UCRL-ID-140768

AMR vs High Order Schemes, Wavelets as a Guide

L. Jameson

October 18, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

Approved for public release; further dissemination unlimited

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Work performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
http://apollo.osti.gov/bridge/

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
http://www.ntis.gov/

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
http://www.llnl.gov/tid/Library.html

AMR vs High Order Schemes
Wavelets as a Guide

Leland Jameson

Lawrence Livermore National Laboratory
University of California
P.O. Box 808, MS L-312
Livermore, CA 94551
Email: jameson3@llnl.gov

Abstract

The final goal behind any numerical method is give the smallest wall-clock time for a given final time error or, conversely, the smallest run-time error for a given wall clock time, etc. Here a comparison will be given between adaptive mesh refinement schemes and non-adaptive schemes of higher order. It will be shown that in three dimension calculations that in order for AMR schemes to be competitive that the finest scale must be restricted to an extremely, and unrealistic, small percentage of the computational domain.

Contents

1 Introduction **4**

2 Adaptive Numerical Methods **5**
2.1 Preliminary Notation 5
2.2 Computational Work for AMR and High Order Schemes . . . 6
2.2.1 Work of the High Order Scheme 6
2.2.2 Work of the Adaptive Scheme 7
2.2.3 Grid Point Density, Order and Error 9
2.2.4 Example 1: 2nd Order AMR, 4th Order Uniform . . . 10
2.2.5 Example 2: 2nd Order AMR, 8th Order Uniform . . . 12
2.2.6 Example 3: 4nd Order AMR, 8th Order Uniform . . . 12
2.2.7 Comments on Spectral Methods 13
2.2.8 AMR Overhead 13

3 A Tighter Estimate **14**
3.1 Confirmation of Estimate 16
3.2 A Few Examples 16

4 Wavelet-Based Criteria for Adaptation **18**
4.1 Fraction at a Given Scale 18

5 Conclusion **21**

List of Figures

1 An example of a flame front. 19
2 The grid for the flame front indicating the fraction of the domain at a given grid point density. 20

1 Introduction

In the numerical solution of partial differential equations one has many choices of solution techniques. Here we address this issue of when it is appropriate to choose and adaptive numerical method over a non-adaptive numerical method. Adaptive numerical methods are considered to be advantageous when the scales in the flow variables differ greatly throughout the computational domain. A good example is a propagating flame front in which the relevant physics is restricted to the very localized area of the burn or region of combustion , see ([2]), ([4]). In this case, the adaptive numerical method places many grid points in the combustion region and the physics can essentially be resolved. The critical issue here is that the flame front be restricted to an extremely small percentage of the domain and this restriction is tighter and tighter as one increases the dimension of the computation.

Here precise estimates will be given on savings that one can expect to obtain from an AMR scheme and when, and if, an AMR scheme should be considered at all for a given computation. Note that the acronym AMR is often used to denote a particular type of adaptive method. But, here, AMR will be used to denote any scheme which maintains a fixed order of approximation and which adapts the mesh in order to reduce the truncation error in specific parts of the computational domain. Consequently AMR will denote not only traditional "AMR" schemes but also schemes which are wavelet based, etc.

This manuscript provides two estimates. The first estimate allows is very simple to make and considers only the work of the AMR scheme at the finest computational scale. This is certainly unrealistic but provides a first easy bound on work. Our second estimate is more realistic and does consider the work in the remainder of the domain. Of course the second estimate is much tighter than the first.

2 Adaptive Numerical Methods

As mentioned in the introduction, AMR schemes change the grid point density in various regions of the computational domain so that the truncation error can be reduced in regions, say, where steep gradients exist, etc. In this presentation, we are not concerned with the mechanism by which the grid is chosen, but only on the work after a computational grid has been prescribed. Note that the only reason to consider a AMR scheme is to reduce work for a given computation. One, therefore, hopes to be able to do bigger problems which can not be executed on a uniform grid. Here we examine the issue of computation carefully in the context of comparing AMR to numerical schemes of higher order.

