Bayesian Scientific Computing
Exercises of Day 1

Daniela Calvetti, Erkki Somersalo

Case Western Reserve University
Department of Mathematics, Applied Mathematics and Statistics

Lyngby, December 2019

Problem 1: Estimating the strength of one's prior

A person tells you that he can read other people's minds and guess a number from 1 to 100 they are thinking. He has a record of succeeding 8 times out 10. You suspect the claim and in your mind, you give a certain probability x for such a gift, but decide to give the poor devil a try. He guesses correctly the number you thought. You are still not convinced, that is, even after the positive demonstration, you still think that he is a swindler and that the probability of such an extraordinary gift is less than 0.5. How low must your prior belief x have been for this to happen?

Problem 2: Backwards Heat Equation

Consider the heat equation over a unit interval,

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2},$$

with boundary conditions

$$u(0,t) = u(1,t) = 0, \quad t \geq 0,$$

and initial condition

$$u(x,0) = u_0(x) = \begin{cases} 1, & \text{if } 1/3 < x < 2/3, \\ 0 & \text{otherwise.} \end{cases}$$

Numerical solution by method of lines

- Discretize the interval $[0, 1]$ in n subintervals
- Discretize the second order derivative w.r.t. x using finite differences:

$$\frac{\partial^2 u}{\partial x^2}(x_j) \approx \frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}).$$

We get a vector U of $n-1$ unknowns, $u_j(t) = u(jh, t)$,

$$U = \begin{bmatrix} u_1(t) \\ \vdots \\ u_{n-1}(t) \end{bmatrix},$$

satisfying

$$\frac{dU}{dt} = \mathsf{L}U.$$

Numerical solution by method of lines

Matlab code for discretization:

```
n = 150; % Number of intervals
h = 1/n;
D = 0.1; % Heat diffusion coefficient

% Construct the FD matrix
aux    = zeros(n-1,1);
aux(1) = -2;
aux(2) = 1;
L = toeplitz(aux);

% Define the rhs for ODE call
rhs = @(t,u) (D/h^2)*(L*u + f);
```

Numerical solution by method of lines

```
% Initial value

x   = (1/n)*(1:n-1)';
u0 = zeros(n-1,1);
I = find(x>1/3 & x <2/3);
u0(I) = 3*ones(length(I),1);

figure(1)
plot([0:1/n:1],[a;u0;a],'r-','LineWidth',3)
axis([0,1,0,3.2])
set(gca,'FontSize',20)
```

Numerical solution by method of lines

```
% Propagating

tspan = [0:0.02:1];

[time,u] = ode15s(rhs,tspan,u0);

figure(2)
for j = 1:length(time)
    plot([0:1/n:1],[0,u(j,:),0],'k-','LineWidth',3)
    text(0.2,2,['t =' num2str(time(j))],'FontSize',20)
    axis([0,1,0,3.2])
    set(gca,'FontSize',20)
    pause(0.1)
end
```

Backwards Heat Equation

1. Starting from the final temperature, test how far you can propagate backwards the solution, simply by reversing the time.
2. Try the same adding a low amplitude artificial noise to the final state.
3. Comment your findings.