PRAgMaTIc – Parallel Anisotropic Adaptive Mesh Toolkit

Georgios Rokos[1] and Gerard Gorman[2]

[1] Software Performance Optimisation Group, Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK
georgios.rokos09@imperial.ac.uk
http://www.doc.ic.ac.uk/~gr409/
[2] Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, UK
g.gorman@imperial.ac.uk
http://www3.imperial.ac.uk/people/g.gorman

Abstract. The numerical methods used to model complex geometries required by many scientific applications often favour the use of unstructured meshes and finite element discretisation methods over structured grid alternatives. This flexibility introduces complications, such as the management of mesh quality and additional computational overheads arising from indirect addressing [5]. Using the Finite Element Method for the numerical solution of PDEs, a posteriori error estimations on the PDE solution help evaluate a quality functional [4] and determine the low-quality mesh elements. Mesh adaptivity methods ([3], [1]) provide an important means to control solution error by focusing mesh resolution in regions of the computational domain when and where it is required.

Adaptive algorithms are grouped into two main categories, h-adaptivity and r-adaptivity algorithms. The first category contains techniques which try to adapt the mesh by changing its topology. This can be done by removing existing mesh elements, a technique known as coarsening, increasing local mesh resolution by adding new elements, a procedure called refinement, or replacing a group of elements with a different group, which can be achieved through swapping. The second group of adaptive algorithms encompasses a variety of vertex smoothing techniques, all of which leave mesh topology intact and only attempt to improve quality by relocating mesh vertices. Algorithm 1 demonstrates the general procedure for the solution of a PDE on an adaptive mesh.
A problem is said to be anisotropic if its solution exhibits directional dependencies. An anisotropic mesh contains elements which have some suitable orientation. In this case, the error estimation is given in the form of a metric tensor field $M(\mathbf{x})$, i.e. a tensor which, for each point in the domain, represents the desired length and orientation of a mesh edge containing this point. Adapting a mesh so that it distributes the error uniformly over the whole mesh is equivalent to constructing a uniform mesh consisting of equilateral triangles with respect to the non-Euclidean metric $M(\mathbf{x})$.

R. Keller et al. (Eds.): Facing the Multicore-Challenge III 2012, LNCS 7686, pp. 143–144, 2013.

Algorithm 1. General algorithm for the adaptive solution of PDEs

Mesh $\mathcal{M}_0 \leftarrow$ initial auto-generated mesh
solve PDE on $\mathcal{M}_0$
$\mathcal{E}_0 \leftarrow$ a posteriori estimation of solution error
while $\mathcal{E}_i \geq$ *predefined_tolerance* **do**
  compute metric tensor field $\mathcal{T}_i$ from $\mathcal{E}_i$
  perform initial coarsening on $\mathcal{M}_i$
  repeat
    perform refinement on $\mathcal{M}_i$
    perform coarsening on $\mathcal{M}_i$
    perform swapping on $\mathcal{M}_i$
    $L_{max} \leftarrow$ longest mesh edge
  until (pre-defined number of iterations is reached) **or**(L_{max} - $\sqrt{2.0} < 0.01$)
  perform smoothing on $\mathcal{M}_i$
  solve PDE on $\mathcal{M}_i$
  $\mathcal{E}_i \leftarrow$ a posteriori estimation of solution error

PRAgMaTIc is an open-source mesh adaptivity framework, built with large-scale multiprocessing in mind. It implements coarsening, refinement and swapping alongside an optimisation-based vertex smoothing algorithm proposed by Freitag et al. [1]. Parallel execution is based on an older parallel framework [2], improved through a novel approach which combines the idea of mesh partitioning with low-level intervention in mesh data structures in order to achieve good data locality, high performance and thread safety. PRAgMaTIc supports both NUMA (via OpenMP) and distributed-memory (via MPI) systems. Current work is on improving performance and scalability of adaptive algorithms. Support for CUDA/OpenCL is planned for the near future. PRAgMaTIc can be downloaded from Launchpad under the BSD licence: `https://launchpad.net/pragmatic`.

References

1. Freitag, L., Jones, M., Plassmann, P.: An efficient parallel algorithm for mesh smoothing. In: Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, pp. 47–58. Citeseer (1995)
2. Freitag, L.F., Jones, M.T., Plassmann, P.E.: The Scalability Of Mesh Improvement Algorithms. In: IMA Volumes in Mathematics and its Applications, pp. 185–212. Springer (1998)
3. Li, X., Shephard, M., Beall, M.: 3d anisotropic mesh adaptation by mesh modification. Computer Methods in Applied Mechanics and Engineering 194(48-49), 4915–4950 (2005)
4. Vasilevskii, Y., Lipnikov, K.: An adaptive algorithm for quasioptimal mesh generation. Computational Mathematics and Mathematical Physics 39(9), 1468–1486 (1999)
5. Piggott, M.D., Farrell, P.E., Wilson, C.R., Gorman, G.J., Pain, C.C.: Anisotropic mesh adaptivity for multi-scale ocean modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 367(1907), 4591–4611 (1907)