On the quality of space-filling curve induced partitions

Gerhard Zumbusch *

Abstract

The solution of partial differential equations on a parallel computer is usually done by a domain decomposition approach. The mesh is split into several partitions mapped onto the processors. However, partitioning of unstructured meshes and adaptive refined meshes in general is an NP-hard problem and heuristics are used. In this paper space-filling curve based partition methods are analysed and bounds for the quality of the partitions are given. Furthermore estimates for parallel numerical algorithms such as multigrid and wavelet methods on these partitions are derived.

AMS/MSC91 classification: 65Y20, 68Q22, 65N50

1 The partition problem

Finite-Element, Finite-Volume and Finite-Difference methods for the solution of partial differential equations are based on meshes. The solution is represented by degrees of freedoms attached to certain locations on the mesh. Numerical algorithms operate on these degrees of freedom during steps like the assembly of a linear equation system or the solution of an equation system. A natural way of porting algorithms to a parallel computer is the data distribution approach. The mesh with attached degrees of freedom is decomposed into several partitions and mapped to the processors of the parallel computer. Accordingly also the operations on the data are partitioned. Goals of a partitioning scheme are load-balancing and little communication between the processors. Sometimes also singly-connected partitions are required. If the partitions are determined during run-time, furthermore a fast partitioning scheme itself is sought. This is e.g. the case within adaptive mesh refinement of a PDE solver.

The partitioning problem in general is NP-hard [18]. There are many heuristics based on graph connectivity or geometric properties to address this problem [2,6,12,13,19]. In practice fast heuristics are known. However, there is not much known about general quality of these methods. In contrary there exist examples, where single heuristics give really bad results.

In this paper we analyse a specific geometry based heuristic based on space-filling curves. It is cheap and helps to simplify the implementation of parallel algorithms [9,15,16,17,20,23]. We are interested in bounds for the quality of the partitions. This will lead us to general estimates on the parallel performance of advanced numerical algorithms on these partitions.

*Institut für Angewandte Mathematik, Universität Bonn, Wegelerstr. 6, D-53115 Bonn, Germany, e-mail zumbusch@iam.uni-bonn.de, URL http://wissrech.iam.uni-bonn.de/people/zumbusch.html

2 Space-filling curves

First we have to define curves. The term curve shall denote the image of a continuous mapping of the unit interval to the $\mathbb{R}^d$. Mathematically, a curve is space-filling if and only if the image of the mapping does have a classical positive d-dimensional measure. The curve fills up a whole domain. For reasons of simplicity we restrict our attention to simple domains. We are interested in a mapping

$$f \;:\; [0,1] =: I \;\mapsto \Omega \subset \mathbb{R}^d, \qquad f \text{ continuous and surjective} \tag{1}$$

There are classical curves like the Hilbert-, the Peano- and the Lebesgue-curve, see [21]. However, we will also construct special space-filling curves on an unstructured mesh.

The space-filling curve can also be used for the inverse mapping f from a domain $\Omega \subset \mathbb{R}^d$ to the unit interval I. This means that we can map geometric entities in $\mathbb{R}^d$ to the one dimensional interval such as elements or nodes. Entities, which are neighbours on the interval, are also neighbours in the volume $\mathbb{R}^d$. Unfortunately the reverse cannot be true and neighbours in the volume may be separated through the mapping.

However, we can solve the resulting one-dimensional partition problem: We cut the interval I into disjoint sub-intervals I_j of equal workload with $\bigcup_j I_j = I$. This gives perfect load-balance and small separators between the partitions. The partition $f(I_j)$ of the domain Ω induced by the space-filling curve with $\bigcup_j f(I_j) \supset \Omega$ also gives perfect load-balance. However, the separators $\partial f(I_j) \setminus \partial\Omega$ are larger than the optimal separators in general as we will see.

3 Quality of a partition

We use a basic performance model for a distributed memory computer. The execution time of a program consists of computing time, which is proportional to the number of operations on a processor, and of communication time. Communication between the processors is done with message passing through some network and requires time linear in the size of data $t = t_{\text{startup}} + n * t_{\text{bandwidth}}$.

