A Hybrid Method for the Wave Equation

http://www.math.unibas.ch/~beilina

The mathematical model

The model problem is the wave equation

$$\frac{\partial^2 u}{\partial t^2} = \nabla \cdot (a^2 \nabla u) + f, \quad x \in \Omega \subset \mathbf{R}^3, \ t > 0, \quad (1)$$

$$u(x,0) = 0, \quad x \in \Omega, \tag{2}$$

$$\frac{\partial}{\partial t}u(x,0) = 0, \quad x \in \Omega, \tag{3}$$

$$u = 0, \quad x \in \Gamma, \ t > 0 \tag{4}$$

where u(x, t) is unknown, a is the wave velocity, x ranges over the points of the space domain, t over the time, f is the source function, and Γ denotes the boundary of the domain Ω . We can rewrite this second order equation as a system of first order equations in time using the substitution $u_1 = \frac{\partial}{\partial t}u$ and $u_2 = u$, thus obtaining

$$\frac{\partial}{\partial t}u_1 - \nabla \cdot (a^2 \nabla u_2) = f, \quad x \in \Omega, \tag{5}$$

$$\frac{\partial}{\partial t}u_2 = u_1, \quad x \in \Omega, \tag{6}$$

$$u_1(x,0) = u_2(x,0) = 0, x \in \Omega.$$
 (7)

In many wave equation applications, only a small part of the computational domain Ω is complex enough to motivate a more complex unstructured discretisation, whereas quite large regions of the computational domain are sufficiently discretised with simple, Cartesian grids. For our exposition, our model domain consists of two regions, Ω_{FEM} and Ω_{FDM} , not necessarily simply connected. In the relatively small Ω_{FEM} domain, we assume that an unstructured discretisation is appropriate. In the Ω_{FDM} domain, we assume that a structured, Cartesian, grid is suitable.



Figure 1: Domain decomposition. The hybrid mesh (c) is a combination of the structured mesh (a) and the unstructured mesh (b) with a thin overlapping of structured elements. The unstructured grid is constructed so that the grid contains edges approximating an ellipse.

Fig. 1 illustrates the principle in two dimensions.



Figure 2: In the overlapping domain⁶ the finite element grid is created by

The FEM grid is generated such that the thin overlapping domain consists of simplexes obtained by splitting the structured cells as described in Fig. 2. In the interior part of the FEM grid the discretisation is allowed to be truly unstructured.

In most of our test cases, we have used Dirichlet boundary conditions. We have also used an absorbing boundary condition. At a boundary Γ , we then use $\frac{\partial}{\partial t}u - \frac{\partial}{\partial n}u|_{\Gamma} = 0$, where $\partial/\partial n$ is the normal derivative.

The numerical method



Figure 3: Coupling of FEM and FDM. The nodes of the unstructured FEM grid of (a) is shown in (b), where rings and crosses are nodes which are shared between the FEM and FDM grids. The remaining nodes are marked with stars. The ring nodes are interior to the FDM grid, while the nodes crosses are interior to the FEM grid. At each time iteration, the FDM solution values at ring nodes are copied to the corresponding FEM solution values. At the same time at cross nodes the FEM solution values are copied

With a hybrid discretisation of the computational domain we are now in a position to formulate our hybrid algorithm. We observe that the interior nodes of the computational domain belong to either of the following sets:

- ω_o : Nodes interior to Ω_{FDM} and boundary nodes to Ω_{FEM} ,
- ω_{\times} : Nodes interior to Ω_{FEM} and boundary nodes to Ω_{FDM} ,
- ω_* : Nodes interior to Ω_{FEM} and not contained in Ω_{FDM} ,
- ω_D : Nodes interior to Ω_{FDM} and not contained in Ω_{FEM} .

Fig. 3 illustrates the situation for a two-dimensional domain where some nodes are confined to an ellipse, which requires an unstructured discretisation. The exterior and the interior of the ellipse may use a structured discretisation. For clarity, nodes belonging to Ω_D are not shown.

