



Studenckie laboratorium obliczeniowe

dr inż. Jacek Olszewski

dr inż. Rafał Orlik

dr inż. Grzegorz Pawlik

mgr inż. Karol Tarnowski

pod redakcją

dr. hab. inż. Włodzimierza Salejdy, prof. nadzw. PWr

E-skrypt opracowany w ramach projektu pt. „Wzrost liczby absolwentów w Politechnice Wrocławskiej na kierunkach o kluczowym znaczeniu dla gospodarki opartej na wiedzy” nr UDA-POKL.04.01.02-00-065/09-01



Recenzent: dr. inż. Paweł Scharoch

Redaktor serii: dr. hab. inż. Włodzimierza Salejdy, prof. nadzw. PWr

©Copyright by Politechnika Wroclawska
OFICYNA WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

ISBN 978-83-7493-607-1



Spis treści

1. Podstawy symulacji komputerowych metodą dynamiki molekularnej	19
1.1. Podstawowe algorytmy	19
1.1.1. Całkowanie równań ruchu Newtona	19
1.1.2. Oddziaływania krótkozasięgowe	22
1.1.3. Periodyczne warunki brzegowe	22
1.1.4. Zasada minimalnego obrazu	25
1.2. Wielkości zredukowane	26
1.2.1. Położenie, pole powierzchni oraz gęstość	26
1.2.2. Energia potencjalna	27
1.2.3. Czas	27
1.2.4. Energia kinetyczna	28
1.2.5. Temperatura	28
1.2.6. Ciśnienie	29
1.3. Dwuwymiarowy układ Lennarda-Jonesa	31
1.3.1. Potencjał Lennarda-Jonesa	31
1.3.2. Bezwymiarowe równanie ruchu Newtona	32
1.4. Symulacje w układzie NVE	33
1.4.1. Warunki początkowe	33
1.4.2. Skalowanie prędkości	34
1.4.3. Modelowy program dynamiki molekularnej	34
1.5. Zakończenie	35
2. Metoda Monte Carlo (MC)	41
2.1. Statystyczny opis układu	41
2.2. Metoda Monte Carlo - koncepcje	42
2.2.1. Algorytm Metropolisa	42
2.3. Dwuwymiarowy model Isinga	43
2.3.1. Opis modelu	43
2.3.2. Symulacja	45
2.3.3. Obliczanie wartości średnich	47
2.3.4. Zadania do wykonania w ramach laboratorium	51
3. Rozwiązywanie równań Maxwella metodą FDTD	55
3.1. Wstęp	55
3.2. Równania Maxwella	56
3.2.1. Równania Maxwella – trzy wymiary	57
3.2.2. Równania Maxwella – dwa wymiary	58
3.2.3. Równania Maxwella – jeden wymiar	58
3.3. Dyskretyzacja pochodnych na siatce Yee	58



3.3.1.	Siatka Yee w jednym wymiarze	59
3.4.	Źródła promieniowania	62
3.4.1.	Źródła twarde	62
3.4.2.	Źródła miękkie	62
3.5.	Pierwszy program	63
3.6.	Warunki brzegowe	66
3.7.	Propagacja w ośrodku dielektrycznym	67
3.8.	Propagacja w ośrodku absorbującym	69
4.	Modelowanie właściwości światłowodów metodą elementów skończonych	77
4.1.	Wprowadzenie	77
4.2.	Sformułowanie problemu	77
4.3.	Rozwiązanie zagadnienia brzegowego metodą MES	80
4.3.1.	Dyskretyzacja	80
4.3.2.	Aproksymacja	82
4.3.3.	Scalenie i rozwiązanie układu równań	85
4.4.	Przykładowy program	87
4.5.	Podsumowanie	93
4.6.	Zadania	94



Przedmowa

Fizyka obliczeniowa, zwana także mało precyzyjnie fizyką komputerową, jest interdyscyplinarną dziedziną fizyki powstałą w drugiej połowie XX wieku na pograniczu:

- fizyki teoretycznej [1–14],
- matematyki konkretnej i matematyki obliczeniowej [15–35, 37, 39, 40, 42–101, 103–106, 108–119, 121, 125, 126, 132]
- informatyki stosowanej, co obejmuje m.in. języki programowania (C, C#, C++, Delphi, JAVA, Perl, PHP, Python, Ruby, Turbo Pascal, Visual Basic, Visual C#, Visual C++), biblioteki programów numerycznych, środowiska programistyczne i obliczeniowe, oprogramowanie otwarte [36, 38, 41, 72, 79, 83, 102, 107, 120, 122–124, 127–131].

