# Examples of IN5270/IN9270 exam questions from earlier semesters

IN5270/IN9270 Lecturer

December, 2020

## Note:

This note only has the purpose of showing some examples of IN5270/IN9270 exam questions from earlier semesters. **It is not to be considered as a scoping of the course curriculum, which can be found on the semester webpage.** The actual amount of work needed during this semester's home exam will approximately equal 2∼3 such questions.

## Example question 1

You are asked to approximate the function $f(x) = 1 + 2x - x^2$ in the domain $x \in [0, 1]$ by the projection method and using finite element basis functions.

1. Show the details and result of the calculation when a single P2 element is used to cover the domain.
2. Show the details and result of the calculation when two equal-sized P1 elements are used to cover the domain.
3. Extend the projection method to using $N$ equal-sized P1 elements. Show the details of how to set up the corresponding linear system. (There's no need to solve the linear system.)
4. If we want in addition that the approximation result, when using $N$ equal-sized P1 elements, should attain the same value of $f(x)$ at $x = 0$ and $x = 1$, what are the changes needed in the calculation above?

**High-level solution suggestions**

*1.1:* Recall that the general formula for setting up the needed linear system $Ac = b$ is that each entry in the matrix $A$ is computed by $A_{i,j} = \int_\Omega \varphi_i(x)\varphi_j(x)dx$ and each entry in the right-side vector $b$ is computed by $b_i = \int_\Omega f(x)\varphi_i(x)dx$.

The actual expression for the three P2 basis functions can either be derived directly (by using the physical coordinates of the three nodes $x_0 = 0$, $x_1 = \frac{1}{2}$, $x_2 = 1$) or mapped from the standardised basis functions that are pre-defined using the reference local coordinate $X \in [-1, 1]$. Here, the mapping is given by $x = x_m + \frac{h}{2}X$ or, the other way around, $X = \frac{2}{h}(x - x_m)$. Remark that $x_m$ is the physical coordinate of the mid-point of an element, which is $x_m = \frac{1}{2}$ for the case of a single P2 element covering the physical domain

$x \in [0,1]$. The value $h$ is the actual length of the element, which in this case is $h = 1$. *Note: Details of the* $3 \times 3$ *matrix* $A$ *and the right-hand side vector* $b$ *are* **not** *given here, neither is the resulting* $c$ *vector after solving the linear system* $Ac = b$.

*1.2:* Here, it is recommended to compute the element matrices and vectors for the two P1 elements separately. (Actually, the element matrix remains the same.) Also, mapping to the standardised reference element is recommended (although strictly speaking not absolutely necessary for this two-element case). The resulting $3 \times 3$ linear system (by assembling the element matrices and vectors) is different from that of Task 1.1. *Note: The actual details are* **not** *given here.*

*1.3:* Here, it is beneficial to map the computation of the element matrices and vectors to the standardised reference element. Note that now each element is of length $h = 1/N$. For element $e$, $0 \le e < N$, which spans from $x = eh$ to $x = (e+1)h$, the midpoint is thus $x_m = (e + \frac{1}{2})h$. *The details are* **not** *given about the resulting* $(N+1) \times (N+1)$ *linear system, which arises from assembling all the element matrices and vectors.*

*1.4:* The only change needed for the $(N+1) \times (N+1)$ linear system from above is to replace the first and last equation with $c_0 = 1 + 2 \cdot 0 - 0^2 = 1$ and $c_N = 1 + 2 \cdot 1 - 1^2 = 2$, respectively.

# Example question 2

You are asked to solve the 1D Poisson equation

$$-u_{xx} = 1, \quad 0 < x < 1$$

by a finite difference method. On the left boundary point of $x = 0$, the following mixed boundary condition

$$u_x + Cu = 0$$

is valid, where $C$ is a scalar constant. On the right boundary point of $x = 1$, the Dirichlet boundary condition $u = D$ is valid. We assume that a uniform mesh of $N + 1$ points is used by the finite difference method.

