due to the ill-posedness of the problem. In linear algebra terms, the condition of the matrix A diverges for large N.

i) Consider the perturbed data $g^\delta(x) = g(x) + \delta \sin(n\pi x)$, where δ is the noise parameter. Decompose the total error $f - R_N(g^\delta)$ into a reconstruction error and a data noise error. What convergence behavior in N of the total error with fixed δ do you expect?

Solution:

$$\begin{aligned} ||f - R_N(g^\delta)||_{L^2} &\overset{R_N \text{ lin.}}{=} ||f - R_N(g) + R_N(g) - R_N(g^\delta)||_{L^2} \\ &\overset{\Delta}{\leq} ||f - R_N(g)||_{L^2} + ||R_N(g - g^\delta)||_{L^2} \\ &\leq ||f - R_N(g)||_{L^2} + ||R_N||_{\mathcal{L}(Y,X)} \cdot \underbrace{||g - g^\delta||_{L^2}}_{\leq C_2 \delta} \end{aligned}$$

The bound involving δ comes from $||g - g^\delta||_{L^2}^2 = \int_0^1 (\delta \cdot \sin(\pi x))^2 \, \mathrm{d}x \leq \delta^2 \cdot 2$.

Assume fixed δ. For small N, one expects to see the convergence of the reconstruction error, whereas for large N the operator norm of R_N should diverge.

j) For each fixed $\delta = 10^{-k}, k = 0, 2, 4, 6, 8, 10$, compute the numerical approximation of the solution to the inverse problem and use overkill quadrature (eg. Gaussian quadrature with 10^4 points) to compute the L^2-norm of the total error $e_N^\delta = f(x) - R_N(g^\delta)$ and plot it vs. N for $N = 1, 2, \ldots, 12$ for $n = 1$. Can you verify your prediction from subproblem i)?

Bitte wenden!

Solution: See Figure 4 for the convergence plot. The following code uses the `err_L2.m` function to compute the error:

```
%% j) perturbation
figure(3);
n = 1;
N_vals = [1:12];
delta_vals = 10.^-(0:2:10);
perturberr = zeros(size(N_vals,2),size(delta_vals,2));
i=1; j=1;
for N_basis = N_vals
  fprintf('N_basis = %d\n',N_basis);
  j=1;
  for delta = delta_vals
    fprintf('  delta = %f\n',delta);
    gdelta = @(x) g(x) + delta*sin(n*pi*x);
    mu = IP_spectral_galerkin(gdelta,K,N_basis);
    perturberr(i,j) = err_L2(mu,f)
    j=j+1;
  end
  i=i+1;
end

semilogy(repmat(N_vals',1,size(perturberr,2)),perturberr),
ylabel('||total err||_{L^2}')
legend('delta=1e0','delta=1e-2','delta=1e-4','delta=1e-6','delta=1e-8','
    delta=1e-10')
print -painters -depsc -r600 err_perturbation.eps
```

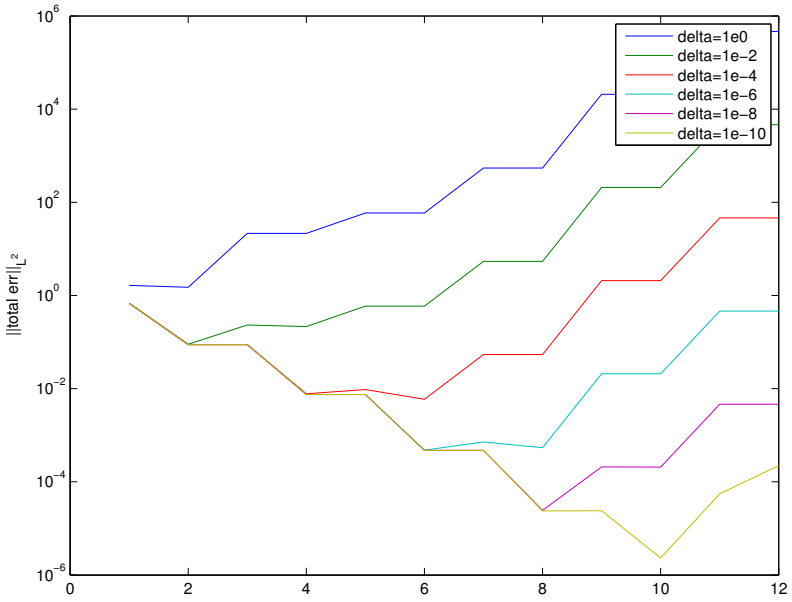

Abbildung 4: L^2 norm of $e_N^\delta(x) = f(x) - R_N(g^\delta)$ for various δ.

k) In order to gain more insight into the divergence of the reconstruction operator norm, plot the 2-norm condition of the matrix A vs. N for $N = 1, 2, \ldots, 12$. How does the condition grow as a function of N?

