S. Mishra
L. Scarabosio
J. Sukys

Spring Term 2014

Numerical Methods for Partial Differential Equations

ETH Zürich
D-MATH

Homework Problem Sheet 1

Introduction. All the problems are devoted to study some general concepts of partial differential equations considering the one-dimensional Poisson equation as model problem.

Problem 1.1 Green's function for Poisson equation

We consider the 1d Poisson equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -u''(x) &= f(x), \quad \forall x \in \Omega = (0,1) \\ u(0) &= u(1) = 0. \end{aligned} \tag{1.1.1}$$

In the lecture, we learned that the solution u to this equation can be obtained using the Green's function:

$$u(x) = \int_\Omega G(x,y) f(y)\,\mathrm{d}y, \quad x \in \Omega$$

(in case of nonhomogeneous boundary conditions, we would have also bounday terms).
The Green's function for this problem is defined as:

$$G_x(y) := G(x,y) = \begin{cases} y(1-x) & \text{for } 0 \le y \le x \\ x(1-y) & \text{for } x \le y \le 1. \end{cases} \tag{1.1.2}$$

Its expression was obtained by simple integration by parts of (1.1.1).

However, Green's function can be introduced in a more general way for any PDE.
Before this, we need to recall the concept of Dirac delta distribution.
The delta distribution $\delta_x(y) := \delta(x-y)$ centered at the point y can be thought, at first glance (rigorous definition goes beyond the scope of this course), as a function

$$\delta(x-y) = \begin{cases} 0 & \text{for } y \neq x \\ \infty & \text{for } y = x. \end{cases} \tag{1.1.3}$$

Physically, it describes an impulse (e.g. the force excited when you hit strongly a table with an hammer).
The delta distribution has the property of "selecting" function values:

$$\int_\Omega \delta(x-y) f(y)\,\mathrm{d}y = f(x) \quad \text{for any } f \in C([0,1]) \tag{1.1.4}$$

(from which $\int_\Omega \delta(x-y)\,\mathrm{d}y = 1$ follows).
Then the Green's function $G(x,y) := G_x(y)$ for a PDE is defined as the impulse response, i.e.

as the solution to the PDE when at the right-handside we have $f(y) = \delta_x(y)$. For the Poisson's equation, this means:

$$\begin{aligned} -G_x''(y) &= \delta_x(y), \quad \forall y \in \Omega = (0,1) \\ G_x(0) &= G_x(1) = 0. \end{aligned} \tag{1.1.5}$$

(1.1a) Show that the Green's function as given by (1.1.2) solves (1.1.5).

HINT: When a function $g(y)$ has a jump at a point x, then, it is not differentiable in that point. However, a generalized derivative can be defined, so that $g'(y)|_{y=x} = \delta_x(y)|_{y=x}$.
Consequently, the derivative of a piecewise constant function $g(y)$ with jump at x is $g'(y) = \delta_x(y)$.

Solution: From (1.1.2), we get that

$$G_x'(y) \begin{cases} (1-x) & \text{for } 0 \le y \le x \\ -x & \text{for } x \le y \le 1. \end{cases}$$

and thus, using the hint:

$$G_x''(y) = \delta_x(y).$$

Moreover, it is clear from (1.1.2) that the boundary conditions are fulfilled.

(1.1b) Show that the function u defined as

$$u(x) = \int_0^1 G(x,y) f(y)\,\mathrm{d}y, \quad x \in (0,1), \tag{1.1.6}$$

satisfies (1.1.1).

HINT: Formula (1.1.4) may be useful.

Solution: Differentiating (1.1.2), we get:

$$u''(x) = \int_0^1 \frac{\partial^2}{\partial x^2} G(x,y) f(y)\,\mathrm{d}y = -\int_0^1 \delta_x(y) f(y)\,\mathrm{d}y = -f(x),$$

where for the last equality we used (1.1.4).
Moreover, u given by (1.1.6) satisfies the homogeneous Dirichlet boundary conditions because, from (1.1.5), the Green's function does.

(1.1c) Implement a function

```
function val = Green(x,y)
```

which accepts in input two column arrays `x` and `y` of grid points, and returns in the matrix `val` the Green's function (1.1.2) evaluated at those grid points, with the convention that `val(i,j)=Green(x(i),y(j))`.

Solution: See listing 1.1 for the code.

