Implicit method

Brak plików do pobrania.

In the implicitly called "backward Euler" method, used to solve non-stationary problems, the differential operator describing the "physics" of the modeled phenomenon is calculated in the "current" instant of time.

$\frac{u_{t+1}-u_t}{dt} - \mathcal{L}(u_{t+1}) = f_{t+1}$

So at each time step it is necessary to solve the equation for the next state u_{t+1}.

Let us follow the operation of the backward Euler method in the case of the isogeometric finite element method.

To develop a solver that uses the method implicitly in the isogeometric finite element method, we need to transform a strong formulation into a weak formulation.

So we multiply our equation by the test functions $(u_{t+1}, w) - (dt * \mathcal{L}(u_{t+1}), w) = (u_t + dt * f_{t+1}, w)$

We will use a linear combination of the B-spline function to approximate the state of our system at a given moment in time. For this purpose, we select the basis of two-dimensional B-spline functions, specifying the node vectors in the direction of the axis of the coordinate system, for example, the two-dimensional basis of the second-order B-spline function

$\{B^x_{i,2}(x)B^y_{j,2}(y)\}_{i=1,\dots,N_x;j=1,\dots,N_y} u_{i,j}$

They will be used to approximate the simulated scalar field of the current time instant

$u(x,y;t) \approx \sum_{i=1,\dots,N_x;j=1,\dots,N_y} u^t_{i,j} B^x_i(x) B^y_j(y)$

Similarly, we'll use the B-spline base functions for testing:

$w(x,y) \in \{B^x_k(x)B^y_l(y)\}_{k=1,\ldots,N_x;l=1,\ldots,N_y} w^{k,l}$

Assuming the differential operator $\mathcal{L}$ describing "physics" is linear, our equation now looks like this:

$\sum_{i=1,\ldots,N_x;j=1,\ldots,N_y} u^{t+1}_{i,j}(B^x_i(x)B^y_j(y) - dt * \mathcal{L}(B^x_i(x)B^y_j(y)), B^x_k(x)B^y_l(y)) = RHS$. We do not establish a specific

$\forall k = 1,\ldots,N_x; l = 1,\ldots,N_y$

problem here, the conclusions presented here refer to any physical problem that can be simulated with the method described. For example, for the problem of heat transport we have

$$\sum_{i=1,\ldots,N_x;j=1,\ldots,N_y} u^{t+1}_{i,j} \cdot$$

$$(B^x_i(x)B^y_j(y) - dt * \left(\frac{\partial^2(B^x_i(x)B^y_j(y))}{\partial x^2} + \frac{\partial^2(B^x_i(x)B^y_j(y))}{\partial y^2}\right), B^x_k(x)B^y_l(y)) = RHS$$

$\forall k = 1,\ldots,N_x; l = 1,\ldots,N_y$

Thanks to the weak wording, we can integrate by parts

$$\sum_{i=1,\ldots,N_x;j=1,\ldots,N_y} u^{t+1}_{i,j} \cdot$$

$$\left((B^x_i(x)B^y_j(y), B^x_k(x)B^y_l(y)) - dt * (\frac{\partial(B^x_i(x)B^y_j(y))}{\partial x}, \frac{\partial(B^x_k(x)B^y_l(y))}{\partial x}) - dt * (\frac{\partial(B^x_i(x)B^y_j(y))}{\partial y}, \frac{\partial(B^x_k(x)B^y_l(y))}{\partial y})\right) = RHS$$

$\forall k = 1,\ldots,N_x; l = 1,\ldots,N_y$

Due to the structure of the tensor product of the B-spline function and due to the fact that the derivative in the y direction of the B-spline in the x-axis direction gives 0 (because these functions are constant in the y-axis direction) and similarly for the derivative in the y-direction of the B-spline in the x-axis direction the x axis we have

$$\sum_{i=1,\ldots,N_x;j=1,\ldots,N_y} u^{t+1}_{i,j} \cdot$$

$$\left((B^x_i(x)B^y_j(y), B^x_k(x)B^y_l(y)) - dt * (\frac{\partial B^x_i(x)}{\partial x}B^y_j(y), \frac{\partial B^x_k(x)}{\partial x}B^y_l(y)) - dt * (B^x_i(x)\frac{\partial B^y_j(y)}{\partial y}, B^x_k(x)\frac{\partial B^y_l(y)}{\partial y})\right) = RHS$$

