NEW PARALLEL APPROACHES FOR FAST MULTIPOLE SOLVERS

J. Peeters, J. Fostier, F. Olyslager, D. De Zutter †

† Ghent University, Belgium,
Department of Information Technology (INTEC),
Sint Pietersnieuwstraat 41, B-9000 Ghent,
E-mail: joris.peeters@intec.ugent.be

Keywords: Electromagnetic scattering, Method of Moments, Parallel MLFMA

Abstract

In [4] we presented a new asynchronous kernel-independent parallel Multilevel Fast Multipole Algorithm (MLFMA), applied to full-wave electromagnetic simulations in two dimensions and in the time-harmonic domain. In this contribution, we report the progress in the extension of this work to three dimensions. An asynchronous algorithm becomes indispensable when more complicated geometries are considered, with multiple dielectrics and conductors, possibly attached to or embedded into each other. Previous efforts in literature were largely focused on single, perfect electric conducting (PEC) objects. The underlying Method of Moments (MoM) makes use of the Rao-Wilton-Glisson (RWG) basis functions and the conventional choices for the boundary integral equations (BIE). The MLFMA algorithm is applied to a variety of three dimensional simulations, demonstrating its versatility.

1 Introduction

The MoM is a powerful method for full-wave electromagnetic simulations. The restriction to homogeneous objects allows for the use of Boundary Integral Equations (BIE) in terms of the unknown induced (equivalent) surface currents. The surface current is discretized by means of the Rao-Wilton-Glisson (RWG) basis functions. A linear system is obtained by testing the BIE with the same number of test functions as basis functions. This linear system can be solved directly, leading to an $O(N^3)$ time complexity, or iteratively, leading to $O(N_{it}N^2)$ where the number of iterations is expected to have a sub-linear dependence on the number of unknowns. The memory complexity is generally $O(N^2)$, because the impedance matrix must be stored. In the past, much effort has been invested in improving the MoM, to allow the simulation of ever larger problems. The Fast Multipole Method (FMM) and its extension, the Multilevel Fast Multipole Algorithm (MLFMA), succeeded in dramatically reducing the time complexity per iteration to $O(N^{1.5})$ and $O(N\log N)$ respectively.

Further reduction of the calculation time can be achieved by solving the problem on a cluster of processors, connected through a network, which is called distributed parallel computing. The parallelization of the MLFMA proves difficult, due to intense communication requirements between the nodes, hence requiring fast and expensive intercommunication networks. Most of the previous efforts focused essentially on the calculation of the RCS of very large three dimensional perfect electric conducting (PEC) objects such as aircraft, etc. [6, 11].

The workload is divided by partitioning the MLFMA tree among the different nodes in the parallel architecture. A synchronous approach, i.e. an algorithm where each node performs the same type of operations at a given point in time, is only suitable for scattering at a single PEC or dielectric object. The extension to a parallel MLFMA that is capable of handling multiple dielectric and conducting objects gives rise to a more complicated asynchronous algorithm, i.e. a parallel algorithm where different nodes can perform different types of operations at a given point in time. This avoids synchronization between consecutive levels of the MLFMA trees and also eliminates or reduces idle time in the nodes. Furthermore, a better spreading of the communication through time alleviates the need for a fast intercommunication network, making the algorithm suitable for non-dedicated clusters that are connected through a cheap network, such as the common Gigabit Ethernet standard.

Prior to the iterative process itself, most of the calculation time in the initialization phase is spent while calculating the near interaction elements. In many cases however, the geometries contain a lot of (local) symmetry, resulting in only a small number of unique interaction elements. By utilizing so-called Splay Trees as a cache for near interaction elements, these symmetries can be recognized and the recalculation of interactions is avoided. This results in a drastic reduction of the setup time.

This paper is organized as follows: first an outline of the MoM and the MLFMA is presented in sections 2 and 3, followed by a short overview of the asynchronous parallel methodology in section 4. The Splay Tree method is addressed in section 5. Finally, some numerical results are presented in section 6, which demonstrate the performance in both speed and accuracy.

2 Method of Moments

We assume that the reader is familiar with the general application of the MoM to three dimensional electromagnetic scattering problems. This section is dedicated to specifying the choices and possibilities within our implementation.