2.1 Preliminary Notation

If we let j denote the scale in our adaptive calculation, so that,

$$\Delta x_j = 2^j \Delta x_f, \tag{1}$$

where Δx_f denotes the grid spacing at the finest scale, and if N denotes the number of grid points in each direction on the finest scale of a d dimensional calculation, then the number of grid points at coarser scales will be,

$$Grid = \left(\frac{N}{2^j}\right)^d. \tag{2}$$

Next, if we assume the equations are hyperbolic or dominated by hyperbolic effects, then we can expect that,

$$\Delta t = c\Delta x_f, \tag{3}$$

and that the time step on coarser scales will be,

$$\Delta t_j = 2^j \Delta x_f. \tag{4}$$

Further, let us denote by $R_j(t)$ the fraction of the domain at resolution j so that,

$$1 = \sum_{j=0}^{J} R_j(t). \tag{5}$$

The meaning of R_j can be seen from Figures (1) and (2). In Figure (1) we can see a steep gradient region and a Gaussian shaped perturbation in the field. In Figure (2) we can see the wavelet-selected grid. This grid contains four different grid point densities. But the majority of the domain is dominated by two densities: the finest and the coarsest with the two middle densities having very little representation. If this domain were covered by the finest grid everywhere, there would be 128 by 64 grid points total. In Figure (2) we can simply count the number of points at the finest scale, and we see that there are about 1200 such fine scale grid points and thereby they comprise roughly 15% of the domain. An automatic procedure using wavelet analysis will soon be introduced.

2.2 Computational Work for AMR and High Order Schemes

Here we will build the expressions for the work for one time step on the grid of the high order scheme, and we will assume that the coarsest scale of the adaptive calculation is at the same grid point density as that of the high order scheme. This assumption is merely for convenience for a slightly more compact final expression.

2.2.1 Work of the High Order Scheme

Again we assume that N denotes the grid point density at the finest scale in each direction of the adaptive calculation. And, we assume that at the final time that our high order scheme at order $b * O_s$ and at grid point density $\frac{N}{a}$ yields that same error as the adaptive calculation. O_s will denote the order of the adaptive scheme. Our expression for cost of one time step of the High Order scheme is,

$$HO_{cost} = \left(\frac{N}{a}\right)^d * b * O_s * O_t, \tag{6}$$

where O_t denotes the order of the temporal discretization.

2.2.2 Work of the Adaptive Scheme

Since the scheme is *adaptive*, we do expect that this fraction will be a function of time. The cost of one time step on the finest grid will be roughly,

$$cost = N^d * O_s * O_t. \tag{7}$$

Now we will assume that the scheme has been optimized so that one takes no more times steps on coarse scales than is necessary. So, if at a given period during our calculation, it is such that there are 3 scales, let J denote the maximum scale, the finest scale $j = 0$, the second finest scale $j = 1$, and the coarsest scale $j = J = 2$. Then we expect that 4 time steps must be taken on scale $j = 0$ and 2 time steps at scale $j = 1$ for each time step taken taken on the coarsest scale $j = 2$. We can now construct an expression which is proportional to the number of flops during one time step on the coarsest scale. We sum up the pieces from the various scales, beginning with the finest,

$$fine - scale - work = N^d * 2^J * O_s * O_t, \tag{8}$$

where 2^J is needed because we must take more time steps on finer scales to equal one time step on a coarse scale. Similarly, one time step on the next to finest scale would be,

$$second - finest - scale - work = \left(\frac{N}{2}\right)^d * 2^{J-1} * O_s * O_t. \tag{9}$$

Our expression for one time step on the coarsest scale then becomes,

$$AS_{cost} = \sum_{j=0}^{J} R_j \left(\frac{N}{2^j}\right)^d 2^{J-j} * O_s * O_t. \tag{10}$$

In order to choose an adaptive scheme over a scheme of higher order, a requirement is that the work be less for the adaptive scheme. That is, we require that $HO_{cost} > AS_{cost}$. The grid point density difference between the high order scheme and the adaptive scheme is such that the high order scheme has $\frac{N}{a}$ points in each direction and therefore Δx and Δt are proportional to $\frac{a}{N}$. We assume the same CFL for the adaptive scheme and the high order scheme. We must find how many time steps we must take on the grid of the

high order scheme in order to equal one time step of the adaptive scheme at the coarse scale,

$$\frac{a}{N} * TS = \frac{2^J}{N}, \tag{11}$$

where TS denotes number of time steps. So we must take,

$$TS = \frac{2^J}{a} \tag{12}$$

time steps on the high order grid in order to equal one time step on the coarse scale grid.

