Project 3: A semilinear elliptic equation

Spring 2023

Question 1

We want to find the weak formulation to the problem of finding u_{n+1} satisfying

$$\begin{cases} -\Delta u_{n+1} + \alpha u_n^2 u_{n+1} = f & \text{in } \Omega, \\ u_{n+1} = 0 & \text{on } \partial\Omega, \end{cases}$$

where u_n is the current iterate, with $u_n = 0$ on $\partial\Omega$.

We multiply by a test function $v \in H_0^1(\Omega)$ and integrate to obtain:

$$\int_\Omega fv = \int_\Omega \left(-\Delta u_{n+1} v + \alpha u_n^2 u_{n+1} v\right).$$

We use integration by parts to rewrite the right hand side:

$$\begin{aligned} \int_\Omega (-\Delta u_{n+1} v) &= -\int_\Omega \operatorname{div}(\nabla u_{n+1} v) + \int_\Omega \nabla u_{n+1} \nabla v && \text{integration by part} \\ &= -\oint_{\partial\Omega} v(\nabla u_{n+1} \cdot \vec{n}) + \int_\Omega \nabla u_{n+1} \nabla v && \text{by the divergence theorem} \\ &= \int_\Omega \nabla u_{n+1} \nabla v && \text{by } v \in H_0^1(\Omega), \end{aligned}$$

where we denote $\vec{n}$ the outer normal vector on the boundary $\partial\Omega$.

The weak formulation is:

$$\int_\Omega \nabla u_{n+1} \nabla v + \int_\Omega \alpha u_n^2 u_{n+1} v = \int_\Omega fv \quad \forall v \in H_0^1(\Omega). \tag{E1}$$

We have a stiffness integral, and a mass integral with the reaction term αu_n^2, as well as a Poisson right hand side.

Question 2

The code for the implementation of the fixed-point scheme is in the files `integrate.py` and `Project_SEZAM.py`. ADD WHAT EACH FILE CONTAIN For $\alpha = 0.1$, the value $||u_n - u_{n+1}||_\infty$ drops below 10^{-6} after 11 iterations. We obtain the figure 1 below, by running `fixed_point_method(0.1, 10e-6, 500, 0.05)`.

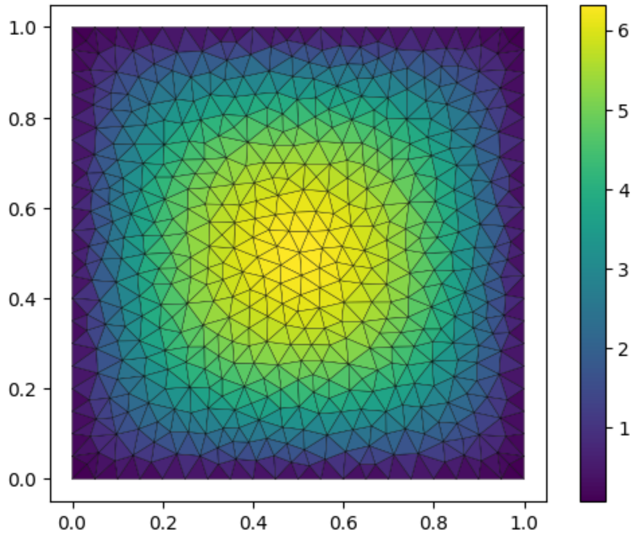

Figure 1: Solution of the fixed-point scheme for $\alpha = 0.1$

We also show the convergence rate in the figure 2.

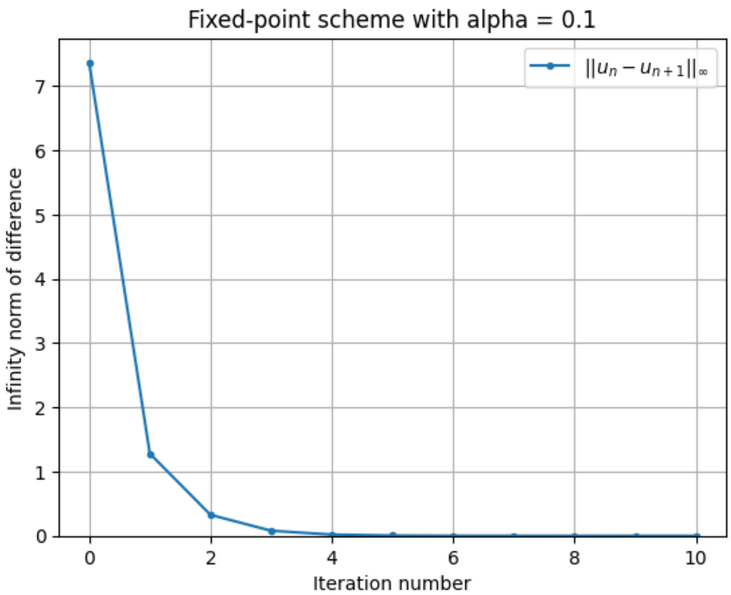

Figure 2: Convergence of the fixed-point scheme for $\alpha = 0.1$

For $\alpha = 2$, it drops below 10^{-6} after 360 iterations, which is much longer. We obtain the figures 3 and 4 by running `fixed_point_method(2, 10e-6, 500, 0.05)`.