First of all, the geometry must be piecewise homogeneous, consisting of dielectric or PEC objects. These objects are allowed to touch and can also be embedded into each other. This restriction allows for a formulation of the problem in terms of (equivalent) surface currents, which essentially means that only the interfaces between media must be discretized and not their interior. The surface currents (electric and/or magnetic) are discretized using the Rao-Wilton-Glisson basis functions [8], which extend over two triangles and hence require triangulation of the interface between each medium. The handling of the junctions is based on [13], and allows for the testing functions to be RWGs as well, thus maintaining a Galerkin scheme.

The choice of Boundary Integral Equation (BIE) has a significant influence on the performance of the MoM, both in terms of speed and accuracy. For every domain, two independent BIE exist: the Electric Field Integral Equation (EFIE) and the Magnetic Field Integral Equation (MFIE), which essentially demand the fields to satisfy the electric and magnetic field boundary condition at the interfaces.

Each can be split in two by using either the tangential part (tEFIE and tMFIE) or the cross product of the normal on the surface with the tangential part (nEFIE and nMFIE). Any linear combination of these four also is a valid BIE. The choice of linear combination influences the accuracy and speed (number of iterations), generally in an opposite way. Additionally, not every linear combination leads to correct results in the presence of internal resonances, which places some important restrictions, particularly for large-sized objects. Also, when testing any of the aforementioned four BIE with an RWG function, either the electric current or magnetic current is well tested, but never both at the same time. For an accurate solution for both electric and magnetic currents (as is necessary for dielectric objects) it is obvious that at least two BIEs will have to be used. Finally, the MFIE is only valid for closed domains and can not be used in the case of open PEC surfaces. Despite these restrictions there are still an infinite number of possibilities, from which the following combinations are generally picked: tEFIE for open PEC surfaces, tEFIE or tCFIE (Combined Field Integral Equation) for closed PEC surfaces and CTF (Combined Tangential Field) [12] for dielectric interfaces. These choices generally maximize the accuracy. A lower number of iterations can generally be achieved (for the closed domains) by other combinations, but at the cost of a decrease in accuracy, in particular for (but not limited to) geometries that contain sharp features [2], which is less optimal for our current purposes.

In principle a variety of iterative solvers is at our disposal, but we generally use the Transpose-Free Quasi-Minimal Residual (TFQMR) and Bi-conjugate gradient (Bi-GCSTAB) method, with (optionally) a block-diagonal preconditioner with variable block size.

3 MLFMA

We use a high frequent MLFMA algorithm to speed up the MoM. For an introduction to the MLFMA and its applications, we refer the reader to [1]. The required interpolations and anterpolations of the radiation patterns are handled entirely through FFTs [9], so the sampling in k-space is uniform for both the theta and phi angles. Each (sufficiently large) homogeneous domain leads to a tree. It has been shown that both the time and memory complexity of this method are O(NlogN). Note that, due to the low frequency breakdown, this method can only be used to treat high frequent simulations.

4 Asynchronous Parallelization

Traditional implementations of a parallel MLFMA are essentially synchronous. They are characterized by alternating phases of calculation and communication, and synchronization occurs between the participating nodes at the consecutive levels of the tree. This synchronous approach works well for single objects (either PEC or dielectric), because each processor has approximately the same amount of work in each phase and delay times (when a processor is inactive) can generally be minimized.

When multiple dielectric objects are considered -each one having its own MLFMA tree- a possible approach is to distribute each tree about equally amongst the different nodes, hence using local space filling curves for each homogeneous domain.

This approach however, does not lead to a maximized data locality and will therefore lead to increased communication costs. Also, the distribution of small trees among many nodes is generally inefficient and will potentially lead to a bad load balancing and hence longer delay times. This is especially the case when many smaller dielectric objects are involved in the geometry.

The key ideas of a new asynchronous approach were introduced in [4] and here we restrict ourselves to a short summary of the method.