We consider $O(n)$ algorithms linear in the size of data n, e.g. FEM matrix assembly for n finite elements, sparse matrix multiply or components of a multigrid algorithm such as a grid transfer or smoother, see [2,10]. The parallel computing time is $C_1 \cdot n/p$ for a partition of n data onto p processors. We call $v := n/p$ the *volume*. The runtime depends on the communication time. The data to be transferred is proportional to the separator or surface s_j of the partition $s_j := \partial f(I_j) \setminus \partial\Omega$.

$$t = C_1 \frac{n}{p} + C_2 (t_{\text{startup}} + s * t_{\text{bandwidth}}) \tag{2}$$

This model suggest that we have to minimise the surface to volume ratio s/v of the partition for a high parallel efficiency of

$$\text{efficiency} = 1/\big(1 + \frac{C_2}{C_1}(\frac{1}{v} t_{\text{startup}} + \frac{s}{v} * t_{\text{bandwidth}})\big) \;. \tag{3}$$

While the lowest continuous surface to volume ratio is obtained for the sphere by $s = \sqrt[d]{2d^{d-1}\frac{\pi^{d/2}}{\Gamma(d/2)}}v^{(d-1)/d}$, we usually deal with partitions aligned with the mesh. Hence the cube with $s = 2dv^{(d-1)/d}$ is of interest. In general we regard estimates of type

$$s \ \leq C_{\text{part}} \cdot v^{(d-1)/d} \tag{4}$$

with low constants C_{part} as optimal.

4 Estimates for space-filling curves

The estimate for the locality of a discrete space-filling curve F we will use with $F : [1, \ldots , k^d] \mapsto [1, \ldots , k]^d$ is of type

$$\|F(x) - F(y)\|_2 \leq C \sqrt[d]{|x - y|} \ . \tag{5}$$

Gotsman and Lindenbaum [8] give an upper bound $C = (d + 3)^{d/2}2^d$ for the Hilbert curve and tighter bounds for $C = 6\frac{2}{3}$ for $d = 2$ and $C = 23$ for $d = 3$, which has been improved by [1]. Analogous estimates have been derived for the Hilbert curve [22] and the Peano curve [7]. It turns out that a similar curve, called H-index gives even better constants, see [5,14].

Lemma 1. *Given a connected discrete space-filling curve F on a domain $[1, \ldots , k]^d$ and a partition $F([j, \ldots , j + v - 1])$ of v nodes, the surface s of the partition is bounded by eqn. 4. The constant C_{part} depends on the curve.*

Proof. is based on eqn. 5 and the connectedness of the partition. It is sufficient to consider s of the bounding box.

This lemma does not hold for curves of Lebesgue also called *bit-interleaving* [3], because the discrete partitions tend to be disconnected. However, we generalise the situation to unstructured and adaptively refined meshes by the following construction: We create an enumeration of a mesh by some heuristic in order to obtain a 'local' discrete space-filling curve. Then we do mesh refinement by some geometric refinement rules, see [2,4]. Each coarse element E_j is substituted by several smaller $E_{j,k}$ elements. The enumeration is changed such that it cycles through these new elements $E_{j,k}$ right after the elements E_{j-1} or $E_{j-1,k}$. This leads in the limit to a continuous space-filling curve, see [11,15,20]. Alternatively a standard, continuous space-filling curve can be super-imposed onto the grid, see [9,17].

Corollary 2. *Estimate 4 also holds for a space-filling curve partitioning of a (quasi-) uniform mesh by superposition of f or mesh dependent construction of f.*

Estimate 3 combined with corollary 2 gives a speedup for large problems of

$$\text{efficiency} = 1/\big(1 + \frac{C_2 C_{\text{part}} t_{\text{bandwidth}}}{C_1} \cdot \frac{p}{n^{1/d}}\big) \ . \tag{6}$$

This implies optimal parallel efficiency for very large problems, $n \to \infty$. Estimate 6 holds for a code for the solution of partial differential equations in the steps of setting up an equation system, a single matrix multiply, a fixed number of Krylov iterations. Furthermore, using the same space-filling curve an all grid levels, this also holds for an additive multigrid implementation and for standard multigrid if we neglect terms $\log n \cdot t_{\text{startup}}$ proportional to the number of grid levels. For the scalability of a gobal PDE solver an $O(n)$ multigrid solver is essential. Solvers with higher than linear complexity may scale in p like eqn. 6 but scale completely different in n.