In our algorithm, we store nodes belonging to ω_o and Ω_{\times} twice, both as nodes belonging to Ω_{FEM} and Ω_{FDM} . For explicit time stepping

schemes, the main loop of the simulation can be sketched:

For every time step

- 1. Update the solution in the interior of Ω_{FDM} , i.e. at nodes ω_D and ω_o using FDM
- 2. Update the solution in the interior of Ω_{FEM} , i.e. at nodes ω_* and ω_{\times} using FEM
- 3. Copy values at nodes ω_{\times} from Ω_{FEM} to Ω_{FDM}
- 4. Copy values at nodes ω_o from Ω_{FDM} to Ω_{FEM}

The FEM and the FDM schemes that we use are well-known. For the sake of completeness, we present them below, and we also point out that we can regard the FDM sheme as a reformulation of the FEM scheme for a structured grid. Therefore, our hybrid approach can be analysed as a pure FEM scheme.

Finite element formulation

In the Ω_{FEM} domain where an unstructured grid is assumed, we use FEM. To formulate the finite element method for problem (1) we use the standard Galerkin finite element method with linear elements in space and a centralised finite difference approximation for the second order time derivative. We introduce finite element space V_h for u, consisting of standard piecewise linear continuous functions on a mesh and satisfying Dirichlet boundary conditions. Let V_h^0 denote the corresponding finite element spaces satisfying homogeneous Dirichlet boundary conditions. The finite element method now reads: find u_h in V_h such that

$$\left(\frac{u_h^{k+1} - 2u_h^k + u_h^{k-1}}{\tau^2}, v\right) + \left(a^2 \nabla u_h^k, \nabla v\right) = (f^k, v), \quad \forall v \in V_h^0.$$
(8)

This produces the system of linear equations

$$M\mathbf{u}^{k+1} = \tau^2 F^k + 2M\mathbf{u}^k - \tau^2 K\mathbf{u}^k - M\mathbf{u}^{k-1}, \qquad (9)$$

with proper initial and boundary conditions. Here, M is the mass matrix, K is the stiffness matrix depending on a possible varying wave velocity a, k = 1, 2, 3... denotes the time level, F^k is the load vector, \mathbf{u} is the unknown discrete field values of u, and τ is the time step.

The explicit formulas for the entries in system (9) at the element level can be given as:

$$M_{i,j}^e = \left(\frac{1}{c^2}\varphi_i, \varphi_j\right)_e, \tag{10}$$

$$K_{i,j}^e = (\nabla \varphi_i, \nabla \varphi_j)_e, \qquad (11)$$

$$F_j^e = (f, \varphi_j)_e, \tag{12}$$

where
$$(a,b)_e = \int_{\Omega_e} a \ b \ d\Omega_e$$
, (13)

where Ω_e is domain of the element e.

The matrix M^e is the contribution from element e to the global assembled matrix M, K^e is the similar contribution to global assembled matrix K,

 F^e is the contribution from element e to the assembled source vector F.

To obtain an explicit scheme we approximate M with the lumped mass matrix M^L , the diagonal approximation obtained by taking the row sum of M. By multiplying (9) with $(M^L)^{-1}$ we obtain an efficient explicit formulation:

$$\mathbf{u}^{k+1} = \tau^2 (M^L)^{-1} F^k + 2\mathbf{u}^k - \tau^2 (M^L)^{-1} K \mathbf{u}^k - \mathbf{u}^{k-1}, \quad (14)$$

where matrix M^L is the approximation of the global mass matrix M by

$$M_{i,j}^{L} = \begin{cases} \sum_{n} M_{i,n} &, i = j, \\ 0 &, i \neq j, \end{cases}$$
(15)

that is the diagonal elements of the matrix M^L are the row-sums of M. To formulate the finite element method for system (5 -7) we use the standard Galerkin finite element method in space and the forward finite difference approximation to the first order time derivative. We introduce finite element spaces V_h , W_h for u_1 , u_2 , consisting of standard piecewise linear continuous functions on a mesh and satisfying Dirichlet boundary conditions. Let V_h^0 , W_h^0 denote the corresponding finite element spaces satisfying homogeneous Dirichlet boundary conditions. The finite element method now reads: find (u_{1h}, u_{2h}) in $(V_h \times W_h)$ such that

$$\left(\frac{u_{1h}^{k+1} - u_{1h}^{k}}{\tau}, v\right) + \left(a^{2}\nabla u_{2h}^{k}, \nabla v\right) = (f^{k}, v), \quad \forall v \in V_{h}^{0}, \quad (16)$$
$$\left(\frac{u_{2h}^{k+1} - u_{2h}^{k}}{\tau}, w\right) = (u_{1h}^{k}, w), \quad \forall w \in W_{h}^{0}. \quad (17)$$