Listing 1.1: Implementation for `Green`

```
1 function val = Green(x,y)
```

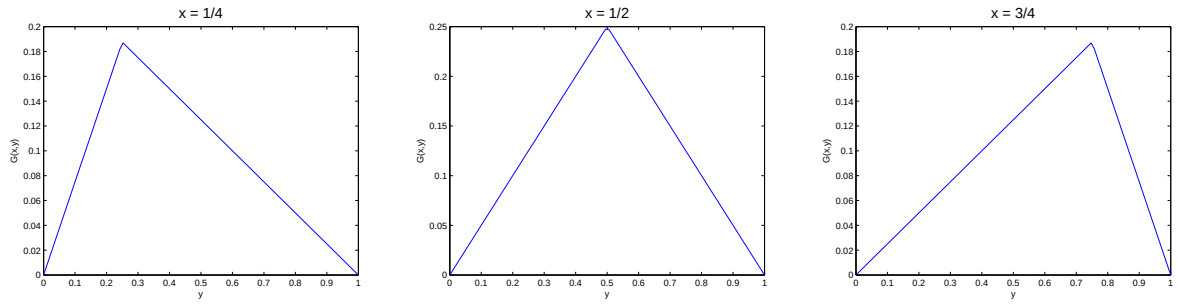

Figure 1.1: Plots for subproblem (1.1d).

```
for j=1:length(y)
    for i=1:length(x)
    if y(j)<=x(i)
        val(i,j) = y(j)*(1-x(i));
    else
        val(i,j) = x(i)*(1-y(j));
    end
    end
end
```

(1.1d) Use the routine of subproblem (1.1c) to plot the Green's function for $x = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ ($y \in [0, 1]$).

Solution: See Figure 1.1.

(1.1e) Implement a function

```
function u = PoissonGreen(x,FHandle)
```

which accepts as input a column vector `x` of equispaced grid points $\{x_0 = 0, x_1, \ldots, x_N, x_{N+1} = 1\}$; and a function handle `FHandle` to the right-handside f of (1.1.1); in output it returns a column vector `u` containing the value of the solution u to (1.1.1) computed at the points `x`. The solution is computed using the formula (1.1.2).
To compute the integral, use the composite trapezoidal quadrature rule:

$$\int_0^1 g(x)\,\mathrm{d}x \approx \frac{g(0)h}{2} + h\sum_{i=1}^{N} g(x_i) + \frac{g(1)h}{2}, \tag{1.1.7}$$

where $h = |x_1 - x_0|$.
Use the array `x` itself as quadrature points.

HINT: Use the implementation from subproblem (1.1c) for the Green's function.

Solution: See listing 1.2 for the code.

Listing 1.2: Implementation for `PoissonGreen`

```
function u = PoissonGreen(x,FHandle)

greenval = Green(x,x);
h = x(2)-x(1);
u = zeros(length(x),1);

for i=1:length(x)
    u(i) = FHandle(x(1))*greenval(1,i)*h/2 +
       sum(FHandle(x(2:end-1)).*...
         greenval(2:end-1,i),1)*h+FHandle(x(end))*greenval(end,i)*h/2;
end
```

We are now going to discretize (1.1.1) using the Finite Differences method, namely the centered finite differences.
To this aim, we subdivide the interval $[0,1]$ in $N+1$ subintervals using equispaced grid points $\{x_0 = 0, x_1, \dots, x_N, x_{N+1} = 1\}$.
The discretized problem can be written as a linear system

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{L}, \tag{1.1.8}$$

where $\boldsymbol{A}$ is a $N \times N$ matrix, $\boldsymbol{L}$ a $N \times 1$ vector and $\boldsymbol{u}$ the $N \times 1$ vector containing of unknowns $u(x_j)$, $j = 1, \dots, N$, the value of the function u at the grid points.
Let us denote by $h = |x_1 - x_0|$ the meshsize.

(1.1f) Refresh your mind on what we saw in class on the Finite Difference scheme and in particular on the central finite differences.
Write the matrix A and the right-handside L. For the right-handside, write it in terms of a generic force term $f(x)$ in (1.1.1).

Solution: We have:

$$\boldsymbol{A} = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & \dots & -1 & 2 \end{pmatrix} \qquad \boldsymbol{L} = h^2 \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{pmatrix}$$

(1.1g) Implement a function

```
function A = PoissonMatrix(N)
```

which computes the matrix $\boldsymbol{A}$ for (1.1.8). Here the input parameter `N` denotes the number of *internal* grid points.