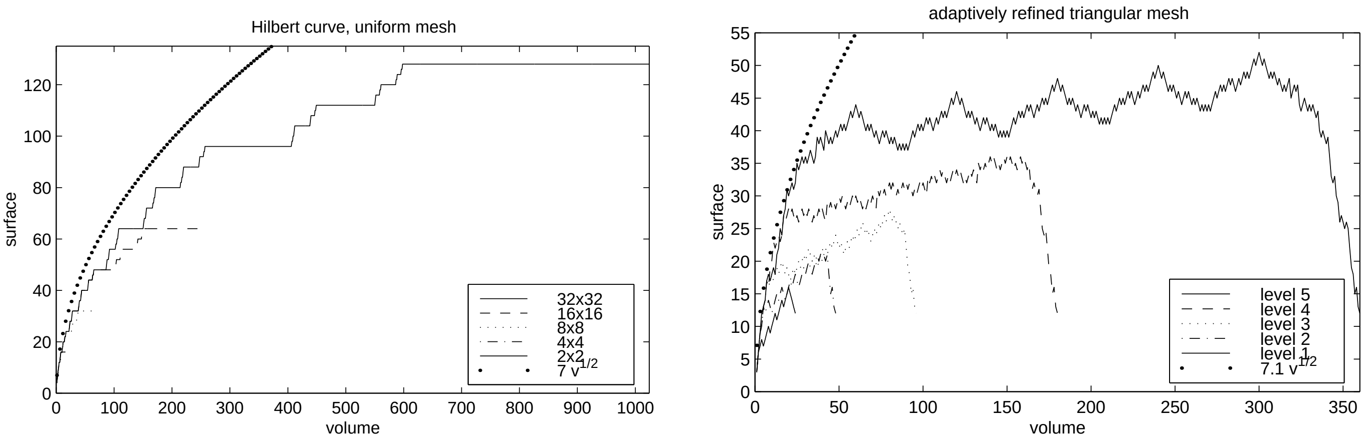

Figure 1: Locality of partitions defined by a space-filling curve. Hilbert curve on an uniform mesh (left) and an unstructured mesh with adaptive mesh refinement (right).

5 Numerical experiments

The proof of lemma 1 only gives a crude estimate on the constant in eqn. 4. Hence we look at two examples for two-dimensional partitions. In figure 1 the maximum surfaces s to different volumes v are given. We consider a uniform square $[0, 2^k]^2$ (counting the complete boundary) and a triangulation (counting the interior boundary only). The triangulation starts with a hexagon and angles of $\pi/3$ and is refined adaptively. The triangulation is shown in figure 2 left. The different graphs in figure 1 show the ratios for different grid levels. The surface of small partitions comes close to the expected $\sqrt{n}$ behaviour while larger partitions have a limited boundary. $\partial\Omega$ is a natural limit.

Lemma 1 did not deal with adaptive mesh refinement. Although, moderate refinement seems to give similar estimates. However, very strong refinement with an arithmetic progression of nodes during refinement shows a different picture. In this example, figure 2 right, s is proportional to v. This behaviour limits the usefulness of the partition method. This 'counter' example is related to examples where other heuristics like spectral bisection [19] also fail to perform well.

