Adapted *a posteriori* meshes

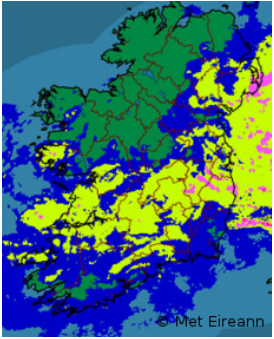
Róisín Hill, Niall Madden

National University of Ireland, Galway
Postgraduate Modelling Research Group

12 October, 2018

Introduction

- Partial differential equations (PDEs) represent many real life phenomena.
- There is often some subregion of the domain that is of particular interest.
- The goal is to resolve small scale effects without excessive computational cost.



What can we do?

Our aim is to compute accurate numerical solutions:

- We divide the domain into a finite set of subregions ("the mesh").
- Then solve a discrete version of the PDE on this mesh.

When resolving locally small effects, using a uniform mesh can be computationally expensive.

Choosing our mesh

1. If we know the location of the regions of interest, we can
   - generate a non-uniform mesh which is very fine in these areas, and then solve the problem - *a priori* mesh.
2. Alternatively, we can
   - solve the problem initially, for example, on a uniform mesh, and use information gained from the solution to create a more suitable mesh - *a posteriori* mesh.

a posteriori meshes

- Methods to refine the mesh to better resolve regions of interest, include:

    1. *h*-refinement: reduce the local mesh width (h) in regions of interest,

    2. *r*-refinement: mesh points are relocated to these areas, but the total number of points is unchanged.

a posteriori meshes

- Methods to refine the mesh to better resolve regions of interest, include:

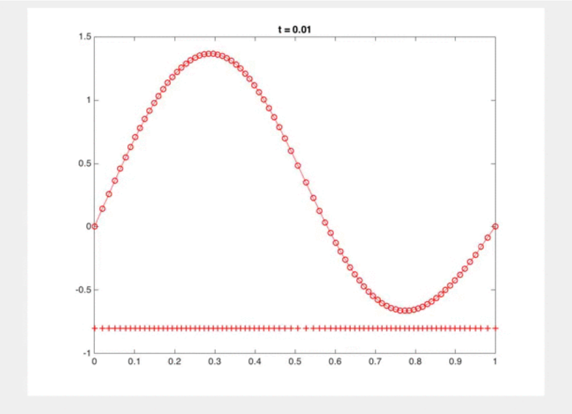
  1. *h*-refinement: reduce the local mesh width (*h*) in regions of interest,

  2. *r*-refinement: mesh points are relocated to these areas, but the total number of points is unchanged.

We'll focus on *r*-refinement.

1-d Burger's equation

Solution and mesh for the one-dimensional Burger's equation, from [Huang, 2018]



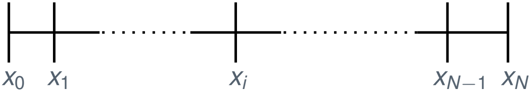
Implementation of r-refinement method

Equidistributing a monitor function

A monitor function is an arbitrary, strictly positive function defined on the domain Ω, that is used to indicate where a mesh should be fine/coarse. A classic choice is

$$M_{arc}(x) = \sqrt{1 + (u'(x))^2},$$

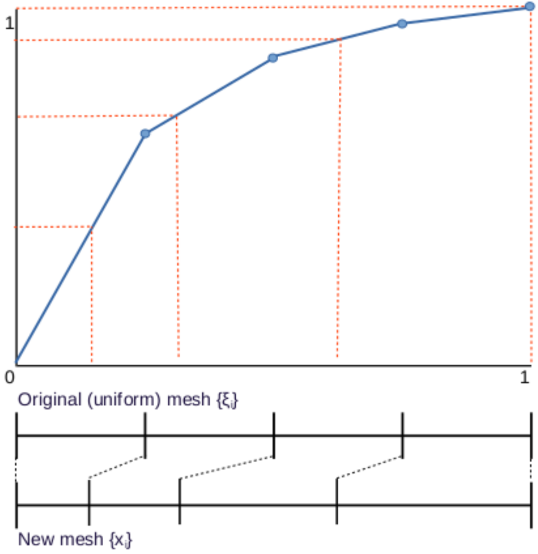
where u represents the solution of the problem. A mesh $\{x_i\}$,



equidistributes $M(\cdot)$ when

$$\int_{x_{i-1}}^{x_i} M(x)dx = \frac{1}{N}\int_{\Omega} M(x)dx, \text{ for } i = 1, 2, ..., N.$$

Equidistributed arc-length monitor function



Methods to equidistribute the monitor function

The de Boor Algorithm [de Boor, 1973]:

This generates a sequence of meshes by constructing an interpolant to the monitor function and then computing the mesh that equidistributes it. Since M is a strictly increasing function, this problem always has a solution.