We, therefore, require that

$$\frac{2^J}{a} * \left(\frac{N}{a}\right)^d * b * O_s * O_t > \sum_{j=0}^{J} R_j \left(\frac{N}{2^j}\right)^d 2^{J-j} * O_s * O_t. \tag{13}$$

First we see that the variables representing temporal and spatial order drop out to get,

$$\frac{2^J}{a} * \left(\frac{N}{a}\right)^d * b > \sum_{j=0}^{J} R_j \left(\frac{N}{2^j}\right)^d 2^{J-j}, \tag{14}$$

and take the grid density N and 2^J out of the summation,

$$N^d * \frac{1}{a} * \frac{b}{a^d} > N^d \sum_{j=0}^{J} R_j \left(\frac{1}{2^j}\right)^d 2^{-j}. \tag{15}$$

Eliminating the grid point density we arrive at,

$$\frac{b}{a^{d+1}} > \sum_{j=0}^{J} R_j 2^{-j(1+d)} > R_0. \tag{16}$$

In the above inequality we have used the fact that each member of the summation is non-negative and is therefore smaller than the full summation. To use the last inequality is extremely generous for AMR schemes since it says that the other members make no contribution to the summation all. In the next section, we obtain much more realistic and tighter estimate. But, for now a minimal criteria for the fraction of the calculation at the finest scale is that,

$$\frac{b}{a^{d+1}} > R_0. \tag{17}$$

Note that this estimate simply bounds the maximum work which can come from the finest scale and ignores contributions from other scales. Later we will consider the contribution from other scales and the bound will become much tighter. But, we will see that even this very loose and generous bound is very restrictive.

2.2.3 Grid Point Density, Order and Error

Choosing the variables b and a will depend on many issues: the total runtime, the final error criteria and the spectrum of the flow variables. That is, the calculations which run a long time and calculations which have a tight error tolerance strongly favor high order schemes since the number of grid points, and hence wall-clock time, scale as, see [3],

$$N = constant * \left(\frac{Runtime}{Error}\right)^{1/O_s}, \tag{18}$$

where we have assumed quite reasonably that the dominate source of error is spatial and not temporal. We give computational examples to find the number of grid points such that a lower order scheme gives the final runtime error which is the same as a higher order scheme. Our computational example is that of,

$$U_t(x,t) = U_x(x,t) \tag{19}$$

with periodic boundary conditions,

$$U(0,t) = U(2\pi,t) \tag{20}$$

and an initial condition of a Gaussian,

$$U(x,0) = e^{-d(x-\pi)^2}, \tag{21}$$

for $x \in (x, 2\pi)$. The time advancement is 4th order Runge-Kutta with the CFL is set to .5. With this high order time advancement we can expect that that the spatial errors are larger than the temporal errors. We simply find the number of grid points so that the calculation yields roughly the same final L_2 and L_∞ errors at the final time. One can always estimate these numbers theoretically, but a simple example is sufficient.

Scheme	N	L_2 Error	L_∞ Error	Cycles	Time Steps	IC
FD_2	128	.0044	.0099	1	256	Gaussian d=1
FD_4	32	.0041	.0096	1	64	Gaussian d=1
FD_8	18	.0035	.0070	1	36	Gaussian d=1

Table 1: A Gaussian pulse propagated 1 cycle around a periodic domain with $u_t = u_x$. L_2 Error tolerance is set to .004.

Scheme	N	L_2 Error	L_∞ Error	Cycles	Time Steps	IC
FD_2	256	.0011	.0025	1	512	Gaussian d=1
FD_4	44	.0012	.0028	1	88	Gaussian d=1
FD_8	22	.0010	.0022	1	44	Gaussian d=1

Table 2: A Gaussian pulse propagated 1 cycle around a periodic domain with $u_t = u_x$. L_2 Error tolerance is set to .001.