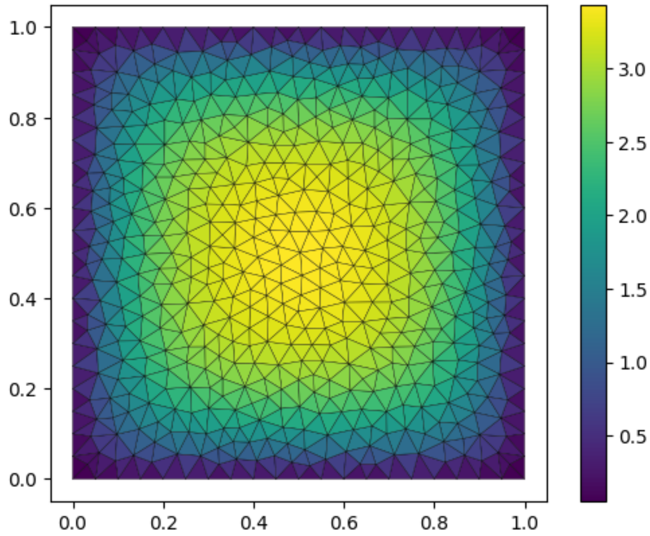

Figure 3: Solution of the fixed-point scheme for $\alpha = 2$

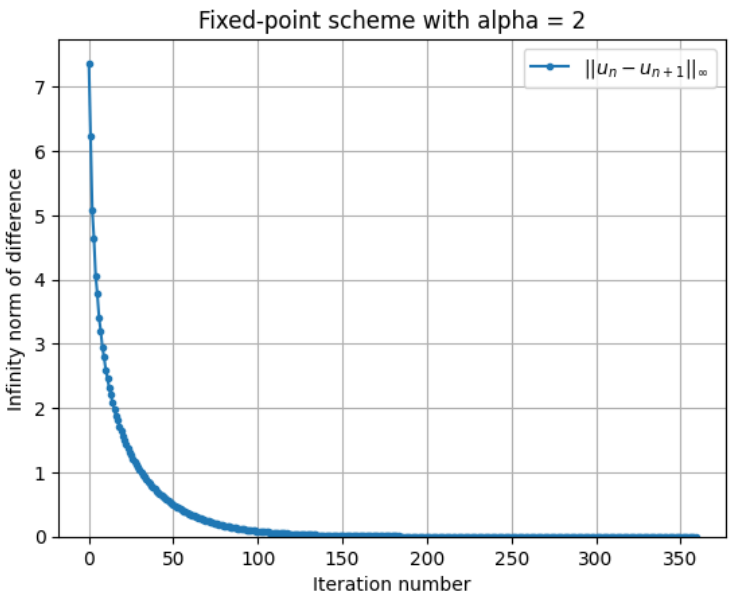

Figure 4: Convergence of the fixed-point scheme for $\alpha = 2$

Question 3

The code for the implementation of the Newton scheme is also in the files `integrate.py` and `Project_SEZAM.py`. ADD PRECISE FUNCTIONS At each iteration, we need to search ∂u_n that satisfies

$$\int_\Omega \nabla\phi \cdot \nabla\partial u_n + 3\alpha u_n^2 \phi \partial u_n \mathrm{d}\Omega = -\int_\Omega \nabla\phi \cdot \nabla u_n + \phi(\alpha u_n^3 - f)\mathrm{d}\Omega \quad \forall\phi \in H_0^1(\Omega),$$

$$\partial u_n = 0 \quad \text{on } \partial\Omega.$$

Then, we seek the solution u as the limit of the recursive sequence $u_{n+1} = u_n + \partial u_n$.

For $\alpha = 0.1$, we obtain $||u_n - u_{n+1}||_\infty < 10^{-6}$ after 5 iterations, the solution is shown in figure 5 below.