Solution: See listing 1.3 for the code.

Listing 1.3: Implementation for `PoissonMatrix`

```
function A = PoissonMatrix(N)

e = ones(N,1);
A = spdiags([-e 2*e -e], -1:1, N, N);
```

(1.1h) Implement a function

```
function L = PoissonRHS(FHandle,N)
```

to compute the right-handside $\boldsymbol{L}$ for (1.1.8). The input parameter `FHandle` is the function handle for the right-handside $f(x)$ and `N` is again the number of interior grid points.

Solution: See listing 1.4 for the code.

Listing 1.4: Implementation for `PoissonRHS`

```
function L = PoissonRHS(FHandle,N)

h = 1/(N+1);

L = h^2*FHandle((h:h:1-h)');
```

(1.1i) Implement the function

```
function uh = PoissonSolve(FHandle,N)
```

to solve the Poisson problem (1.1.1).
The input parameters are as in subproblem (1.1h). The output `uh` is the array $\{u_h(x_j)\}_{j=1}^N$ containing the approximate value of the solution u at the interior grid points $\{x_j\}_{j=1}^N$.

HINT: Use the routines from subproblems (1.1g) and (1.1h).

Solution: See listing 1.5 for the code.

Listing 1.5: Implementation for `PoissonSolve`

```
function uh = PoissonSolve(FHandle,N)

A = PoissonMatrix(N);
L = PoissonRHS(FHandle,N);
uh = zeros(N+2,1);
uh(2:end-1) = A\L;
```

(1.1j) Run the routine `PoissonSolve` for $f(x) = \sin(2\pi x)$ and $N = 50$ and plot the solution.

Solution:

See Fig. 1.2 for the plot.

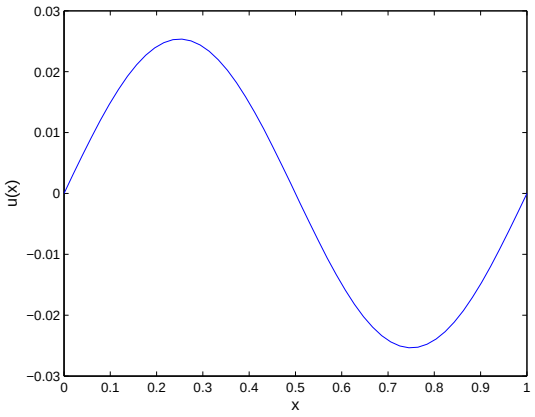

Figure 1.2: Plot for subproblem (1.1j)

We saw in the lecture that the centered finite difference schemes satisfies is stable and consistent, and thus it converges to the exact solution u to (1.1.1) when the mesh is refined.
Here we are going to test the convergence of our scheme through a convergence study.

(1.1k) Write a function

```
function [h err] = Poissoncvg(FHandle)
```

to perform the convergence study. The input argument is a function handle to the right-hanside $f(x)$. As output the routine returns the array `h`=$[h_1, \ldots, h_6]$ of meshsizes and the array `err`=$[e_1, \ldots, e_6]$ of computed errors.
As error between the discrete solution u_h and the exact solution u, we consider the maximum norm error $\|u - u_{h_i}\|_\infty = \max_{x\in[0,1]} |u - u_h|$, $i = 1, \ldots, 6$; we can compute this error just in an approximate way: we consider a very fine grid with meshsize $h_{ref} = \frac{1}{2^{10}}$ and grid points $x_0 = 0, x_1 = h_{ref}, \ldots, x_{N+1} = 1$ and approximate the maximum norm by

$$\|u - u_h\|_\infty \approx \max_{x_0 \ldots x_{N+1}} |u(x_i) - u_h(x_i)| \tag{1.1.9}$$

The standard steps for a convergence study then:

1. compute the *reference solution*: compute the exact solution u to (1.1.1) that you can obtain by call to the routine `PoissonGreen` from subproblem (1.1e) on the grid with meshsize h_{ref};
2. start from a meshsize $h_1 = \frac{1}{4}$, corresponding to $N_1 = 3$ interior grid points;
3. compute the discrete solution u_{h_1} to (1.1.1);
4. compute the error $e_1 \approx \|u - u_{h_1}\|_\infty$; inside each mesh interval, consider the linear interpolant for u_h;
5. refine the grid, considering $h_2 = \frac{h_1}{2} = \frac{1}{8}$ and repeat the algorithm from step 3;