2.2.4 Example 1: 2nd Order AMR, 4th Order Uniform

We first give an example with an error tolerance which is not tight. In this first example that number of grid points required to make the second order calculation equal to the fourth oder calculation is $a = 4$, see Table (1).

And we consider a slightly tighter error tolerance where the error is set to .001 and we can see the number of grid points for the lower order scheme to have the same error as the higher order scheme is almost six times that of the higher order scheme, $a = 5.8$, see Table (2).

Scheme	N	L_2 Error	L_∞ Error	Cycles	Time Steps	IC
FD_4	160	.0000072	.000017	1	160	Gaussian d=1
FD_8	58	.0000073	.000016	1	128	Gaussian d=1

Table 3: A Gaussian pulse propagated 1 cycle around a periodic domain with $u_t = u_x$. L_2 Error tolerance is set to .000007.

Three Dimensions

With these two cases of a and b we can get upper bounds for the fraction of the domain at the finest grid point density in three dimensions.

$$\frac{2}{4^{d+1}} > R_0, \tag{22}$$

If we let the dimension be $d = 3$ we obtain,

$$\frac{1}{128} > R_0. \tag{23}$$

And, for the tighter error tolerance,

$$\frac{2}{(5.8)^{d+1}} > R_0, \tag{24}$$

and then if we let the dimension be $d = 3$ we obtain,

$$\frac{1}{566} > R_0. \tag{25}$$

Two Dimensions

$$\frac{2}{4^{d+1}} > R_0, \tag{26}$$

If we let the dimension be $d = 2$ we obtain,

$$\frac{1}{32} > R_0. \tag{27}$$

And, for the tighter error tolerance we get,

$$\frac{2}{(5.8)^{d+1}} > R_0, \tag{28}$$

and with $d = 2$ we obtain,

$$\frac{1}{98} > R_0. \tag{29}$$

2.2.5 Example 2: 2nd Order AMR, 8th Order Uniform

For this case, there is really no chance that AMR can be competitive even in two dimensions.

Three Dimensions

$$\frac{4}{(11.6)^{d+1}} > R_0, \tag{30}$$

and then if we let the dimension be $d = 3$ we obtain,

$$\frac{1}{4526} > R_0. \tag{31}$$

Two Dimensions

$$\frac{4}{(11.6)^{d+1}} > R_0, \tag{32}$$

and then if we let the dimension be $d = 2$ we obtain,

$$\frac{1}{390} > R_0. \tag{33}$$

2.2.6 Example 3: 4nd Order AMR, 8th Order Uniform

Here we finally encounter a scenario where one might consider AMR in two dimensions and possibly even in three. But once again, the requirement that the fine scales be restricted to less than 3% of the domain is very restrictive, see (3).

Three Dimensions

$$\frac{2}{(2.76)^{d+1}} > R_0, \tag{34}$$

If we let the dimension be $d = 3$ we obtain,

$$\frac{1}{29} > R_0. \tag{35}$$

Two Dimensions

$$\frac{2}{(2.76)^{d+1}} > R_0, \tag{36}$$

and then if we let the dimension be $d = 2$ we obtain,

$$\frac{1}{10.5} > R_0. \tag{37}$$

2.2.7 Comments on Spectral Methods

Our comparisons are focused on the finite difference approach where our cost is $order * N$ for taking a derivative. For a spectral the $order$ will be replaced by a $c \log N$ term for a more efficient calculation of the derivative. But given that a second order AMR can not compete with either 4th order or 8th order non-spectral methods, we have no expectation that it can beat the more efficient spectral approach.

2.2.8 AMR Overhead

Note that the we have assumed no overhead cost for the adapting process which will certainly make the restriction on R_0 much tighter. That is, our entire argument has been biased in favor of adaptive schemes, and even with these biases we obtain that the finest scale must comprise fractions of the domain which are simply too small for the adaptive method to be a candidate.

3 A Tighter Estimate

In the previous section we presented an argument that was very generous to AMR in that the estimate only considered work at the finest scale $j = 0$ and not work at other scales. Certainly the finest scale will not fill up the domain and now we will consider the work from a coarser scale. Still, our argument remains very generous to AMR.