The aforementioned difficulties with the synchronous algorithm are alleviated by using a global space filling curve for the complete geometry. This means that large trees will be distributed among many nodes while smaller trees are located on just a few nodes, or possibly even on a single one. This obviously implies that the trees can no longer be processed sequentially, as the work on a certain tree would stall calculations on the nodes that do not contain any part of this tree. Asynchronous treatment would allow nodes to perform calculations for any tree in a flexible way, i.e. a node can switch from working on one tree to another and minimize the time it spends idling. As soon as a node has completed some data that is required by another node, it will indicate that the data is now available. By using the so-called intermediate communication routines that are available within the Message Passing Interface (MPI) [5], it is not necessary for communicating nodes to synchronize and communications and calculations can overlap. This means that communication between nodes is spread through time instead of occurring in bursts, which results in a more efficient use of the available bandwidth, making it useful for a slower, but cheaper Gigabit Ethernet network interconnect.

One of the difficulties with asynchronous methods are the order relations that must be fulfilled, leading to a more complex implementation. For example, the aggregations to a higher level can only occur when the radiation patterns for lower levels are available. These radiation patterns may be local to another node, which will send them as soon as they are calculated. To ensure minimal waiting times, a priority queue is used to schedule the workload on each node. When possible, a node will perform calculations that are required by other nodes. Near interactions, which are not needed by any other node, will be treated only when nothing else can be done (either because everything has been calculated already or because the node is still waiting for other data to arrive). The priority queue lists the tasks in order of priority and a node will pick the most urgent work package. In the event that the priority queue becomes empty before all calculations are finished, nodes are said to be *starving for data.* A heuristic set of rules was developed for generating the priority queue, in such way that data to other processors will be sent as soon as possible and an ongoing flow of calculations is ensured.

5 Splay Trees

During the set-up stage, the near interaction elements between each pair of RWG functions associated with triangles that are geometrically near to each other need to be calculated. Each element requires two numerical integration schemes (basis and test function) and is hence a costly operation.

A triangular mesh often contains a lot of symmetry, for instance when a plate, a sphere or a cuboid is simulated. More complex geometries may also contain symmetry, albeit of a local nature. It would be very desirable if only the unique pairs of these interactions were calculated, while not spending too much time and memory to extract these symmetries. An efficient method to handle this is the usage of Splay Trees [10], which are capable of rapid storage and retrieval of data. A Splay Tree is a binary search tree that uses a heuristic to avoid becoming unbalanced. As opposed to other balanced binary search trees, the Splay Tree requires no additional memory to keep track of its balancing. Furthermore, frequently accessed elements are stored close to the root making the Splay Tree a good choice for implementing caches. A Splay Tree has an amortized complexity of O(logN) for its elementary search and store operations, with N the number of elements stored in the tree.

An interacting pair of triangles is identified by a number of keys which represent the geometrical degrees of freedom between their relative position and the medium through which they interact. When calculating the basic interactions between these triangles, the keys are fed to the splay tree, which will then return the interactions if they were calculated previously and otherwise calculate and store them for possible future use. When there is no symmetry in the structure the added cost is very limited, while the gains are formidable when the symmetry is high. For more details regarding the symmetry extraction using Splay Trees, see [7].

6 Results

The developed asynchronous method (Nexus) is independent of the kernel and the dimensionality of the problem. It is therefore suited for both acoustic and electromagnetic problems in the time harmonic domain, in two or three dimensions, and for both the Helmholtz (high frequency) and Laplace (low frequency) equations. In [3, 4], the algorithm was linked to a 2D TM MoM solver (Nero), showing its huge potential. This Nero solver is presented to the community as an open software package and can be downloaded free of charge at http://openfmm.sf.net. In this section we will study the performance of the Nexus library when linked to our general purpose 3D MoM solver (Cassandra).

All simulations were performed on a cluster consisting of 5 machines with 4 cores each (two dual core AMD Opteron 270 processors) with 2 gigabytes of memory per core. The machines were interconnected through a Gigabit Ethernet switch. The message passing interface (MPI) was used as a communication library. The popular open implementation LAM-MPI was chosen for this task.

We first verify the 3D MoM code by comparing the simulated bistatic RCS of a 6 lambda dielectric sphere (relative permittivity = 4) with the analytical solution obtained by the Mie-series. The results are shown in Figure 1 and are clearly in excellent agreement.