Począwszy od drugiej połowy XX wieku opracowane zostały w oparciu o teorie i modele fizyki klasycznej i kwantowej efektywne i stabilne algorytmy, biblioteki programów numerycznych, metody obliczeniowe i metody symulacji komputerowych procesów fizycznych [15, 16, 21, 22, 25–27, 29, 30, 32, 33, 36–41, 43, 46–50, 52–56, 59, 60, 63–65, 67, 68, 70–72, 74, 77–81, 83, 87, 88, 93–95, 97–102, 107–112, 114–121, 125, 126].

Intensywny rozwój fizyki obliczeniowej czego dobitnym wyrazem jest bardzo obszerny ale mimo to niepełny spis załączonej literatury w układzie i porządku historycznym był i jest możliwy dzięki postępowi technologii przemysłu komputerowego, które odbywa się w tempie wykładniczym zgodnie z prawami Moore’a.

Fizyka obliczeniowa zajmuje się implementacją algorytmów numerycznych do rozwiązywania zagadnień, których jakościowy oraz ilościowy opis zadają i określają teorie i modele wypracowane w poszczególnych działach fizyki. Są to m.in. fizyka jądrowa i fizyka plazmy (m.in. modelowanie funkcjonowania urządzenia do prowadzenia kontrolowanej fuzji lekkich jąder, którego skonstruowanie jest obecnie celem głównym projektu ITER (International Thermonuclear Experimental Reactor)), fizyka cząstek elementarnych (m.in. projektowanie akceleratorów cząstek elementarnych), fizyka materii skondensowanej (m.in. wyznaczanie struktury pasmowej ciał stałych oraz ich właściwości fizycznych, materii miękkiej), astrofizyka, fizyka kwantowa (m.in. modele sieciowe kwantowej chromodynamiki).

Problemy, z którymi ma do czynienia fizyka, prawie zawsze nie są rozwiązywalne analitycznie, tj. ściśle. Przykładowo równanie ruchu ciał, zadawane II zasadą dynamiki Newtona, jest równaniem różniczkowym zwyczajnym drugiego rzędu, które jest ściśle rozwiązywalne przy założeniu stałej wartości masy ciała i niezależnej od czasu siły działającej na nie. Rzeczywisty ruch ciała jest zjawiskiem o wiele bardziej skomplikowanym i w ogólnym przypadku nierozwiązywalnym analitycznie ze względu na niepewności co do wartości siły działającej na ciało w ruchu (przykładowo: ruch samochodu, ruch samochodu wyścigowego Formuły 1, ruch rakiety kosmicznej podczas startu itp.). Z podobną sytuacją mamy do czynienia w fizyce kwantowej, gdzie analityczne rozwiązanie stacjonarnego równania Schrödingera jest znane w kilku zaledwie przypadkach.



Warto wymienić w tym miejscu wybrane rodzaje zagadnień matematycznych, które są rozwiązywane metodami fizyki obliczeniowej. Są nimi m.in.: równania różniczkowe zwyczajne i cząstkowe, obliczanie wartości całek, zagadnienie własne polega na wyznaczeniu wartości i wektorów własnych macierzy lub operatorów hermitowskich, wyznaczanie potencjałów termodynamicznych układów oddziaływujących atomów lub molekuł, modelowanie rozkładu pola elektromagnetycznego w przestrzeni i materiałach. Ten e-skrypt dotyczy dwóch ostatnich problemów z wymienionych powyżej.

Oddawany do rąk Czytelniczek i Czytelników e-skrypt jest skromnym wkładem do fizyki obliczeniowej potwierdzającym jej istotną rolę i rosnące znaczenie dla rozwoju nauk podstawowych oraz współczesnych technologii. Założone cele główne, które przyświecały Autorom e-skryptu, były następujące:

1. Pozyskanie przez studentów kierunku fizyka i pokrewnych podstawowej wiedzy z zakresu fizyki i matematyki dotyczącej wybranych procedur numerycznych i algorytmów komputerowych powszechnie stosowanych obecnie w badaniach właściwości fizycznych materii skondensowanej. W szczególności dotyczy to układów oddziaływujących wielu ciał wyznaczone są charakterystyki termodynamiczne metodami dynamiki molekularnej i Monte Carlo oraz właściwości oddziaływania światła z materią modelowany jest metodą różnic skończonych w domenie czasowej (FDTD) lub elementów skończonych (MES) rozkład pola elektromagnetycznego i propagacja fal elektromagnetycznych w strukturach fotonicznych (światłowody, falowody, kryształy i światłowody fotoniczne).
2. Nabycie umiejętności: (a) samodzielnego, sprawnego i efektywnego w sensie informatyczno-komputerowym oraz zgodnego z prawami i zasadami fizyki posługiwania się opisanymi dość szczegółowo w tym skrypcie procedurami i algorytmami do jakościowych oraz ilościowych badań właściwości struktur materii skondensowanej; (b) samodzielnego pogłębiania wiedzy i umiejętności.
3. Przystwojenie kompetencji osobistych i społecznych polegających na: (a) świadomości ograniczeń i przybliżonego charakteru opisanych numerycznych metod obliczeniowych, (b) potrzebie dalszego zdobywania wiedzy i umiejętności posługiwania się bardziej zaawansowanymi metodami obliczeniowymi.