Note that there must be *at least* two scales in the adaptive calculation but not necessarily more. That is, we require at least part of the computation to be at the finest scale in order to satisfy the error requirements. Second, the fine scale computation is too expensive and can not fill up the entire domain. So, there must be at least one other scale present. We seek to find the maximum size of the smallest scale and so we set the second scale to be the cheapest or coarsest scale.

From the previous section we have the expression,

$$\frac{b}{a^{d+1}} > \sum_{j=0}^{J} R_j 2^{-j(1+d)}, \tag{38}$$

The key change in the current estimate is that we will keep two terms in the above summation, namely the first and the last terms but first we must go through some algebra.

The logic for keeping the first and last terms only is that we are looking for bounds on the fraction of the domain at the finest scale and the coarsest scale J is the least costly of all scales allowing scale $j = 0$ to be as large as possible. Recall that,

$$\sum_{j=0}^{J} R_j = 1,$$

so we get,

$$R_J = 1 - \sum_{j=0}^{J-1} R_j.$$

We also need the expression,

$$\sum_{j=0}^{J} R_j 2^{-j(1+d)} = R_0 + R_J 2^{-J(1+d)} + \sum_{j=1}^{J-1} R_j 2^{-j(1+d)}. \tag{39}$$

Combine the previous two expressions to get,

$$\sum_{j=0}^{J} R_j 2^{-j(1+d)} = R_0 + (1 - \sum_{j=0}^{J-1} R_j) 2^{-J(1+d)} + \sum_{j=1}^{J-1} R_j 2^{-j(1+d)}, \tag{40}$$

or,

$$\sum_{j=0}^{J} R_j 2^{-j(1+d)} = 2^{-J(1+d)} + R_0(1 - 2^{-J(1+d)}) + \sum_{j=1}^{J-1} R_j (2^{-j(1+d)} - 2^{-J(1+d)}). \tag{41}$$

But, since $j < J$ we get that,

$$2^{1+d}(2^{-j} - 2^{-J}) > 0. \tag{42}$$

Therefore we can drop the non-negative summation term on the right hand side of Equation (41) to get,

$$\sum_{j=0}^{J} R_j 2^{-j(1+d)} \geq 2^{-J(1+d)} + R_0(1 - 2^{-J(1+d)}), \tag{43}$$

and incorporating Equation (38) we arrive at,

$$\frac{b}{a^{d+1}} \geq 2^{-J(1+d)} + R_0(1 - 2^{-J(1+d)}). \tag{44}$$

With a bit of algebra we get,

$$\frac{b}{a^{d+1}} - \frac{1}{2^{J(1+d)}} \geq R_0 \left(1 - \frac{1}{2^{J(1+d)}}\right). \tag{45}$$

and

$$\frac{b2^{J(d+1)} - a^{d+1}}{a^{d+1}} > R_0(2^{J(d+1)} - 1), \tag{46}$$

and finally we arrive at,

$$\frac{b2^{J(d+1)} - a^{d+1}}{a^{d+1}(2^{J(d+1)} - 1)} > R_0. \tag{47}$$

Note that as J gets large, i.e., there are a large number of levels of refinement, that the above expression tends toward,

$$\frac{b}{a^{d+1}} > R_0, \tag{48}$$

which is the same expression as the previous section.

3.1 Confirmation of Estimate

The above estimate is a bit complicated, but with a simple test case we can see that it does indeed make sense. Consider a case in two dimensions where the adaptive scheme is second order and the high order scheme is 4th order. Also, in order to have the final runtime error, let the grid point ratio be 4. Note, for 3 scales that the coarse scale grid point density corresponds to the grid point density of the high order scheme. One time step of the high order scheme will cost,

$$HO_{cost} = 4(order) * \left(\frac{N}{4}\right)^2. \tag{49}$$

The cost of the adaptive scheme on the finest scale will cost,

$$AS_{fine} = 4(timesteps) * 2(order) * N^2 * R, \tag{50}$$

where R denotes the fraction of the domain at the fine scale. Likewise, the cost of adaptive scheme at the coarsest scale will cost,

$$AS_{coarse} = 2(order) * \left(\frac{N}{4}\right)^2 (1 - R). \tag{51}$$

So, we have,

$$8 * N^2 R + 2 * \left(\frac{N}{4}\right)^2 (1 - R) = 4 * \left(\frac{N}{4}\right)^2. \tag{52}$$

Solving for R we get,

$$R < \frac{1}{63} \tag{53}$$

which one can also obtain from Equation (47).