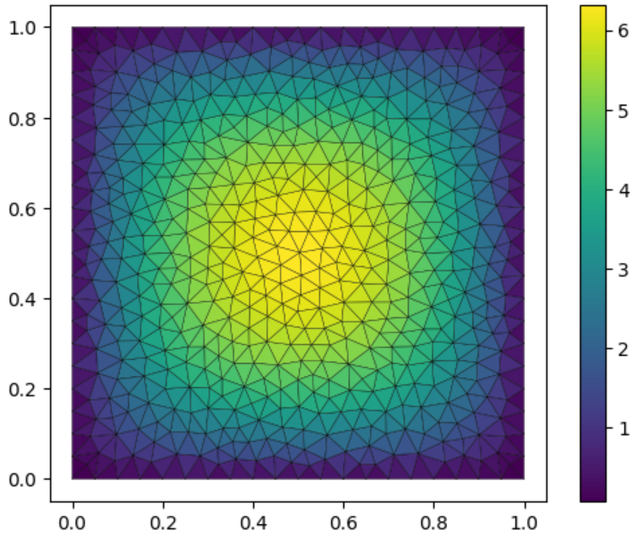

Figure 5: Solution with the Newton scheme for $\alpha = 0.1$

We see that we find the same solution as with the fixed-point method (see figure 1), and we just need 5 less iterations with the Newton scheme (which actually divide the number of iterations by 2 in this case). The convergence evolution of the Newton scheme for $\alpha = 0.1$ is in the figure 6:

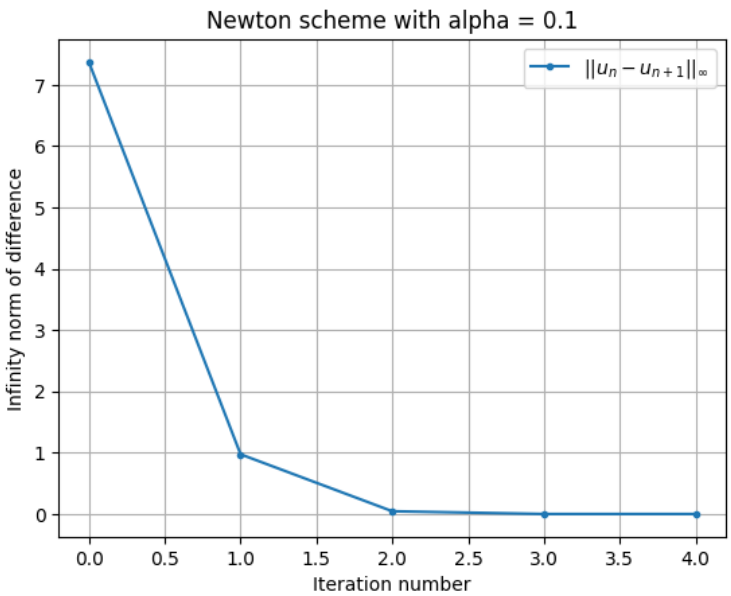

Figure 6: Convergence of the Newton scheme for $\alpha = 0.1$

We obtain those figures by running `newton_method(0.1, 10e-6, 500, 0.05)`. For $\alpha = 2$, we converge in 7 iterations, which is considerably faster than with the fixed-point scheme (we needed 360 iterations to converge with this scheme).

We obtain the following results by running `newton_method(2, 10e-6, 500, 0.05)`:

