1a)					
Write the system					
e	ori etteri				
- +c1) = -	ne ucarj -	12-1/2-3 Wi-1-1 - 0	h2 wijsts - wijs-1/2	m:13-4 + a:+4/5/3400	102 Hay
	TBA CALAR				
A = 1	Chich B2 Cznah	B	- M(A13) GA+4(-13		
No			Section - 12 (21)	:	
	Cont Bu	-O+1/2	σ _{0-4,j} - κ(ν,j) σ _{0-4,j} - κ(ν,j)	33	
			21/		
Cu =	[41,14				
CK -					
	Gu,N				
b)					
Coro problem) In t	the template file finite_differ	sense ann implement the funct	ion		
oid createPorous	sMediaMatrix2D(SparseMat	rix& A, FunctionPointer	sigma, int N, double dx),		
set matrix a	from one vector	containing the making	es B and C		
double k -	to to mus d				
Special condition	ans to push_back i	n the right places			
around we	Langeire				
c)					
Core problem) In t	he template file finite_differ	ence can implement the funct	ion		
	Vector& rhs, FunctionPoi				
			,		
F = dx -{	(mm) -> Serie 3				
d)				<u>:-4</u>	مرح ب هذا في المعادة معن المعادة المعا
C ore problem) In t	the template file finite_differ	cence.cpp, implement the func	tion		N _S
ector porousMedi	iaSolve(FunctionPointer f	, FunctionPointer f, int	N),		
Noinel L	.U solve like in the	Serie!			
Sporse matrix	requires Eigen/Spars4	<i>علنا</i>			
f)					
Core problem) In t	the template file finite_differ	rence.cpp, implement the fund	tion —		
oid convergeSt	tudy(FunctionPointer F	, FunctionPointer sig	ma),		
l. = 0h	s (u - exacts)				
	,, (m = exect =)				

```
2a)
  Write the variational formulation for (4)-(5).
   Write an expression for the entries of \mathbf{A} and \mathbf{F} in (7).
     AN = F
     Aij = [(\alpha(\frac{1}{2}) \rightarrow \varphi_{\beta}^{\beta}(\frac{1}{2}) \rightarrow \varphi_{\beta}^{\beta}(\frac{1}) \rightarrow \varphi_{\beta}^{\beta}(\frac{1}{2}) \rightarrow 
    Fr = 1 + (x) 4: (x) 4x
  (Core problem) Complete the template file shape.hpp implementing the function
 inline double lambda(int i, double x, double y)
   Implemented the 3 conduitions given
  2d)
  (Core problem) Complete the template file grad_shape.hpp implementing the function
 inline Eigen::Vector2d gradientLambda(const int i, double x, double y)
       Suitch to decide the shape functions shouling from 1
2e)
(Core problem) Complete the template file stiffness_matrix.hpp implementing the routine
template<class MatrixType, class Point>
void computeStiffnessMatrix(MatrixType& stiffnessMatrix.
      Given: Ju, det Ju, (Ju"), fuction
                          coordit , vol + , elem Mop
     Fix every cell of the matrix by the
      integration:
      AC_{i} = \int \left( \alpha(\underline{x}) \triangleright Q_{i}^{N}(\underline{x}) \triangleright Q_{i}^{N}(\underline{x}) + cQ_{i}^{N}(\underline{x}) Q_{i}^{N}(\underline{x}) \right)
2f)
(Core problem) Complete the template file load_vector.hpp implementing the routine
template<class Vector, class Point>
 void computeLoadVector(Vector& loadVector, const Point& a, const Point& b,
          Giren: In, det Su = du, fuerion to integrate
          Some as notion but here we integrate
            Fi = [ f(x) 4, (x) dx to get the load vector
 (Core problem) Complete the template file stiffness_matrix_assembly.hpp implementing the
 template<class Matrix>void assembleStiffnessMatrix(Matrix& A, const Eigen::MatrixXd& vertices,
      - Get indices of vertices
       - Build the 3x3 matrix
           . b Arch herices and motive as a higher
             Buid He natix
2h)
(Core problem) Complete the template file load_vector_assembly.hpp implementing the routine
void assembleLoadVector(Eigen::VectorXd& F, const Eigen::MatrixXd& vertices,
   - get hiangle vehices
      - b get load vector for Hat highe
      - push enorths in the biffer board verter
```

(Core problem) Complete the template file fem_solve.hpp with the implementation of the function

— int solveFiniteElement(Vector& u, const Eigen::MatrixXd& vertices,

+ build A and +

_s Set Bouderies

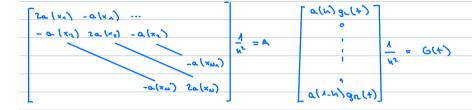
_s Solve "inherios" system, where there are no bandaries

- D combine bondary with interal solution

3a)

Denote by h the mesh width, that is $h = \frac{1}{N+1}$. Write down the matrix **A** and the vector G(t)

explicitly.



3b)

Apply the forward Euler scheme to (11), denoting by $\boldsymbol{u}^k = \{u_i^k\}_{i=1}^N$ the approximate value of the vector \boldsymbol{u} at time k, for $k=0,\ldots,K$, and by $\Delta t = \frac{T}{K}$ the time step. How does the update formula at each time step look like?

3c)

(Core problem) In the template file ${\sf create_poisson_matrix.cpp},$ implement the function

SparseMatrix createPoissonMatrix(int N, const std::functional<double(double)>& a),

- prepare matices at the right side

- Assign value of a(x) on the "dioponals"

Le quel right/left

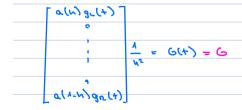
A blive d-

- Bivide by h

3d)

(Core problem) In the template file forward_euler.cpp, implement the function

std::pair<Eigen::MatrixXd, Eigen::VectorXd> forwardEuler(



(Core problem) In the template file crank_nicolson.cpp, implement the function
std::pair <eigen::matrixxd, eigen::vectorxd=""> crankNicolson(</eigen::matrixxd,>
To have a med and to have
$\left(\Gamma + \frac{\lambda}{2} A \right) U^{n+2} = A \cdot U^{n+2} = BU^{n} = \left(\Gamma - \frac{\lambda}{2} A \right) U^{n}$