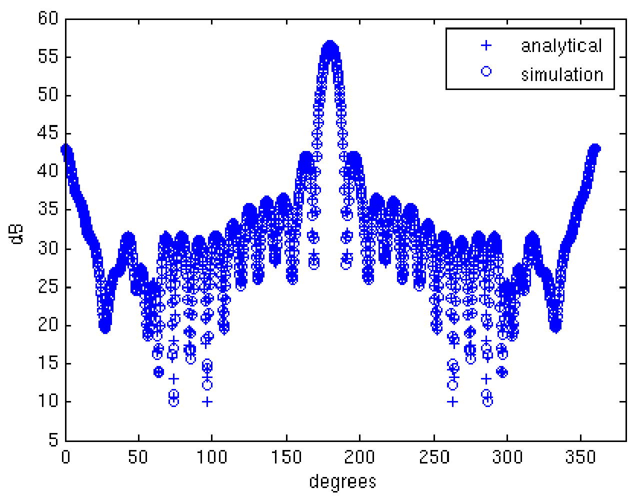


Figure 1: The analytical and simulated RCS of a sphere with a radius of 6 (outer) wavelenghts and a relative permittivity of 4.

A next verification focuses on the near field and the implementation of the junctions. The geometry has a cubic shape and consists of two dielectrics and a PEC plate and is shown in Figure 2. We are looking at the total field along the dashed line, straight through the center of the structure.

Figure 3a and 3b respectively show the results for an x-directed z-polarized and a z-directed x-polarized plane wave with wave number k=1. In both cases the total field along the polarization is shown. Therefore the first one is expected to be continuous (being purely tangential) and the second one should have the correct discontinuities (equal to the ratio of the permittivities at both sides of the interface) at the surfaces (being purely normal). Both results are in perfect accordance with this. The first figure also shows the tangential field becoming zero at the surface of the PEC plate.

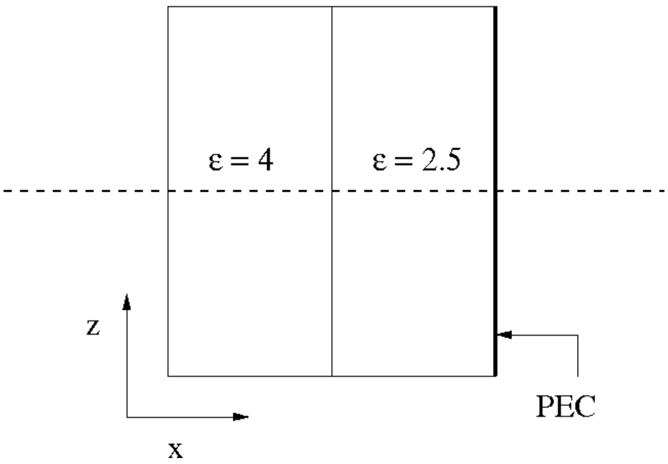


Figure 2: Cubic structure of two layers and a PEC plate. Relative permittivities are indicated.

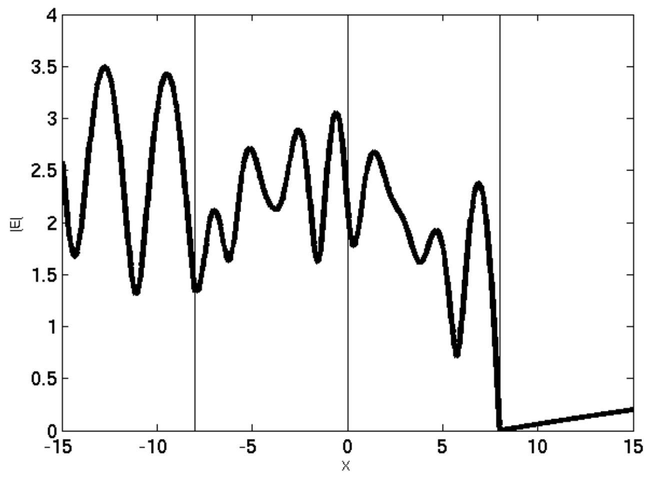


Figure 3a: The amplitude of the total field along the polarization direction of the incoming wave for an x-directed z-polarized wave. The vertical lines indicate medium interfaces.

