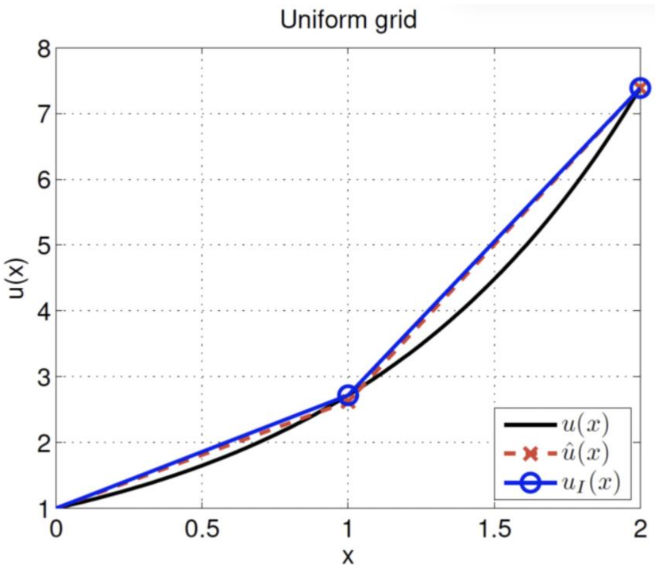
02623 Finite Element Method for Differential Equations

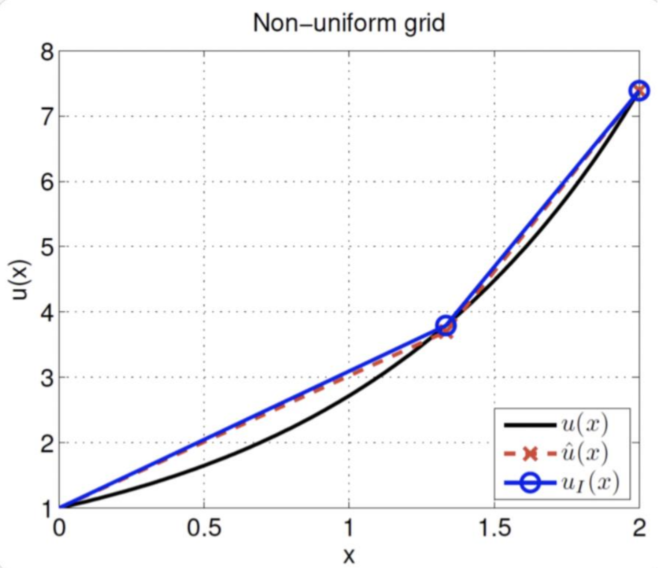
Week 1 Exercise Solutions

Exercise 1.1 b) uniform mesh, uhat2 = 2.6216

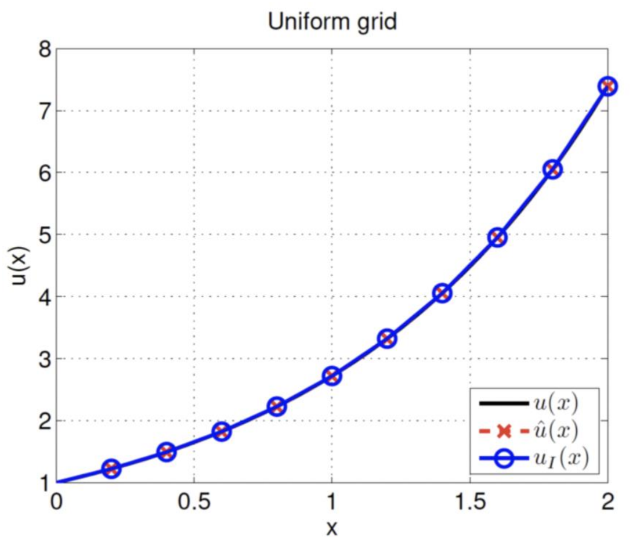
Exercise 1.1 b) non-uniform mesh, uhat2 = 3.6996

Exercise 1.1 c) + d) solutions can be visualized as





Exercise 1.2 b) solutions can be visualized as



Exercise 1.5 b) the element matrix you derive should look like this

$$\mathcal{K}^{(i)} = \begin{bmatrix} \frac{\Psi}{2} + \frac{\epsilon}{h_i} & \frac{\Psi}{2} - \frac{\epsilon}{h_i} \\ -\frac{\Psi}{2} - \frac{\epsilon}{h_i} & -\frac{\Psi}{2} + \frac{\epsilon}{h_i} \end{bmatrix}$$

HINTS TO EXERCISE 1.5

a) From (1.34) derive the result given in the form (1.35).

b) From the weak formulation in a) derive the expressions for the matrix A in the form a_{i,j} = ?? in terms of the finite element basis functions, cf. procedure in sections 1.4-1.5 but done instead for the system to be considered in this exercise. Remark, the contributions to the a_{i,j} an be expressed in terms of local element matrices k_{i,j}. It is instructive to derive the general formula for k_{i,j} and then derive the analytical results for i,j=1,2 that will be needed for implementation.

c) To show that v^TAv>0, we can use the first hint to connect the discrete system with that of evaluating a(v,v). Hence, compute what is the value of a(v,v) to show that a(v,v)>=0 as the first step (i). Then, show that v^TAv=0 implies v=0 using the expression and information about v given.