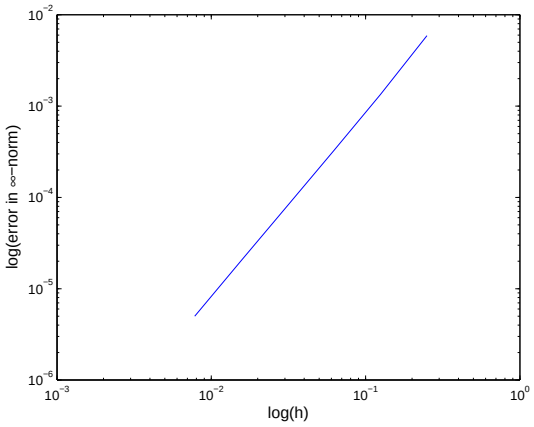

Figure 1.3: Convergence plot for subproblem (1.1l)

6. repeat the previous step till $h_6 = \frac{h_1}{2^5}$.

Solution: See listing 1.6 for the code.

Listing 1.6: Implementation for `Poissoncvg`

```
function [h,err] = Poissoncvg(FHandle)

xref = linspace(0,1,2^10+1);
u = PoissonGreen(xref',FHandle);

h = [1/4];
err = [];

for i=1:6

    uh = PoissonSolve(FHandle,1/h(i)-1);
    x = linspace(0,1,1/h(i)+1);
    err = [err max(abs(linterp(x,uh,xref)'-u))];
    h = [h h(i)/2];
end
h = h(1:end-1);
```

(1.1l) Run the routine `Poissoncvg` for $f(x) = \sin(2\pi x)$.
Make a double logarithmic plot of the errors $e_1, \ldots, e_6$ versus the meshsizes $h_1, \ldots, h_6$. What do you observe? Which is the order of convergence? **Solution:**

See Fig. 1.3 for the plot. As expected, the error goes to zero as the mesh is refined. The curve in the loglog plot is a line with slope 2. This means that $\|u - u_h\|_\infty \approx Ch^2$, for a constant $C > 0$ and 2 is the convergence order, as we expected from the theory.

Problem 1.2 The Poisson equation with Neumann boundary conditions

We consider the Poisson equation with homogeneous Neumann boundary conditions:

$$-u''(x) = f(x), \quad \forall x \in \Omega = (0,1) \tag{1.2.1}$$
$$u'(0) = u'(1) = 0,$$

$f \in C^0([0,1])$.

(1.2a) Show that

$$\int_0^1 f(x)\,\mathrm{d}x = 0 \tag{1.2.2}$$

is a necessary condition to have a solution.

Solution: From (1.2.2) we have:

$$\int_0^1 f(x)\,\mathrm{d}x = -\int_0^1 u''(x)\,\mathrm{d}x = -u'(1) + u'(0) = 0.$$

(1.2b) Show that (1.2.1) does not have a unique solution.

HINT: Suppose that a function $u(x)$ is a solution and consider $v(x) = u(x) + c$, $c > 0$.

Solution: We have that v solves the differential equation:

$$-v''(x) = -u''(x) = f(x)$$

and also the boundary conditions:

$$v'(0) = u'(0) = 0 \quad v'(1) = u'(1) = 0.$$

Thus, if u is a solution, any other function $v(x) = u(x) + c$, $c > 0$ is.

(1.2c) Show that, if (1.2.2) is fulfilled, then (1.2.1) has always a solution $u \in C^2([0,1])$. Moreover, show that the solution is unique if we require

$$\int_0^1 u(x)\,\mathrm{d}x = 0 \tag{1.2.3}$$

Solution: Integrating (1.2.1) once:

$$-\int_0^y u''(z)\,\mathrm{d}z = -u'(y) = \int_0^y f(z)\,\mathrm{d}z,$$

where we used the boundary conditions. Futher integrating we get:

$$-\int_0^x u'(y)\,\mathrm{d}y = -u(x) + u(0) = \int_0^x \int_0^y f(z)\,\mathrm{d}z\,\mathrm{d}y.$$

This means that, under (1.2.2), there exists always a solution given by

$$u(x) = u(0) - \int_0^x \int_0^y f(z)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x.$$

If additionally we require (1.2.3), then

$$\int_0^1 u(x)\,\mathrm{d}x = u(0) - \int_0^1 \int_0^x \int_0^y f(z)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x = 0$$

and thus

$$u(0) = \int_0^1 \int_0^x \int_0^y f(z)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x$$

and the solution is unique.