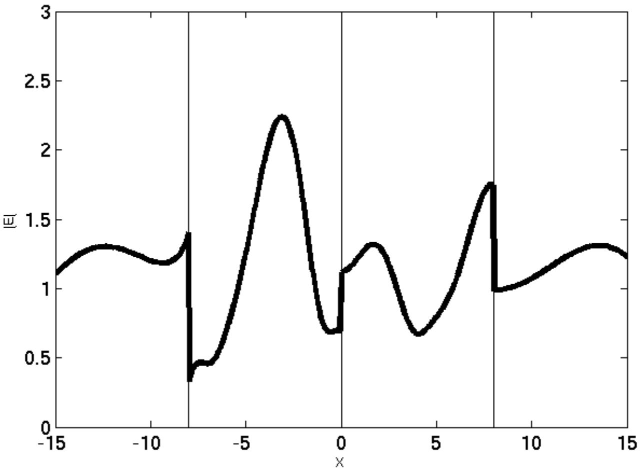


Figure 3b: The amplitude of the total field along the polarization direction of the incoming wave for an z-directed x-polarized wave wave. The vertical lines indicate medium interfaces.

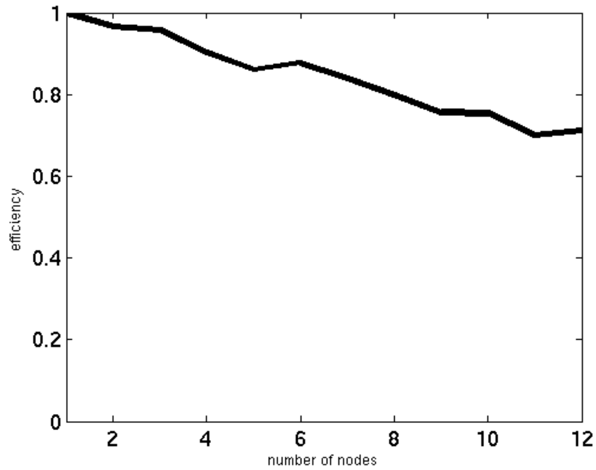


Figure 4: Parallel efficiency in terms of the number of participating nodes.

We now consider a 50m x 50m x 50m dielectric cube for k=1 and a relative permittivity of 2 and study the parallel efficiency when increasing the number of nodes, shown in Figure 4. With 12 nodes the efficiency is still slightly higher than 70%, which is acceptable for this kind of non-dedicated parallel machine. The number of unknowns is this example was approximately 450 000, so it could be run on a single core. Larger problems tend to have a higher efficiency but can no longer run on a single core, which makes it difficult to estimate the parallel efficiency. Also note that more complex geometries with different dielectric objects and different permittivities require more complex load balancing schemes. This will be the work of future research.

As a final example we show the application of dielectric mirrors. These consist of layers of dielectrics, each a quarter of a wavelength in thickness. These are imperfect mirrors which can be used, for instance, as the partially transmitting end of a laser cavity. In this example we again use k=1, and simulate three layers of 80m x 80m in size. The two outer layers are ZnS (n=2.32) and the inner layer consists of MgF_2 (n=1.38). The structure is illuminated with a plane wave and the field along the direction of polarization is shown in Figure 5. Note that the structure clearly functions as a partial mirror: part of the wave is reflected and part of the wave is transmitted.

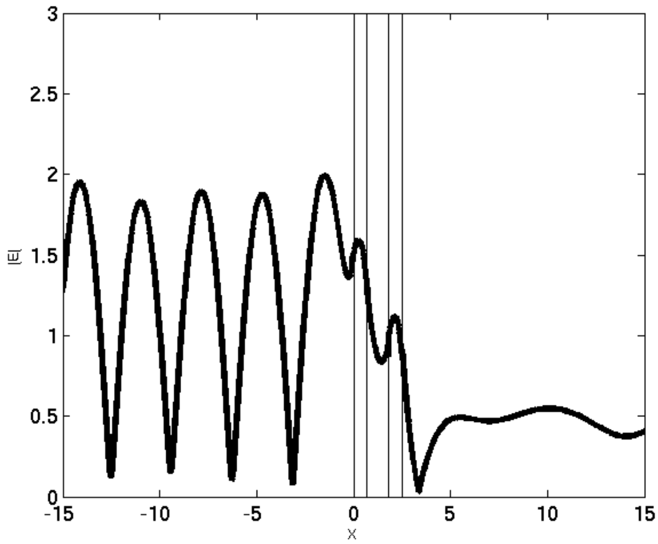


Figure 5: The total field along the polarization direction, along a line through a dielectric mirror with three layers. The vertical lines indicate the location of medium interfaces.