Siehe nächstes Blatt!

Solution: The 2-norm condition grows super-exponentially as a function of N! Thus, small perturbations in the data lead to large perturbations in the resulting solution to the linear system.

```
function c = condition(K,N)
  % compute the condition of the Galerkin matrix constructed with N+1
      basis functions
  %   K:  function handle accepting and returning a scalar value
  %         (represents the kernel)
  %   N:  discretization param. Number of basis fcts is N+1
  %   c:  2-norm condition of A

  %% M: number of quadrature points
  M = 20;

  %% Gauss-Legendre quadrature rule
  quadrule = gauleg(-1,1,M);
  %quadrule = clenshawcurtis(-1,1,M);

  %% Legendre polynomial basis (up to L_{N+1_} evaluated in [-1,1]
  L = legendre(N,quadrule.x); % goes from L_0 to L_N (N+1 elements)
  L = diag(sqrt(((2*(0:N)+1)/2)))' * L; % normalization

  %% construct matrix
  A = zeros(N+1);
  [X,Y] = meshgrid(quadrule.x,quadrule.x);
  for k=1:N+1
    for j=1:N+1
      % quadrature
      A(k,j) = (quadrule.w .* L(k,:)')' * K(Y-X) * (quadrule.w .* L(j,:)
          ');
    end
  end

  c = cond(A);
end
```

To call this code, use

```
%% k) 2-norm condition of A
c = zeros(size(N_vals));
i=1;
for N_basis = N_vals
  c(i) = condition(K,N_basis);
  i = i+1;
end

figure();
semilogy(N_vals,c,'-o'); hold on;
semilogy(N_vals,exp(N_vals),'r'); hold on;
semilogy(N_vals,exp(N_vals.^(exp(1)/2)),'g'); hold on;
xlabel('N')
ylabel('Condition of A');
legend('cond_2(A)','e^N','e^{N^{e/2}}','Location','NorthWest')
print -painters -depsc -r600 cond_vs_N.eps
```

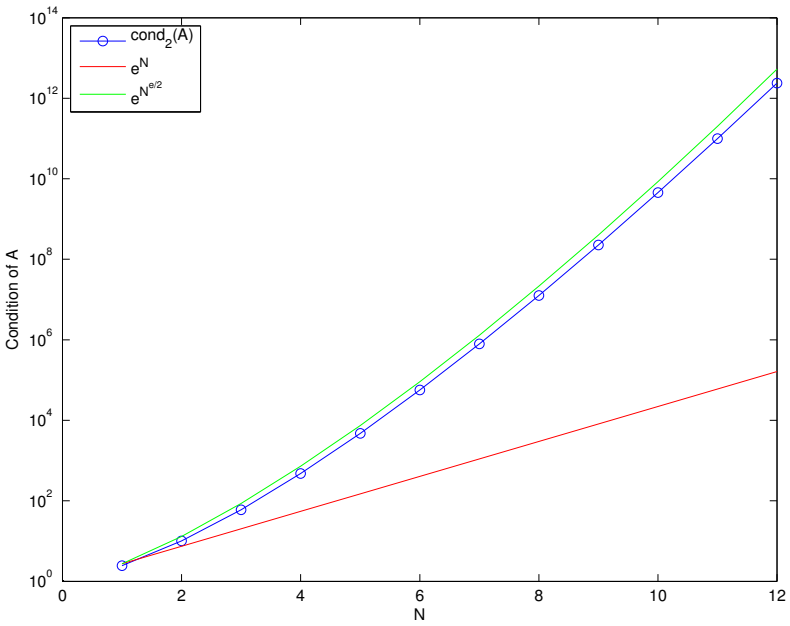

Abbildung 5: 2-norm condition of A vs. N. The condition seems to grow like $\sqrt{e^{Ne}}$.