We are now going to analyse the discretization (1.2.1) through centered differences.
Consider a partition $\{x_0 = 0, x_1, \ldots, x_N, x_{N+1} = 1\}$ of the interval $[0, 1]$, with equispaced points and meshsize h. The discretized equation can be written as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{L}, \tag{1.2.4}$$

where $\boldsymbol{A}$ is a $(N+2) \times (N+2)$ matrix, $\boldsymbol{L}$ is a vector of length $N+2$ and $\boldsymbol{u}$ is the vector of unknowns $\{u_h(x_0), \ldots u_h(x_{N+1})\}$, where u_h denotes the discrete solution.

(1.2d) Compute the matrix $\boldsymbol{A}$ and the right-handside $\boldsymbol{L}$.

Solution:

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & -1 & 2 & -1 \\ 0 & \ldots & \ldots & \ldots & -1 & 1 \end{pmatrix} \qquad \boldsymbol{L} = \begin{pmatrix} 0 \\ h^2 f(x_1) \\ \vdots \\ \vdots \\ h^2 f(x_N) \\ 0 \end{pmatrix}$$

(1.2e) Show that the matrix $\boldsymbol{A}$ is singular. This is because the nonunicity of the solution that we have seen in subproblem (1.2b) is reflected on the discrete case.

Solution: If we change the sign to the last row and then sum up the rows (or the columns), we get the zero-vector, that is the rows/columns are linearly dependent.

(1.2f) We have seen in subproblem (1.2c) that under the constraint (1.2.3) the solution to (1.2.1) is unique.
Apply the composite trapezoidal quadrature rule to (1.2.3) to get an equation in terms of the entries of $\boldsymbol{u}$, i.e. a constraint on the discrete level.

HINT: The composite trapezoidal quadrature rule for a generic function g is:

$$\int_0^1 g(x)\,\mathrm{d}x \approx \frac{g(0)h}{2} + h\sum_{i=1}^{N} g(x_i) + \frac{g(1)h}{2}, \tag{1.2.5}$$

Solution: Let us denote by $\{u_0, \ldots, u_{N+1}\}$ the entries of $\boldsymbol{u}$.
The resulting equation is

$$\frac{u_0 h}{2} + h\sum_{i=1}^{N} u_i + \frac{u_{N+1}h}{2} = 0$$

or equivalently, multiplying by $\frac{2}{h}$,

$$u_0 + 2\sum_{i=1}^{N} u_i + u_{N+1} = 0.$$

(1.2g) Combining the equation we have got from subproblem (1.2f) with the system of equations (1.2.4), we have a linear system of $N+3$ equations for $N+2$ unknowns.
However, these equations are not linearly independent since the rows of $\boldsymbol{A}$ are not. It can be shown (and actually it is expected from subproblem (1.2c)) that the equation we get from subproblem (1.2f) is linearly independent from the rows of $\boldsymbol{A}$. Then, summing up this equation with the first equation in (1.2.4), we get a system

$$\tilde{\boldsymbol{A}}\boldsymbol{u} = \tilde{\boldsymbol{L}} \tag{1.2.6}$$

of $N+2$ linearly independent equations for $N+2$ unknowns.
Write the matrix $\tilde{\boldsymbol{A}}$ and the vector $\tilde{\boldsymbol{L}}$ (the latter in terms of a generic right-handside $f(x)$). **Solution:** They are given by

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} 2 & 1 & 2 & \dots & 2 & 1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 1 & -1 \end{pmatrix} \qquad \tilde{\boldsymbol{L}} = \boldsymbol{L}$$

(1.2h) Write a routine

```
function A = PoissonNeuMatrix(N)
```

to implement the matrix $\tilde{\boldsymbol{A}}$. Here N is the number of *interior* grid points.

Solution: See listing 1.7 for the code.

Listing 1.7: Implementation for `PoissonNeuMatrix`

```
function A = PoissonNeuMatrix(N)

e = ones(N+2,1);
A = spdiags([-e 2*e -e],-1:1,N+2,N+2);

A(1,3:end-1)=2*ones(1,N-1);
A(1,2)=1;
A(1,end)=1;
A(end,end)=-1;
A(end,end-1)=1;
```

(1.2i) Write a routine

```
function L = PoissonNeuRHS(FHandle,N)
```

to implement the right-handside $\tilde{\boldsymbol{L}}$. Here `FHandle` is a function handle to a generic right-handside $f(x)$ and N is the number of *interior* grid points.