7 Conclusion

We have linked the asynchronous parallel MLFMA (Nexus) with a three dimensional MoM solver (Cassandra), capable of simulating generic 3d geometries with multiple dielectric and conducting objects. Embedded objects and junctions are fully supported. The use of Splay Trees for extracting symmetry was explained. We have demonstrated the accuracy of the solutions for a number of examples and demonstrated the parallel performance.

Acknowledgements

The work of J. Fostier was supported by a doctoral grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

[1] W. C. Chew, J. Jin, E. Michielssen, J. Song. "Fast and Efficient Algorithms in Computational Electromagnetics", *Artech House, Boston,* (2001).

[2] Ö. Ergül, L. Gürel, "Improving the accuracy of the magnetic field integral equation with the linear-linear basis functions", *Radio Science*, **volume 41**, (4), RS4004, (2005).

[3] J. Fostier, F. Olyslager. "A GRID Computer Implementation of the Multilevel Fast Multipole Algorithm for Full-Wave Analysis of Optical Devices", *accepted for publication in the IEICE Trans. Comm.*, sep., (2007).

[4] J. Fostier, F. Olyslager. "An asynchronous parallel multilevel fast multipole algorithm", *proc. IEEE Antennas Propag. Symp. (AP-S)* , pp. 3424-3427, (2007).

[5] W. Gropp, et. al. "Using MPI: Portable Parallel Programming with the Message-Passing Interface", *MIT Press, Cambridge*, (1993).

[6] L. Gürel, Ö. Ergül. "Fast and accurate solutions of extremely large integral-equation problems discretized with tens of million of unknowns", *Electronic Letters,* **volume 43***,* No. 9, pp. 499—500, *(2003).*

[7] F. Olyslager, J. Fostier, J. Peeters, D. De Zutter. "Fast and Accurate Evaluation of Enclosures With the Method of Moments by Using Splay Trees", *accepted for presentation at the 18th Int. Zurich Symp. on EMC*, Munich (2007).

[8] S. Rao, D. Wilton, A. Glisson. "Electromagnetic scattering by surfaces of arbitrary shape", *IEEE Trans. Antennas Propag.*, **volume 30**, (3) pp. 409—418, (1982).

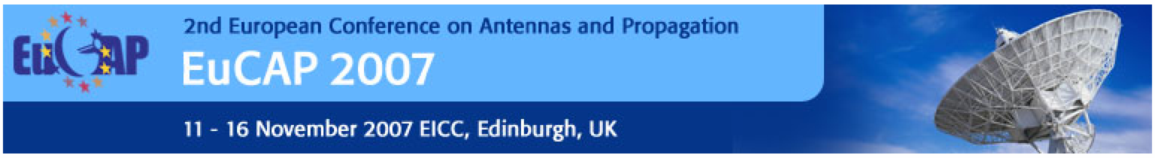
[9] J. Sarvas. "Performing Interpolation and Anterpolation Entirely by Fast Fourier Transform in the 3-D Multilevel Fast Multipole Algorithm", *SIAM J. Numer. Anal.*, **volume 41**, pp. 2180—2196, (2003).

[10] D. D. Sleator, R. E. Tarjan. "Self-adjusting Binary Search Trees", *JACM*, **volume 32**, (3), pp. 652—686, (1985).

[11] S. Velamparambil, W.C. Chew. "Analysis and Performance of a Distributed Memory Multilevel Fast Multipole Algorithm", *IEEE Trans. Antennas Propag.*, **volume 53**, (8), pp. 2719--2727, (2005).

[12] P. Yla-Oijala, M. Taskinen, S. Jarvenpaa. "Surface integral equation formulations for solving electromagnetic scattering problems with iterative methods", *Radio Science*, **volume 40**, (6), RS6002, (2005).

[13] P. Yla-Oijala, M. Taskinen, J. Sarvas. "Surface integral equation method for general composite metallic and dielectric structures with junctions", *Progress In Electromagnetic Research*, **volume 52**, pp. 81-108, (2005).



Welcome to

The Second European Conference on Antennas and Propagation EuCAP 2007

11 - 16 November 2007, The EICC, Edinburgh, UK

Organised by the Institution of Engineering and Technology on behalf of EurAAP AISBL

Sponsored by

ISBN 978-0-86341-842-6