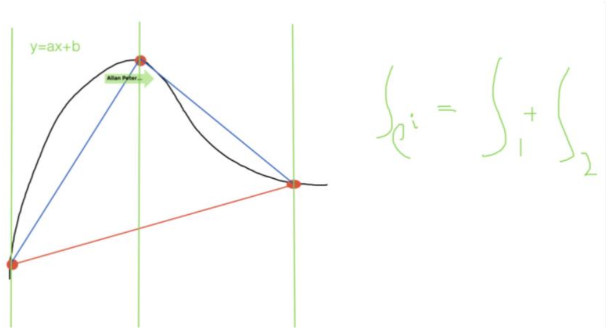
d) This is a plotting exercise and the analytical expressions may be difficult to evaluate by typing in the formulas directly as stated. Hence rewrite them so they work, cf. other comment given in slack (by Fatma) on how to do this.

e) Implement the results derived in b) in a new script tailored to solve the problem at hand. Verify convergence for the parameters epsilon=1,0.01,0.0001. Do you observe anything worth discussing?

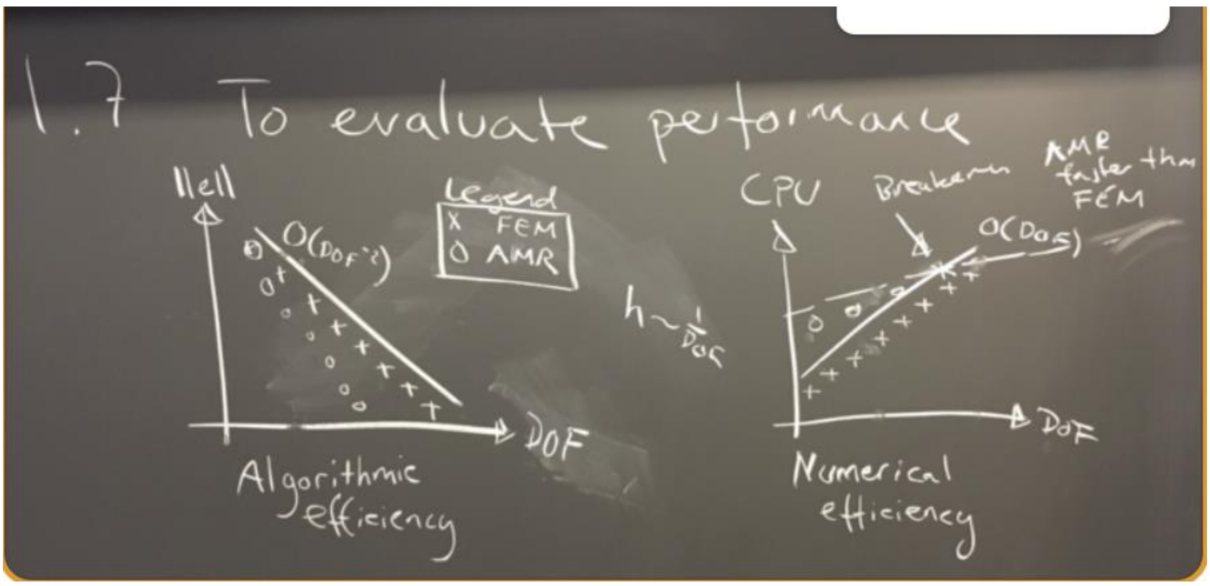
HINTS TO EXERCISE 1.6

Exercise 1.6. For the norm calculation the purpose is to derive a formula that can be implemented. This formula can be expressed in terms of (x_i, u_i) values and should in the end be easy to implement. You are not supposed to use any matlab functions to do this job. Derive the formula yourself.

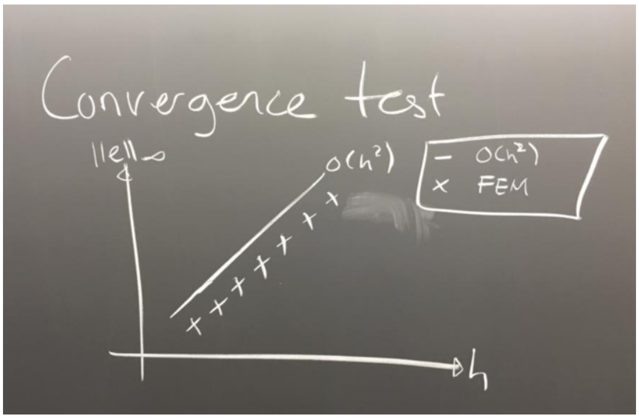
In the illustration given here, you can see visually what goes on when you do interpolation of u(x) that is the black solid line. The red line interpolated the solution at a coarse mesh (h is mesh size_) across element e_i and the blue line does the interpolation on a fine mesh (h/2 is the mesh size) but with an additional point. The fine mesh is based on subdivision of the coarse element, which implies that there are now two regions that should be taken into account. To derive a formula for the error estimation, one can simply derive what is the expressions for the linear equations in the form y=ax+b for each line across the segments. Then substract these analytically and integrate them exactly also analytically. This should lead to a formula that can be (easily) implemented in your code.



In exercise 1.7 …. how to evaluate the performance of the AMR?... compare to a standard refinement strategy using FEM where uniformly distributed elements are used.



Please test your FEM solver in exercise 1.7 so that it shows something like the result in the picture when a uniform mesh is used. If this test works with the result that FEM convergence as O(h^2) then we can thrust your code...


Convergence test
||e||∞
O(h²)
— O(h²)
x FEM
h