E-skrypt składa się z czterech części. Pierwszą z nich pt. Podstawy symulacji komputerowych metodą dynamiki molekularnej, opracował dr inż. Rafał Orlik. Przedstawione zostały fizyczne i matematyczne podstawy tej ważnej komputerowej techniki symulacyjnej pozwalającej wyznaczać charakterystyki dynamiczne i termodynamiczne klasycznych wielociałowych/wieloatomowych układów [13,114]. Metoda dynamiki molekularnej (DM) pozwala analizować, obserwować i śledzić ilościowo oraz jakościowo odbywający się w czasie rzeczywistym termiczny ruch zbioru N atomów/molekuł, których oddziaływania są zadane a priori i podlegają klasycznym równaniom ruchu Newtona oraz klasycznej fizyki statystycznej. Autor zwięźle prezentuje wybrane algorytmy numeryczne całkowania równań ruchu i dyskutuje ich niepewności. Przedstawia i omawia typy oddziaływań międzyatomowych oraz sposoby uwzględniania ich w symulacjach komputerowych przy nałożeniu periodycznych warunków brzegowych. Następnie Czytelnik dowiaduje się o technikach obliczeniowych wyznaczania charakterystyk termodynamicznych takich, jak energie: kinetyczna, potencjalna, całkowita oraz temperatura i ciśnienie. Opisana technologia symulacyjna jest zastosowana do dwuwymiarowego układu atomów umieszczonych w polu potencjalnym typu Lennarda-Jones'a. Tę część skryptu kończą kody programów do symulowania wybranych układów metodą DM. Sprzyja to praktycznemu opanowaniu przez studentów techniki DM.



Dr inż. Grzegorz Pawlik opracował drugą część skryptu, które nadał tytuł Metoda Monte Carlo (MC). Jest ona krótkim wprowadzeniem do techniki obliczeniowej szeroko stosowanej m.in. w fizyce fazy skondensowanej, w fizyce materii miękkiej i fizyce jądrowej [1,6,47,52,108,114,121]. Autor przedstawił podstawy fizyczne i matematyczne metody, której twórcami w latach 40. XX wieku byli John von Neumann, Stanisław Ulam i Nicholas Metropolis. Pracowali oni w Los Alamos National Laboratory przy projekcie Manhattan, zakończonym pełnym sukcesem, którym było skonstruowanie bomby atomowej (pierwszy wybuch 16 VII 1945). do komputerowego modelowania wielu różnych procesów fizycznych. Lektura tego rozdziału pozwala studentom zdobyć i utrwalić wiedzę dotyczącą klasycznej fizyki statystycznej, procesów stochastycznych (łańcuchy Markowa) i algorytmu Metropolisa, najczęściej stosowanej realizacji metody MC. Autor zastosował technikę MC do wyznaczenia termodynamicznych właściwości dwuwymiarowego modelu Isinga, który jest dziś klasycznym paradygmatem fizyki przejść fazowych.

Mgr inż. Karol Tarnowski jest autorem trzeciej części pt. Rozwiązywanie równań Maxwella metodą FDTD. Nazwa metody jest akronimem od Finite-Difference Time-Domain. Jest to jedna z podstawowych technologii komputerowych stosowanych w obliczeniowej elektrodynamice klasycznej [1,7,21,28,44,45,54,57,76,80,106]. Umożliwia numeryczne rozwiązywanie w rzeczywistym czasie i w przestrzeni równań Maxwella, którymi są prawa Faradaya oraz Ampere'a-Maxwella. Istota podejścia polega na dyskretyzacji czasu i przestrzeni oraz zastosowaniu metod różnic skończonych. Autor w przystępny sposób przedstawił najważniejsze aspekty numeryczne metody opartej na klasycznej pracy Kane Yee z 1966 r. Zamieścił także kilka przykładowych zastosowań metody FDTD w postaci programów napisanych w środowisku obliczeniowym MatLab® [72,77,79,83,94,107,126,129]. Zaprezentowana technologia jest obecnie intensywnie stosowana do modelowania oddziaływania fal elektromagnetycznych z materią, co ma fundamentalne znaczenie w badaniach właściwości propagacji i przestrzennych rozkładów pola elektromagnetycznego m.in. w kryształach fotonicznych, światłowodach (klasycznych i fotonicznych) [134–144] oraz metamateriałach [145–156].