Solution: See listing 1.8 for the code.

Listing 1.8: Implementation for `PoissonNeuRHS`

```
function L = PoissonNeuRHS(FHandle,N)

h = 1/(N+1);

L = h^2*FHandle((0:h:1)');
L(1) = L(1)/h;
L(end) = L(end)/h;
```

(1.2j) Write a routine

```
function uh = PoissonNeuSolve(FHandle,N)
```

to solve the system (1.2.6) and store the solution in the vector `uh`.

Solution: See listing 1.9 for the code.

Listing 1.9: Implementation for `PoissonNeuSolve`

```
function uh = PoissonNeuSolve(FHandle,N)

A = PoissonNeuMatrix(N);
L = PoissonNeuRHS(FHandle,N);
uh = A\L;
```

(1.2k) Consider $f(x) = \sin(2\pi x)$.
Verify that this right-handside satisfies the condition (1.2.2).
Compute the analytic solution for this right-handside.

Solution: The condition (1.2.2) is satisfied, because

$$\int_0^1 \sin(2\pi x)\,\mathrm{d}x = -\frac{1}{2\pi}\cos(2\pi x)\Big| = 0.$$

Integrating (1.2.1) once we get:

$$-(u'(x) - u'(0)) = -\frac{1}{2\pi}\cos(2\pi x) + \frac{1}{2\pi}$$

and thus, using the boundary conditions:

$$u'(x) = \frac{1}{2\pi}\cos(2\pi x) - \frac{1}{2\pi}.$$

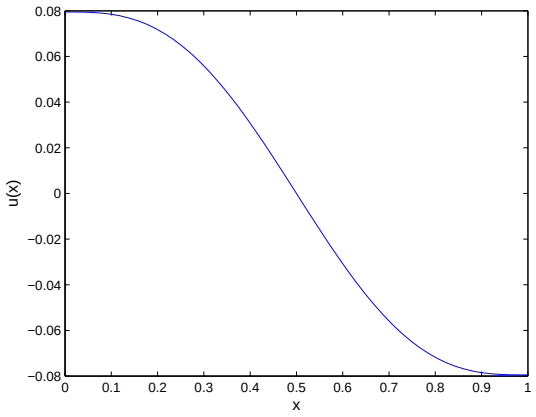

Figure 1.4: Plot for subproblem (1.2l)

Integrating on more time:

$$u(x) - u(0) = \frac{1}{4\pi^2}\sin(2\pi x) - \frac{1}{2\pi}x;$$

applying the constraint (1.2.3) to this last equation we get that $u(0) = \frac{1}{4\pi}$ and thus the solution to (1.2.1) is

$$u(x) = \frac{1}{4\pi^2}\sin(2\pi x) - \frac{1}{2\pi}x + \frac{1}{4\pi}.$$

(1.2l) Use the routine `PoissonNeuSolve` to compute the solution to (1.2.1) with $f(x) = \sin(2\pi x)$. Use $N = 50$ and make a plot of the solution.
You can compare then this plot with a plot of the exact solution computed in (1.2k) to have a first check of your routines (although an exhaustive check would require a convergence study).

Solution:

See Fig. 1.4 for the plot.

Problem 1.3 Stability property for the Poisson equation

The aim of this problem is to better understand what does the *stability* mean for a differential equation. We will do it consider the Poisson equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -u''(x) &= f(x), \quad \forall x \in \Omega = (0,1) \\ u(0) &= u(1) = 0. \end{aligned} \tag{1.3.1}$$

for $f \in C([0,1])$.

In applications the exact source term $f(x)$ is not available. What is available is some perturbation of it

$$\tilde{f}(x) = f(x) + \eta(x), \tag{1.3.2}$$

where $\eta(x)$ is some noise introduced, for example, by some measurement error.
Then, what can be actually computed is the solution $\tilde{u}$ to the perturbed system

$$\begin{aligned} -\tilde{u}''(x) &= \tilde{f}(x), \quad \forall x \in \Omega = (0,1) \\ \tilde{u}(0) &= \tilde{u}(1) = 0. \end{aligned} \tag{1.3.3}$$

(1.3a) Show that

$$\|u - \tilde{u}\|_\infty \leq \frac{1}{8}\|f - \tilde{f}\|_\infty \tag{1.3.4}$$