3.2 A Few Examples

As in the previous section, we will give a few concrete examples. As we mentioned above, longer run times favor higher order schemes. So, we give two cases called short run and long run to illustrate this fact.

Three Dimensions short run

In this case we set $a = 4$, $J = 4$, $d = 3$ and $b = 2$ as we have above that now we obtain that,

$$\frac{1}{255} > R_0, \tag{54}$$

which is roughly twice as tight as our estimate above.

Three Dimensions long run

In this case we set $a = 5.8$, $J = 4$, $d = 3$ and $b = 2$ as we have above that now we obtain that,

$$\frac{1}{1599} > R_0. \tag{55}$$

Two Dimensions short run

In this case we set $a = 4$, $J = 4$, $d = 2$ and $b = 2$ as we have above that now we obtain that,

$$\frac{1}{132} > R_0. \tag{56}$$

Two Dimensions long run

In this case we set $a = 5.8$, $J = 4$, $d = 2$ and $b = 2$ as we have above that now we obtain that,

$$\frac{1}{288} > R_0. \tag{57}$$

4 Wavelet-Based Criteria for Adaptation

In the preceding sections we examined the issue of an adaptive method compared to a method of slightly higher order but which is not adaptive. Our criteria was based on the fraction of the domain at the finest scale for the adaptive scheme. In this section we offer a simple way to find this fraction of the domain. The tool will be wavelet analysis.

4.1 Fraction at a Given Scale

In this subsection we will outline a procedure for using wavelet analysis to find the fraction of the computational domain at a given wavelet scale. Recall that a wavelet coefficient $d_{j,k}$ indicates the deviation of the computational data from a low-order polynomial at a scale j and a location k. Of course, the data will never, except in unusual circumstances, exactly correspond to a low-order polynomial so the user must specify a "wavelet threshold" which we will denote by ϵ. Wavelet coefficients such that,

$$|d_{j,k}| > \epsilon, \tag{58}$$

will denote where grids are to placed, parameter k, and at what density, parameter j. In order to find the fraction of the computational domain at scale j we need the following expression,

$$R_J = \frac{1}{nd_J}\left(nd_J^\epsilon - \sum_{j<J} nd_J^\epsilon \bigcap nd_j^\epsilon\right). \tag{59}$$

The notation is bit complicated. nd_J denotes the total number of wavelet coefficients at scale J throughout the entire domain. nd_j^ϵ denotes the number of wavelet coefficients at scale j which are larger in magnitude than ϵ. And, $nd_J^\epsilon \bigcap nd_j^\epsilon$ denotes the regions of the domain where both $d_{J,k} > \epsilon$ and $d_{j,k} > \epsilon$ simultaneously. The intention of the summation is to count those wavelet coefficients at scale J which occur in regions of the domain which do not contain finer scales.

By using wavelets in this manner we can obtain a simple and reliable method to estimate the fraction of the domain at a given scale.