1. Discretize the Poisson equation on all the $N - 1$ interior points.
2. Discretize the left boundary condition using appropriate finite differencing.
3. Show the details of setting up a linear system $\mathbf{Au} = \mathbf{b}$ which can be used to find the approximations of $u(x)$ on the mesh points. (There's no need to solve the linear system.)
4. How would you validate that the obtained numerical solutions converge towards the exact solution, when the number of mesh points is increased? What is the expected convergence speed?

**High-level solution suggestions**

*2.1:* On each interior point, the finite difference discretisation result is simply $-\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 1$.

*2.2:* We need to combine the finite difference discretisation of the boundary condition at the left boundary point, $\frac{u_1-u_{-1}}{2h}+Cu_0=0$, with $-\frac{u_{-1}-2u_0+u_1}{h^2}=1$, for the purpose of eliminating the temporarily introduced "ghost" value $u_{-1}$. *Note: The actual details are* **not** *fully given here.*

*2.3:* It is simply achieved by combining the result from Task 2.2 (for $i=0$), with the result from Task 2.1 (for $i=1,2,\ldots,N-1$), and $u_N=D$ to form a $(N+1)\times(N+1)$ linear system.

*2.4:* The basic idea is to check the speed of convergence for the $L_2$ norm of the error, that is, the difference between the exact solution (which can be derived, details not shown here) and the numerical solution. This will require a sequence of experiments with finer and finer mesh resolution $h$, as well as using a numerical integration rule (such as the trapezoidal rule) to approximately calculate the $L_2$ norm of the error. (The latter is because we only know the numerical solution at the discrete mesh points in a finite difference method.) The expected convergence rate is second-order with respect to $h$.

# Example question 3

The following 1D stationary convection diffusion equation

$$u_x=\varepsilon u_{xx}$$

is to be solved by finite differencing in the domain $0<x<1$, where $\varepsilon>0$ is a given constant and the boundary conditions are $u(0)=0$ and $u(1)=1$.

1. Show that the above equation has $u(x)=\dfrac{1-e^{x/\varepsilon}}{1-e^{1/\varepsilon}}$ as its exact solution.
2. Assume a uniform mesh that consists of $N+1$ points: $x_0$, $x_1$, …, $x_N$, where $x_i=i\cdot h$. Use centered finite differences to discretize the equation, and show the details of how to set up the resulting linear system. (There's no need to solve the linear system.)
3. Prove that the analytical solution of the centered finite difference scheme is of form $u_i=C_1\beta_1^i+C_2\beta_2^i$, where $\beta_1=1$ and $\beta_2=\frac{1+\frac{h}{2\varepsilon}}{1-\frac{h}{2\varepsilon}}$. The values of $C_1$ and $C_2$ should be determined using the boundary conditions. What is the stability condition for the numerical solution?
4. Derive another numerical scheme where the convection term $u_x$ is discretized by so-called upwind finite difference. That is, $u_x$ is approximated at $x=x_i$ by

   $$\frac{u_i-u_{i-1}}{h}$$

   Any advantage and/or disadvantage of this numerical scheme in comparison with the above scheme?

**High-level solution suggestions**

*3.1:* This is straightforwardly done by inserting $u(x)=\dfrac{1-e^{x/\varepsilon}}{1-e^{1/\varepsilon}}$ into the equation and verify that it perfectly fits. Remember to also verify that the boundary conditions are satisfied. Note that the exact solution is monotonly increasing.

*3.2:* Using centered finite difference will give $\frac{u_{i+1}-u_{i-1}}{2h} = \varepsilon\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}$ on each interior mesh point, which can be rewritten as $(-h-2\varepsilon)u_{i-1}+4\varepsilon u_i+(h-2\varepsilon)u_{i+1}=0$. Using this for $i=1,2,\ldots,N-1$, combined with $u_0=0$ and $u_N=1$, will give a $(N+1)\times(N+1)$ linear system.