- Advantages include its simplicity, reliability and efficiency.
- The main disadvantage is that it does not extend easily to higher dimensions.

Moving meshes in Banff, Canada

Moving meshes in Banff, Canada

Methods to equidistribute the monitor function

Moving Mesh PDE (MMPDE) [Huang and Russell, 2011]:

The equidistributed mesh is obtained by integrating a parabolic differential equation, such as

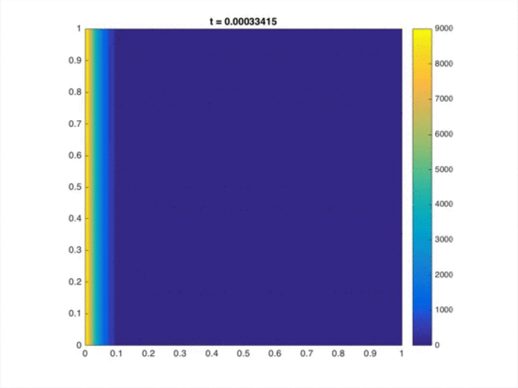
$$\frac{\partial}{\partial \xi}\left(M(x)\frac{\partial x}{\partial \xi}\right) = 0. \tag{1}$$

To implement this numerically we iteratively solve the nonlinear weak form of (1),

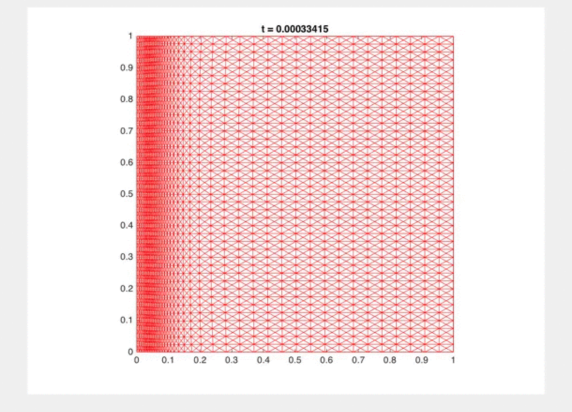
$$\int_{\Omega}(M(x)x'v')d\xi = 0. \tag{2}$$

- Advantages include being suitable to extend to higher dimensions.
- Disadvantages include that it is computationally more expensive than the de Boor Algorithm.

Radiation diffusion solution from [Huang, 2018]



Related mesh movement from [Huang, 2018]



Conclusion

- The choice of mesh depends on the problem being solved.
- An *a posteriori* mesh generated using an r-refinement method is an appropriate mesh for some problems:
  - it automatically concentrates mesh points in regions of interest;
  - it maintains the same mesh topology throughout;
  - it controls the quality of the solution locally.

Future work

- To date we have computed accurate solutions for one-dimensional singularly perturbed reaction-diffusion equations on meshes generated with the r-refinement method using FEniCS [Langtangen and Logg, 2016].
- Our next step is to extend this work to one-dimensional reaction-convection-diffusion equations, and subsequently to problems on higher dimensional domains.

References

[Budd et al., 2009] Budd, C. J., Huang, W., and Russell, R. D. (2009).
Adaptivity with moving grids.
Acta Numer., 18:111–241.

[de Boor, 1973] de Boor, C. (1973).
Good approximation by splines with variable knots.
In *Spline functions and approximation theory*, pages 57–72. Springer.

[Huang, 2018] Huang, W. (2018).
`http://whuang.faculty.ku.edu/research/movingmesh.html`.

[Huang and Russell, 2011] Huang, W. and Russell, R. D. (2011).
Adaptive moving mesh methods, volume 174 of *Applied Mathematical Sciences*.
Springer, New York.

[Langtangen and Logg, 2016] Langtangen, H. P. and Logg, A. (2016).
Solving PDEs in Minutes–The FEniCS Tutorial Volume I.
Springer.

Thank you for listening, any questions!