$\forall k = 1,\ldots,N_x; l = 1,\ldots,N_y$

Note that our system of equations to solve is

$(M_x \otimes M_y - dt * S_x \otimes M_y - dt * M_x \otimes S_y)\, u^{t+1} = F(t)$

where

$\{M_x \otimes M_y\}_{i,j,k,l} = \int B_i^x(x)B_k^x(x)B_j^y(y)B_l^y(y) = \int B_i^x(x)B_j^y(y)B_k^x(x)B_l^y(y) = \mathbf{M}_{i,j,k,l}$ is a mass matrix which is a Kronecker product of two one-dimensional mass matrices,

$\{S_x \otimes M_y\}_{i,j,k,l} = \int \frac{\partial B_i^x(x)}{\partial x}\frac{\partial B_k^x(x)}{\partial x}B_j^y(y)B_l^y(y) = \int \frac{\partial B_i^x(x)}{\partial x}B_j^y(y)\frac{\partial B_k^x(x)}{\partial x}B_l^y(y)$ is the Kronecker product of the one-dimensional stiffness matrix and the one-dimensional mass matrix, and

$\{M_x \otimes S_y\}_{i,j,k,l} = \int B_i^x(x)B_k^x(x)\frac{\partial B_j^y(y)}{\partial y}\frac{\partial B_l^y(y)}{\partial y} = \int B_i^x(x)\frac{\partial B_j^y(y)}{\partial y}B_k^x(x)\frac{\partial B_l^y(y)}{\partial y}$ is the Kronecker product of the one-dimensional mass matrix and the one-dimensional stiffness matrix.

Each of these matrices can be factored in linear time using the variable-directional solver algorithm. However, it is no longer possible to factorize their sum in linear time. This is possible only when we introduce a time step scheme that allows the matrix to be separated into sub-matrices in time sub-steps so that the factorization cost remains linear.

The Peaceman-Rachford diagram allows to split the time step into two sub-steps

$(M_x \otimes M_y - dt * S_x \otimes M_y)\, u^{t+1/2} = F(t+1/2) + (dt * M_x \otimes S_y)\, u^t$

$(M_x \otimes M_y - dt * M_x \otimes S_y)\, u^{t+1} = F(t+1/2) + (dt * S_x \otimes M_y)\, u^{t+1/2}$

where we can merge the left side matrices into a single matrix with a Kronecker product structure that can be factored in linear time using the variable direction solver algorithm

$(M_x - dt * S_x) \otimes M_y u^{t+1/2} = F(t+1/2) + (dt * M_x \otimes S_y)\, u^t$

$M_x \otimes (M_y - dt * S_y)\, u^{t+1} = F(t+1/2) + (dt * S_x \otimes M_y)\, u^{t+1/2}$

The right hand function here represents the changes caused by the force acting on the system during the time step. It is the sum of two elements:

1. The state of our system at the previous moment in time $(u_t, w) = \sum_{i=1,\ldots,N_x; j=1,\ldots,N_y} u_{i,j}^t (B_i^x(x)B_j^y(y), B_k^x(x)B_l^y(y))$ (also multiplied by the test function and area integrated). Of course, we also use a linear combination of B-spline base functions to represent the state of our system in the previous time step $u(x,y;t) = u_t = \sum_{i=1,\ldots,N_x; j=1,\ldots,N_y} u_{i,j}^t B_i^x(x)B_j^y(y)$
2. Changes caused by a force acting on the system during the time step $(f_{t+1}, w) = (f_{t+1}(x,y), B_k^x(x)B_l^y(y))$

Utworzona przez admin. Ostatnia aktualizacja: Wtorek 03 z Listopad, 2020 23:05:15 UTC przez paszynsk@agh.edu.pl. Autor: Maciej Paszynski

STATUS: W opracowaniu Zgłoś do recenzji Edytuj

Jak to działa?
O e-podręcznikach AGH
Regulamin
Polityka prywatności
Licencja CC BY-SA
Partnerzy
Kontakt
Prześlij opinię
About

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
Centrum e-Learningu

Wersja mobilna