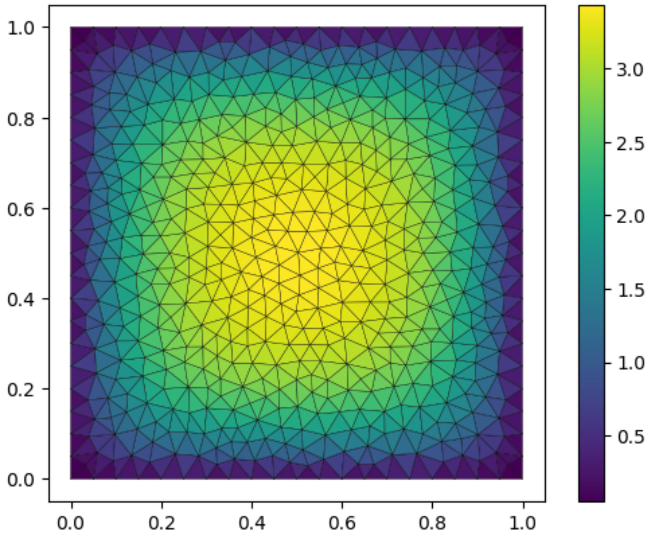

Figure 7: Solution of the Newton scheme for $\alpha = 2$

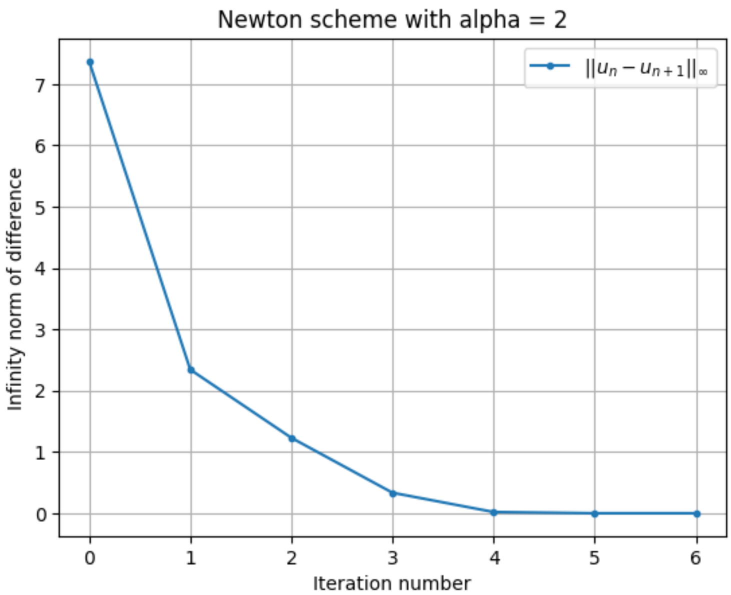

Figure 8: Convergence of the Newton scheme for $\alpha = 2$

We also see that the solutions are the same for both scheme (figure 7 above is the same as figure 3). For $\alpha = 5$, we also converge in 7 iterations, and obtain the results shown in figures 9 and 10 below (by running `newton_method(5, 10e-6, 500, 0.05)`).

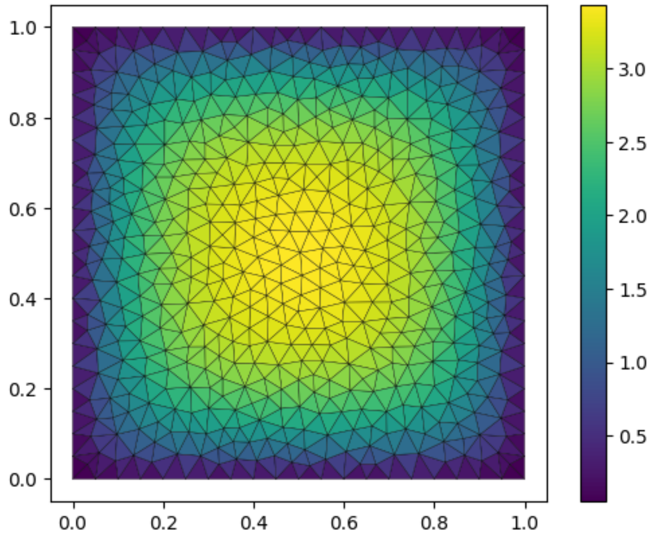

Figure 9: Solution of the Newton scheme for $\alpha = 2$

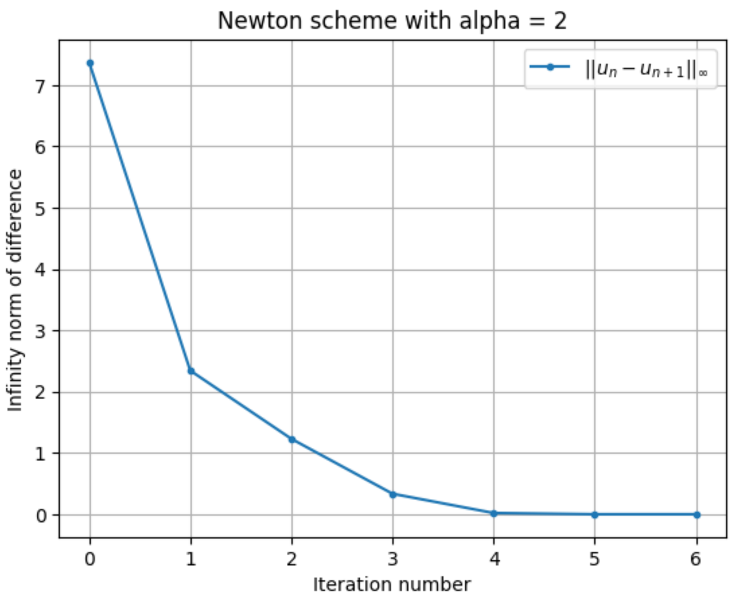

Figure 10: Convergence of the Newton scheme for $\alpha = 2$

The Newton scheme fare much better than the fixed-point scheme. The number of iterations to obtain $||u_n - u_{n+1}||_\infty < 10^{-6}$ is considerably smaller. (We ran the scheme for $\alpha = 100$, and found that it converges in 10 iterations.)