Solution: We know from the lecture that any solution v to the Poisson equation (1.3.1) with right-handside g satisfies the estimate

$$\|v\|_\infty \leq \frac{1}{8}\|g\|_\infty.$$

Subtracting (1.3.3) to (1.3.1), we get that $u - \tilde{u}$ satisfies the equation

$$\begin{aligned} -(u-\tilde{u})''(x) &= f(x) - \tilde{f}(x), \quad \forall x \in \Omega = (0,1) \\ (u-\tilde{u})(0) &= (u-\tilde{u})(1) = 0. \end{aligned}$$

Thus, taking $v = u - \tilde{u}$ and $g = f - \tilde{f}$, we get (1.3.4).

(1.3b) Consider now that we want to compute the solution to (1.3.3) numerically and denote by $\tilde{u}_h$ the discrete solution. We are interested in estimating how well $\tilde{u}_h$ approximates the exact solution u to the *unperturbed* problem (1.3.3).
Show that the following estimate holds:

$$\|u - \tilde{u}_h\|_\infty \leq \frac{1}{8}\|\eta\|_\infty + \|\tilde{u} - \tilde{u}_h\|_\infty. \tag{1.3.5}$$

Note that here we are not making any assumption on the discretization scheme.

Solution: Applying the triangle inequality we have:

$$\|u - \tilde{u}_h\|_\infty \leq \|u - \tilde{u}\|_\infty + \|\tilde{u} - \tilde{u}_h\|_\infty.$$

The result then follows from equation (1.3.4) and remembering that $f - \tilde{f} = \eta$.

(1.3c) Suppose that $\eta(x) = \delta \sin(2\pi x)$, for some $\delta > 0$, so that

$$\tilde{f} = f + \delta \sin(20\pi x). \tag{1.3.6}$$

Let us consider $\delta = 10^{-1}, 10^{-2}, 10^{-3}$ and suppose that for each of this values we made a convergence study for $\tilde{u}_h$ considering the error $\|u - \tilde{u}_h\|_\infty$. The convergence plots are shown in Fig. 1.5
Observe and compare the plots and comment on them:

- Why, for meshsize h small there is a plateau?
- How does the plateau change with δ? Why?

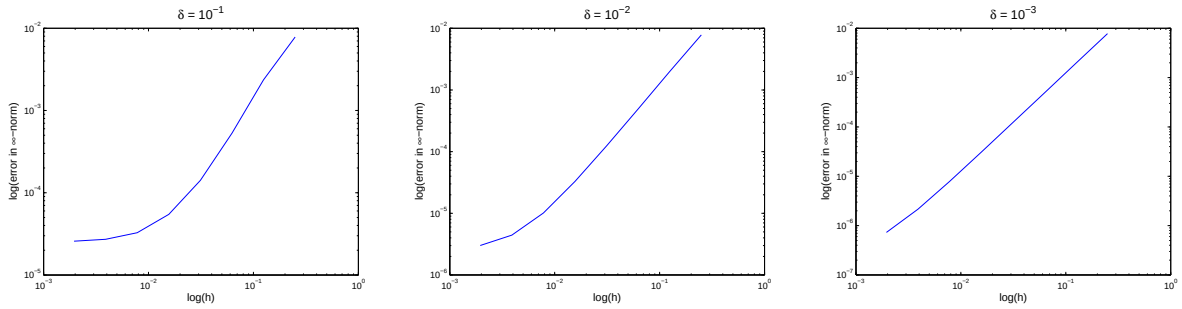

Figure 1.5: Plots for subproblem (1.3c).

Solution: From (1.3.5) we see that the first addend on the left-handside goes to zero as $\|\eta\|_\infty$ does, while, assuming a convergent discretization scheme, the second addend goes to zero as the mesh is refined.
Here we have $\|\eta\|_\infty = \delta$.
In the region that the discretization error dominates over the error introduced by the noise, we can observe convergence.
On the contrary, when the error introduced by the noise dominates over the discretization error, the total error $\|u - \tilde{u}_h\|_\infty$ does not go to zero anymore although the discretization error does. This is the reason why we see a plateau in the first two plots.
The plateau happens at a smaller error as the noise norm δ gets smaller, till the last plot where we don't see it for the meshsizes considered for the convergence study.

Published on March 3.

To be submitted on March 17.

Last modified on March 18, 2014