*3.3:* It is noted that $(-h-2\varepsilon)u_{i-1}+4\varepsilon u_i+(h-2\varepsilon)u_{i+1}=0$ is also a homogeneous difference equation with constant coefficients. The mathematical theory says that the general solution of such difference equations is $C\beta^i$. To determine the possible values of $\beta$, we replace $u_{i-1}$ with $C\beta^{i-1}$, $u_i$ with $C\beta^i$ and $u_{i+1}$ with $C\beta^{i+1}$. Then we will get a quadratic polynomial equation $(h-2\varepsilon)\beta^2+4\varepsilon\beta-(h+2\varepsilon)=0$. Solving this quadratic equation will give two roots, namely, $\beta_1=1$ and $\beta_2=\frac{1+\frac{h}{2\varepsilon}}{1-\frac{h}{2\varepsilon}}$. The values of $C_1$ and $C_2$ can be determined using the boundary conditions. There is a limit of how large $h$ can be, because if $h$ exceeds $2\varepsilon$, the numerical solution will exhibit an unwanted oscillatory feature (due to $\beta_2<0$), thus considered numerically unstable. *Note: The details are* **not** *fully given here.*

*3.4: Note: The details are* **not** *given here, but the take-home message is that the change in the discretisation of $u_x$ will lead to a different quadratic polynomial equation that have positive values for both $\beta_1$ and $\beta$. The numerical scheme is thus unconditionally stable, at the cost of lower accuracy.*

# Example question 4

We consider the following nonlinear diffusion equation (which is applicable for multiple space dimensions):

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \nabla\cdot(\alpha(\mathbf{x},t)\nabla u)+f(u) \quad \mathbf{x}\in\Omega,\ t\in(0,T],\\
u(\mathbf{x},0) &= I(\mathbf{x}) \qquad \mathbf{x}\in\Omega,\\
\frac{\partial u}{\partial n} &= g \qquad \mathbf{x}\in\partial\Omega,\ t\in(0,T].
\end{aligned}$$

Note: $\frac{\partial u}{\partial n}$ denotes the outward normal derivative on the boundary $\partial\Omega$, and $g$ is a constant.

1. Use the Crank-Nicolson scheme in time and show the resulting time discrete problem for each time step.

2. Formulate Picard iterations to linearize the time discrete problem.

3. Use the Galerkin method to discretize the stationary linear PDE per Picard iteration. Show the details of how to derive the corresponding variational form.

4. Restrict now the spatial domain to the 1D case of $x\in(0,1)$, let $\alpha$ be a constant and choose $f(u)=u^2$. (The boundary conditions are now $u_x=-g$ at $x=0$ and $u_x=g$ at $x=1$.) Suppose the 1D spatial domain consists of $N$ equal-sized P1 elements. Carry out the calculation in detail for computing the element matrix and vector for the leftmost P1 element.

5. What is the resulting global linear system $\mathbf{Ax}=\mathbf{b}$?

**High-level solution suggestions**

*4.1:* $\frac{u^\ell - u^{\ell-1}}{\Delta t} = \frac{1}{2}\left(\nabla\cdot(\alpha(\mathbf{x},t_\ell)\nabla u^\ell) + f(u^\ell) + \nabla\cdot(\alpha(\mathbf{x},t_{\ell-1})\nabla u^{\ell-1}) + f(u^{\ell-1})\right)$

*4.2:* $\frac{u^{\ell,k} - u^{\ell-1}}{\Delta t} = \frac{1}{2}\left(\nabla\cdot(\alpha(\mathbf{x},t_\ell)\nabla u^{\ell,k}) + f(u^{\ell,k-1}) + \nabla\cdot(\alpha(\mathbf{x},t_{\ell-1})\nabla u^{\ell-1}) + f(u^{\ell-1})\right)$ for $k = 1,2,\ldots$ until convergence on time level $t_\ell$. Note: This is a stationary linear PDE per Picard iteration. For the initial guess per time step, we will use $u^{\ell,0} = u^{\ell-1}$.