This produces the system of linear equations for model (5-7) at each time

step :

$$M\mathbf{u}_{1}^{k+1} = (F^{k} - a^{2}K\mathbf{u}_{2}^{k})\tau + M\mathbf{u}_{1}^{k}, \qquad (18)$$

$$M\mathbf{u}_2^k = M\mathbf{u}_1^k \tau + M\mathbf{u}_2^k, \tag{19}$$

$$\mathbf{u}_1^0 \mid_{\Gamma} = 0 \tag{20}$$

$$\mathbf{u}_2^0 \mid_{\Gamma} = 0 \tag{21}$$

In these equations, M and K are the same matrices, as in the system (9), k denotes the time level, u_1 and u_2 are the unknown discrete field values of u_1 and u_2 , τ is the time step size, Γ is the boundary of the *inner* region.

To obtain an explicit scheme we approximate M by M^L and multiply the first of the system equations by $(M^L)^{-1}$ so that the system can be rewritten in the more efficient form:

$$\mathbf{u}_{1}^{k+1} = ((M^{L})^{-1}F^{k} - a^{2}(M^{L})^{-1}K\mathbf{u}_{2}^{k})\tau + \mathbf{u}_{1}^{k}, \qquad (22)$$

$$\mathbf{u}_2^{k+1} = \mathbf{u}_1^k \tau + \mathbf{u}_2^k. \tag{23}$$

The disadvantage with explicit schemes is of course that we must choose small time steps to respect a CFL criterion:

$$\tau \le \frac{h}{ac} \quad , \tag{24}$$

where h is the minimal local mesh size of the elements, and c is a constant.

Finite difference formulation

In the Ω_{FDM} domain, we use FDM. The FDM stencil can be derived via the FEM schemes presented in the previous section, when applied to a structured Cartesian grid. For problem (1) we obtain

$$u_{i,j,k}^{l+1} = \tau^2 (f_{i,j,k}^l + a^2 \Delta u_{i,j,k}^l) + 2u_{i,j,k}^l - u_{i,j,k}^{l-1},$$
(25)

where $u_{i,j,k}^l$ is the solution on time iteration l at point (i, j, k), $f_{i,j,k}^l$ is the source function, τ is the time step, and $\Delta v_{i,j,k}^l$ is the discrete Laplacian. In three dimensions, we get the standard seven-point stencil:

$$\Delta v_{i,j,k}^{l} = \frac{v_{i+1,j,k}^{l} - 2v_{i,j,k}^{l} + v_{i-1,j,k}^{l}}{dx^{2}} + \frac{v_{i,j+1,k}^{l} - 2v_{i,j,k}^{l} + v_{i,j-1,k}^{l}}{dy^{2}} + \frac{v_{i,j,k+1}^{l} - 2v_{i,j,k}^{l} + v_{i,j,k-1}^{l}}{dz^{2}}, \qquad (26)$$

where dx, dy, and dz are the steps of the discrete finite difference meshes.

Absorbing boundary conditions

We have also simulated a variation of the problem (1-4) with Dirichlet boundary condition replaced by the absorbing boundary condition. It means, that this boundary conditions approximate the solution on the boundaries. We use the following boundary condition:

$$\frac{\partial}{\partial t}u - \frac{\partial}{\partial x}u\Big|_{x=0} = 0.$$
(27)

We are using forward finite difference approximation in the middle point of the condition (27), which gives a numerical approximation of higher order than ordinary (backward or forward) approximation. For example, for the left boundary of the *outer* domain we obtain:

$$\frac{u_{i,j,k}^{l+1} - u_{i,j,k}^{l}}{dt} + \frac{u_{i+1,j,k}^{l+1} - u_{i+1,j,k}^{l}}{dt} - \frac{u_{i+1,j,k}^{l} - u_{i,j,k}^{l}}{dx} - \frac{u_{i+1,j,k}^{l+1} - u_{i,j,k}^{l+1}}{dx} = 0,$$
(28)

which can be transformed to

$$u_{i,j,k}^{l+1} = u_{i+1,j,k}^{l} + u_{i,j,k}^{l} \frac{dx - dt}{dx + dt} - u_{i+1,j,k}^{l+1} \frac{dx - dt}{dx + dt}.$$
 (29)

For other boundaries of the *outer* domain we find analogous boundary conditions.

