Curvilinear DGTD method for nanophotonics applications

Jonathan Viquerat, Claire Scheid

Book of abstracts of the 6th International Conference
on Advanced Computational Methods
in Engineering, ACOMEN 2014
23–28 June 2014.

Curvilinear DGTD method for nanophotonics applications

Jonathan Viquerat[1] **and Claire Scheid**[1,2]

[1] *Nachos project-team, INRIA Sophia Antipolis Méditerranée*

[2] *J.-A. Dieudonné laboratoy, University of Nice Sophia Antipolis*

e-mails: jonathan.viquerat@inria.fr, claire.scheid@unice.fr

Abstract

Classical finite element methods rely on tessellations composed of straight-edged elements mapped linearly from a reference element, on domains which physical boundaries are indifferently straight or curved. This approximation represents serious hindrance for high-order methods, since they limit the precision of the spatial discretization to second order. Thus, exploiting an enhanced representation of the physical geometry of a considered problem is in agreement with the natural procedure of high-order methods, such as the discontinuous Galerkin method. In the latter framework, we propose and validate an implementation of a high-order mapping for tetrahedra, and then focus on specific nanophotonics setups to assess the gains of the method in terms of memory and performances.

Key words: Discontinuous Galerkin, curvilinear elements, Maxwell equations, nanophotonics

1 Curvilinear DGTD formulation

Classical discontinuous Galerkin time-domain (DGTD) methods rely on a linear mapping from a straight-edged reference element to each physical element of the mesh to evaluate the expressions of the finite-element matrices : this allows to save a lot in terms of computational efficiency and memory consumption. Indeed, in the linear case, the finite element matrices for the physical elements are simply multiples of the precalculated matrices of the reference element, since the Jacobian of the corresponding transformation is a constant. In a curvilinear setting, the reference element is mapped to the physical element *via* a quadratic form, thus allowing a quadratic representation of boundaries. Therefore, the Jacobian of this transformation is no longer a constant, and the matrices have to be evaluated by means of numerical integration, and stored for each physical curved tetrahedron. Efficient quadrature and cubature rules can be easily found up to sufficient order to our purposes.

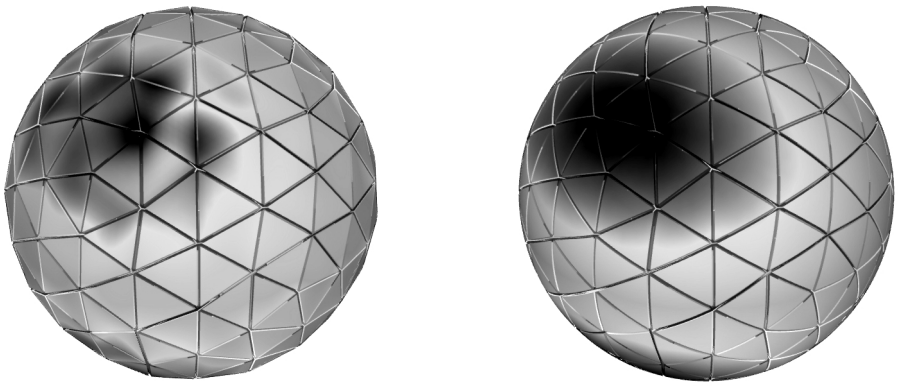
A DGTD scheme accouting for curved elements was formulated and implemented in the framework of Maxwell's equations, using centered numerical fluxes. A validation step was conducted to verify the stability and accuracy of the method. Realistic situations related to the nanophotonics field will then be considered that demonstrate the potential of the approach, such as realistically-rounded metallic nanocubes described by a multipole dispersive model.

2 Validation

A spherical cavity of unit radius is considered, with PEC boundary conditions. A $(0, 1, 1)$ mode is propagated inside it, which exact solution is known. Four different rectilinear meshes of increasing refinement were generated in order to check for h-convergence. $\mathbb{P}_1$ to $\mathbb{P}_4$ polynomial approximations were used. For each simulation, the L^2 error is calculated over the whole mesh, and the maximum error levels is retained. The obtained h-convergence results are displayed on table 1. As can be seen, the use of curvilinear tetrahedra restores quasi-optimal rates. Moreover, it allows to save a lot in terms of degrees of freedom, and therefore in CPU time. Indeed, the curvilinear M1 solution is three times faster and occupies three times less memory than the linear M2 solution, for a roughly similar error level. As an example, the $\mathbb{P}_2$ numerical solution on the M1 and M2 meshes are displayed on figure 2.

Table 1: **Convergence rates of the spherical cavity case.**

	M1		**M2**		**M3**		**M4**	
	Rect.	Curv.	Rect.	Curv.	Rect.	Curv.	Rect.	Curv.
$\mathbb{P}_1$	–	–	1.62	–	1.25	–	1.16	–
$\mathbb{P}_2$	–	–	1.83	2.85	1.70	2.32	1.58	2.28
$\mathbb{P}_3$	–	–	1.75	3.84	1.66	3.53	1.55	3.41
$\mathbb{P}_4$	–	–	1.78	3.79	1.68	3.73	1.57	3.57



(a) Linear mesh (b) Quadratic mesh

Figure 1: $\mathbb{P}_2$ **numerical solution for the** E_x **field.**

Acknowledgements

We would like to thank the French DGA (Defence procurement agency) for partial funding of J. Viquerat's PhD thesis.