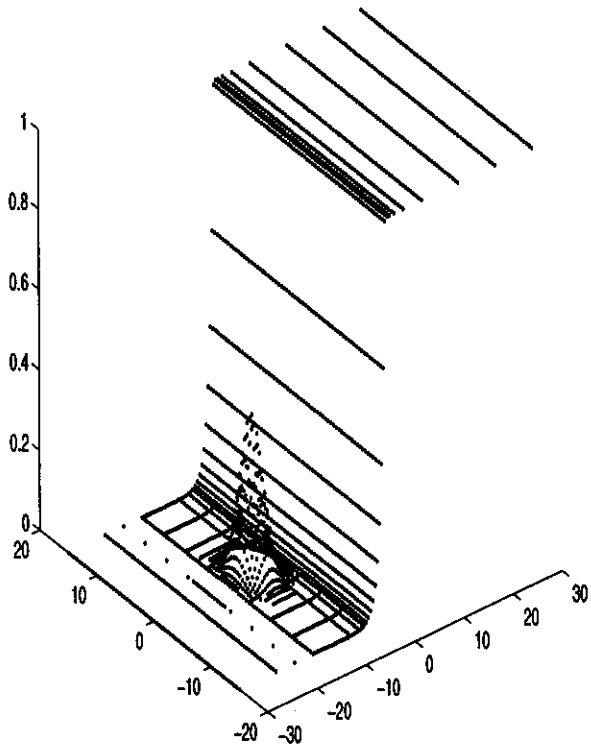

Figure 1: An example of a flame front.

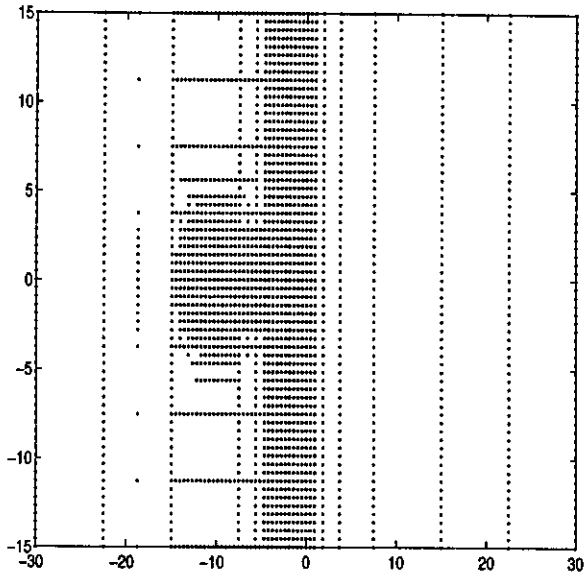

Figure 2: The grid for the flame front indicating the fraction of the domain at a given grid point density.

5 Conclusion

The purpose of this manuscript has been to compare the applicability of AMR schemes when a higher order scheme is an option. We have considered here a comparison of the computational work which is quite fair since AMR schemes are advertised to reduce work by placing grid points only where needed. We conclude that 2nd order AMR schemes have no reasonable chance to compete against schemes of even a slightly higher order such as four. Further, when the 2nd order AMR scheme is compared with an even higher order scheme such as 8th order or perhaps even a spectral method, then AMR should not be considered at all since the percentage of the domain which can be at the finest scale becomes vanishingly small.

On the other hand, as the order of both the adaptive scheme and the non-adaptive scheme increase, their difference becomes smaller. An example of a 4th order AMR scheme and an 8th order non-adaptive scheme was given and it was seen that it might be reasonable to use the AMR scheme for some two dimensional applications. In three dimensions if the order of the schemes are very high, then it will make sense to use an adaptive framework, see [5] and [6].

References

[1] B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, Journal of Computational Physics, v141 (1998), pp.199-224.

[2] L. Jameson, T.L. Jackson, D.G. Lasseigne, "Wavelets as a Numerical Tool", Conference Proceedings from the joint US-Japan workshop on combustion, 1993, J. Buckmaster, T. Takeno (eds.), Springer-Verlag.

[3] H.O. Kreiss and J. Oliger, "Comparison of Accurate Methods for the Integration of Hyperbolic Equations", Tellus XXIV (1972), 3.

[4] D. G. Lasseigne, T. L. Jackson, and L. Jameson, "Stability of Freely Propagating Flames Revisited", Combustion Theory and Modeling, Vol. 3, No. 4, pp. 591-611, Dec. 1999.

[5] J. Hesthaven and L. Jameson, "A Wavelet-Optimized Adaptive Multi-Domain Method", Journal of Computational Physics 145, 280-296 (1998), Article No. CP986012.

[6] L. Jameson, "A Wavelet-Optimized, Very High Order Adaptive Grid and Order Numerical Method", ICASE Report No. 96-30, and SIAM J. Sci. Comput., Vol. 19, No. 6, pp. 1980-2013, November 1998.

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.