6 Sparse grids

Space-filling curve partitions can also be used for the parallelisation of adaptive sparse grid implementations, see [25]. A certain choice of tensor products of (pre-) wavelet basis functions can give approximations with a low number of degrees freedom of the order

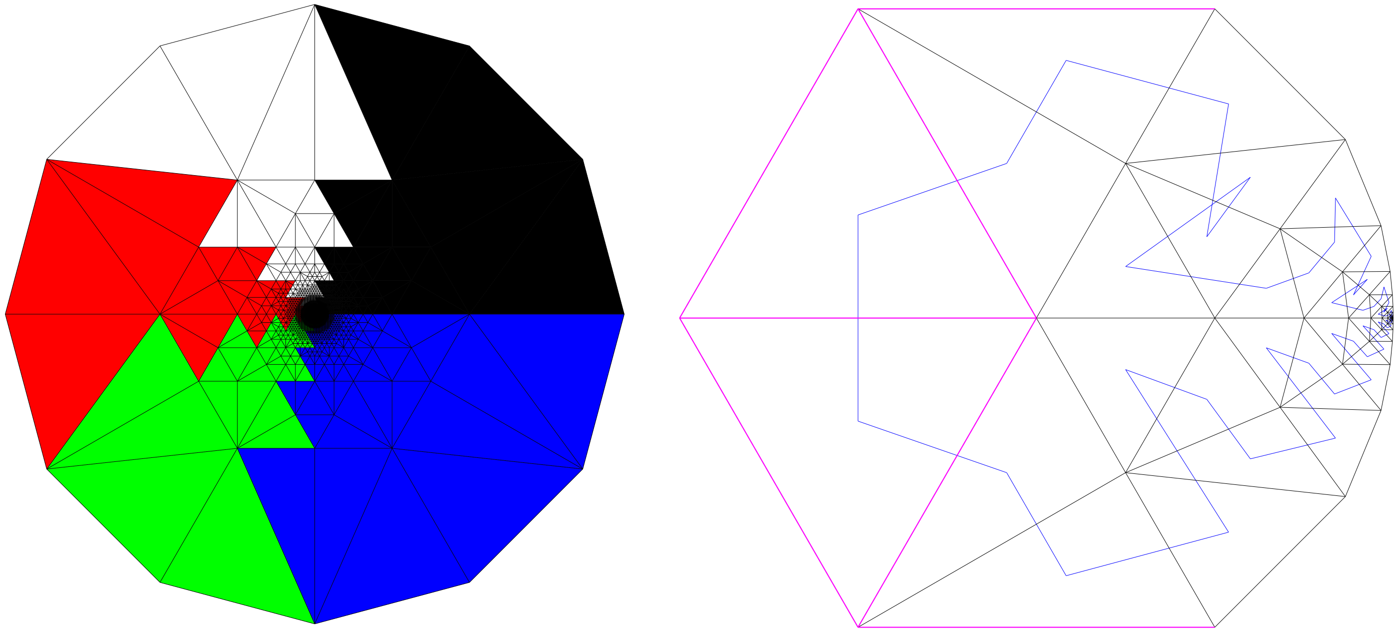

Figure 2: Adaptive mesh refinement. Partitions defined by a space-filling curve (left) and a counter example for a non-local partition (right).

$v = O(n \log^{d-1} n)$ for a spatial resolution of $1/n$, see [24]. The discretization of PDEs on such sparse grids links geometrical nodes on different scales and at different distances. The surface of a rectangular shaped partition is of order $s = O(n \log^{d-2} n)$ which is rather large compared to v. Experimentally space-filling curves and other graph partitions heuristics give partition surfaces of similar size.

$$s \le C_{\text{part}} \frac{v}{\log n} \qquad \text{with } v = O(n \log^{d-1} n) \tag{7}$$

$$\text{efficiency} = 1/\big(1 + C \frac{p}{\log n}\big) \tag{8}$$

We obtain scalability of wavelet algorithms on sparse grids. However, the parallel efficiency grows far slower in the problem size as for standard discretizations which scale excellently, compare eqns. 6 and 8.

7 References

1 Alber, J., Niedermeier, R.: Hilbert indexings in arbitrary dimensions. Theory of computing systems, to appear.

2 Bastian, P.: Parallele Adaptive Mehrgitterverfahren. Teubner (1996).

3 Beichl, I., Sullivan, F.: Interleave in Peace, or Interleave in Pieces. IEEE Computational Science Engineering **2**, (1998), 92–96

4 Bey, J.: Tetrahedral grid refinement. Computing **55**, (1995), 355–378.

5 Chochia, G., Cole, M.: Recursive 3D mesh indexing with improved locality. in: Proc. HPCN '97, LNCS **1225**, Springer (1997).