*4.3:* The main idea is to multiply the stationary linear PDE per Picard iteration (from above) with $v \in V$ and integrate over the entire domain. Using integration by parts (in multiple dimensions, also called Green's first identify) will result in the weak variational formulation. *Note: Details are* **not** *given here.*

*4.4: Details are* **not** *given here.*

*4.5: Details are* **not** *given here.*

# Example question 5

You are asked to solve the 2D Poisson equation:

$$-\nabla\cdot\nabla u = 2$$

in the unit square $(x,y) \in [0,1]^2$. A homogeneous Neumann condition $\frac{\partial u}{\partial n} = 0$ applies on the entire boundary $\partial\Omega$ (which has four sides: $y = 0$, $x = 1$, $y = 1$, $x = 0$).

We will use a 2D uniform mesh consisting of $M \times N$ elements ($N$ elements in the $x$ direction, $M$ elements in the $y$ direction), which all adopt bilinear basis functions.

1. Use the Galerkin method, derive the variational form of the above PDE in detail.
2. What are the degrees of freedom and how many are they in total? How would you number the degrees of freedom, with respect to the rows in a global linear system to be set up?
3. Describe in detail how the bilinear basis functions $\tilde{\varphi}_0(X,Y)$, $\tilde{\varphi}_1(X,Y)$, $\tilde{\varphi}_2(X,Y)$ and $\tilde{\varphi}_3(X,Y)$ are defined in a reference cell $(X,Y) \in [-1,1]^2$.
   (Hint: Each basis function is of the form $(aX+b)\cdot(cY+d)$ with suitable choices of the $a,b,c,d$ scalar values.)
4. For element number $e$, how can the physical coordinates $(x,y)$ be mapped from the local coordinates $(X,Y)$ of the reference cell?
5. Compute the element matrix and vector for element number $e$, with help of the reference cell.

**High-level solution suggestions**

*5.1:* The weak variational formulation is $\int_\Omega \nabla u\cdot\nabla v\, dx = \int_\Omega 2v\, dx$ for all $v \in V$. *Note: Green's first identify has been used, which also indirectly enforces the homogeneous Neumann boundary condition. The details are* **not** *fully given here.*

*5.2:* The number of degrees of freedom is $(M+1)(N+1)$, the same as the number of mesh nodes. Each degree of freedom $c_j$ is the weight of the corresponding basis function $\varphi_j(x,y)$, altogether giving the numerical solution $u(x,y) = \sum c_j \varphi_j(x,y)$. The easiest 1D numbering scheme of the degrees of freedom is to go through all the mesh nodes row-by-row, from bottom to top.

*5.3: Note: Details are* **not** *shown here, but the idea is to find suitable values of $a,b,c,d$ such that each basis function attains value 1 on its corresponding node and zero on the other three nodes.*

*5.4:* For a uniform 2D mesh of rectangular elements, the simplest mapping from the reference local coordinates $(X,Y) \in [-1,1]^2$ to the physical coordinates is given by $x = x_m + \frac{\Delta x}{2} X$ and $y = y_m + \frac{\Delta y}{2} Y$, where $\Delta x = 1/N$ and $\Delta y = 1/M$.

*5.5: Note: Details are* **not** *shown here, but it is worth mentioning that $\nabla\varphi_j \cdot \nabla\varphi_i$ in 2D is $\frac{\partial\varphi_j}{\partial x}\frac{\partial\varphi_i}{\partial x} + \frac{\partial\varphi_j}{\partial y}\frac{\partial\varphi_i}{\partial y}$. For carrying out the element-wise computation using the standardised reference element we need to use the mapping from Task 5.4.*