Performance comparisons

We investigate the performance of the different methods by computing, with each method, the wave equation on structured grids, and measuring the cpu time per node and per iteration.

The size of the used computational grids are shown in Table 1. The performance tests were performed on a Sun workstation with free memory size 773Mb and 2048Mb real memory.

Table 2 and Table 3 present efficiency results, in terms of cpu time per node and iteration. The fractions FEM/Hybrid and FEM/FDM are also presented in the tables. We note that, for two dimensions, the fraction FEM/Hybrid ≈ 3.2 and the fraction FEM/FDM ≈ 3.7 . In our three-dimensional tests, the corresponding fractions have increased. Here, the fraction FEM/Hybrid ≈ 4.4 , and the fraction FEM/FDM is around 6.7.

The tables show that the fractions increase with the size of the grid. This can be explained by cache effects, since the required memory of the FEM sparse matrix is much larger than the corresponding FDM difference molecule. Another effect of importance is that the nodes at the boundary is making up a smaller part of the total number of nodes. For the hybrid method, the relative cost associated with computing the solution in the overlap region with both methods and exchanging solution values, decreases as the grid sizes increase, compare with Table 1.

Remarks on the performance comparisons

For our test cases, the source function evaluations in the simulations are a minor part of the execution time, since f_1 is nonzero only in a small fraction of the time steps. Since the source function evaluations are identical for FEM and FDM methods, would an expensive source function evaluation results in a decreased fraction FEM/FDM.

The experiments are made on structured grids for which it was possible to use reduced FEM matrices. We define the performance P for all the methods as

$$P = \frac{T_{tot}}{N_T \cdot N_\Omega},\tag{30}$$

where T_{tot} is total computational time, N_T is number of the timesteps, and N_{Ω} is number of the nodes in the mesh.

The ratios FEM/FDM in Tables 2 and 3 are essentially due to the difference in execution time of multiplying a row of a sparse matrix

compared to applying the corresponding finite difference molecule. Numerical experiments (not presented here) indicate that the execution time for a sparse matrix multiplication increases linearly with the number of coefficients per row. Therefore, experiments performed with unreduced FEM matrices would result in an increase of the fraction FEM/FDM, approximately with the factors 7/5 and 15/7, for 2D and 3D simulations, respectively. The factors come from the matrix reduction sizes. A corresponding increase would occur for the fraction FEM/Hybrid.

Memory consumption

An important issue is the memory consumption of the FEM version versus the FDM and hybrid implementations. The FDM implementation is here advantageous in two respects. First, the FDM grid representation uses much less memory than the corresponding unstructured FEM grid. Second, a finite difference molecule is used instead of a memory consuming sparse matrix. These two advantages for the hybrid method regarding the memory consumption is probably, for many applications, more important than the speed-up presented above.

size of the mesh elements, h	numberofnodesin Ω_{FEM}	number of nodes in Ω	number of nodes in the overlapping layers	spatial di- mension
0.0025	6561	160801	3192	2
0.005	1681	40401	1592	2
0.01	441	10201	152	2
0.02	121	2601	72	2
0.01	9261	1030301	4348	3
0.02	1331	132651	988	3
0.04	216	17576	208	3

Table 1: Meshes for the performance test.

h	Hybrid	FEM	FDM	FEM/Hybrid	FEM/FDM
0.0025	9.70385e-7	3.25877e-6	8.11587e-7	3.3582	4.0153
0.005	9.25238e-7	3.05127e-6	8.09583e-7	3.2978	3.7689
0.01	9.27085e-7	2.81574e-6	8.09961e-7	3.0372	3.4764
0.02	9.31629e-7	2.69945e-6	7.90749e-7	2.8976	3.4138

Table 2: Performance for the 2D wave equation

h	Hybrid	FEM	FDM	FEM/Hybrid	FEM/FDM
0.01	4.84596e-6	2.21148e-5	3.05454e-6	4.5636	7.2400
0.02	4.88609e-6	2.15223e-5	3.28249e-6	4.4048	6.55670
0.04	4.73657e-6	2.02856e-5	3.17052e-6	4.2828	6.39819

Table 3: Performance for the 3D wave equation