6 Dubinski, J.: A parallel tree code. New Astronomy **1**, (1996), 133–147.
7 Garsia, A. M.: Combinatorial inequalities and smoothness of functions. Bull. Amer. Math. Soc. **82** (1976), 157–170.
8 Gotsman, C., Lindenbaum, M.: On the metric properties of discrete space-filling curves. IEEE Trans. on Image Processing **5** (1996), 794–797.
9 Griebel, M., Zumbusch, G.: Hash-Storage Techniques for Adaptive Multilevel Solvers and their Domain Decomposition Parallelization. in: Proc. Domain Decomposition Methods 10. Mandel, J., Farhat, C., Cai, X.-C., Contemporary Mathematics **218**, AMS (1998), 279–286.
10 Griebel, M., Zumbusch, G.: Parallel adaptive subspace correction schemes with applications to elasticity. Computer methods in applied mechanics and engineering. **184** (2000), 303–332.
11 Heber, G., Gao, G. R., Biswas, R.: Self-Avoiding Walks over Adaptive Unstructured Grids. Concurrency - Practice and Experience. **12** (2000), 85–109.
12 Karypis, G., Kumar, V.: Multilevel k-way Graph Partitioning for Irregular Graphs. J. Parallel and Distributed Comp. **48** (1998), 96–129.
13 De Keyser, J., Roose, D.: Partitioning and Mapping Adaptive Multigrid Hierarchies on Distributed Memory Computers. Report TW 166, Univ. Leuven, Dept. Computer Science (1992).
14 Niedermeier, R., Reinhardt, K., Sander, P.: Towards optimal locality in mesh-indexings. in: Proc. 11th Int. Symp. Fundamentals of Computation Theory, LNCS **1279** (1997), 364–375.
15 Oden, J. T., Patra, A., Feng, Y.: Domain Decomposition for Adaptive hp Finite Element Methods. in: Proc. Domain Decomposition 7, Contemporary Mathematics **180**, AMS (1994), 295–301.
16 Parashar, M., Browne, J. C.: On partitioning dynamic adaptive grid hierarchies. in: Proc. 29th Annual Hawai Int. Conf. System Sciences (1996), 604–613.
17 Pilkington, J. R., Baden, S. B.: Partitioning with Spacefilling Curves. Report CS94-349, UCSD, Dept. Computer Science (1994).
18 Pothen, A.: Graph Partitioning Algorithms with Applications to Scientific Computing. in: Parallel Numerical Algorithms, Keyes, D. E., Sameh, A., Venkatakrishnan, V. (eds.), Kluwer (1997), 323–368.
19 Pothen, A., Simon, H., Liou, K.-P.: Partitioning Sparse Matrices with Eigenvectors of Graphs, SIAM J. Matrix Analysis and Applications **11** (1990), 430–452.
20 Roberts, S., Kalyanasundaram, S., Cardew-Hall, M., Clarke, W.: A Key Based Parallel Adaptive Refinement Technique for Finite Element Methods, in: Proc. CTAC97, Noye, B. J., Teubner, M. D., Gill, A. W. (eds.), World Scientific (1998)
21 Sagan, H.: Space-Filling Curves, Springer (1994).
22 Stout, Q.F.: Topological matching. Proc. 15th ACM Symp. on Theory of Computing (1983), 24–31.
23 Warren, M., Salmon, J.: A portable parallel particle program. Comput. Phys. Comm. **87**, (1995), 266–290.
24 Zenger, C.: Sparse Grids. in: Proc. 6th GAMM Seminar, Hackbusch, W. (eds.), Vieweg (1991).
25 Zumbusch, G.: A Sparse Grid PDE Solver. in: Advances in Software Tools for Scientific Computing, Langtangen, H. P., Bruaset, A. M., Quak, E. (eds.), LNCSE **10**, Springer (2000), 133–177.