Autorem ostatniej, czwartej części skryptu, noszącej tytuł Metoda elementów skończonych - zastosowanie w elektromagnetyzmie, jest dr inż. Jacek Olszewski. Metoda ta w literaturze źródłowej nazywana jest skrótowo w języku angielskim FEM od Finite Element Methods [1,7,26,28,33,45,57,73,80,90,91], a w języku polskim używa się akronimu MES od Metoda Elementów Skończonych. Ma ona bardzo szerokie zastosowania nie tylko w elektromagnetyzmie, ale również m.in. w meteorologii, w dynamice budowli (budynków, mostów), w przemyśle lotniczym oraz do analizowania wytrzymałości układów mechanicznych. Najogólniej mówiąc FEM służy to wyznaczaniu rozwiązań numerycznych równań całkowych i równań różniczkowych cząstkowych [54, 126]. Autor przedstawił podstawy matematyczne MES i jej zastosowanie do modelowania właściwości propagacji fal elektromagnetycznych w światłowodach, falowodach oraz kryształach i światłowodach fotonicznych. Metoda ta znalazła i wciąż znajduje szerokie zastosowania w elektro-magnetyzmie, fotonice, nanofotonice i optoelektronice [134–144]. Jest to dość trudna pod względem matematycznym i fizycznym metoda, której głębsze zrozumienie wymaga od studentów wytrwałości i poświęcenia dość dużo czasu niezbędnego do zdobycia matematycznej wiedzy oraz nabycia praktycznych umiejętności efektywnego posługiwania się MES. Dodajmy, że metody FDTD i FEM są stosowane z powodzeniem do modelowania właściwości metamateriałów elektromagnetycznych [145–156] - nowych ośrodków wykazujących zjawisko ujemnego załamania fal elektromagnetycznych, co stwarza realne możliwości ukrywania obiektów w polach elektromagnetycznych. Wytworzenie okryć/czapek/peleryn niewidek byłoby realizacją idei/marzeń, o których można było do tej pory jedynie fantazjować w



bajkach, filmach i literaturze science fiction. Konkludując pragnę wyrazić moje głębokie przekonanie, że skrypt ten realizuje przedstawione na wstępie główne cele i jest wartościowym materiałem edukacyjnym, który przyczyniać się będzie do podnoszenia poziomu wykształcenia i kompetencji (wiedzy i umiejętności) absolwentów Wydziału Podstawowych Problemów Techniki Politechniki Wrocławskiej.

Wrocław, 9 czerwca 2011

Włodzimierz Salejda



Bibliografia

- [1] L.D. Landau, E.M. Lifshjitz, *Course of Theoretical Physics, vol. 1st Mechanics, vol. 2nd The Classical Theory of Fields, vol. 3rd Quantum Mechanics - Nonrelativistic Theory, vol. 4th Quantum Electrodynamics, vol. 5th Statistical Physics, Part 1., vol. 6th Fluid Mechanics, vol. 7th Theory of Elasticity, vol. 8th Electrodynamics of Continuous Media, vol. 9th Statistical Physics, Part 2., vol. 10th Physical Physics.*
- [2] W. Greiner, B. Müller, J. Rafelski, *Quantum Electrodynamics of Strong Fields*, Springer-Verlag, Berlin 1985.
- [3] W. Greiner, B. Müller, *Quantum Mechanics. Symmetries*, Springer-Verlag, Berlin 1994.
- [4] W. Greiner, J. A. Maruhn, *Nuclear Models*, Springer-Verlag, Berlin 1996.
- [5] W. Greiner, J. Reinhardt, *Field Quantization*, Springer-Verlag, Berlin 1996.
- [6] W. Greiner, *Thermodynamics and Statistical Mechanics*, Springer-Verlag, Berlin 1997; w serii *Classical Theoretical Physics*.
- [7] W. Greiner, *Classical Electrodynamics*, Springer-Verlag, Berlin 1998; w serii *Classical Theoretical Physics*. J.D. Jackson, *Classical Electrodynamics*, A John Wiley & Sons, New York 1990. D. Griffiths, *Introduction to Electrodynamics*, Prentice-Hall, Inc., New Jersey 1999; istnieje tłumaczenie na j. polski, *Podstawy elektrodynamiki*, PWN, Warszawa 2005.
- [8] W. Greiner, B. Müller, *Quantum Mechanics. Special Chapters*, Springer-Verlag, Berlin 1998.
- [9] W. Greiner, B. Müller, *Gauge Theory of Weak Interactions*, Springer-Verlag, Berlin 2000.
- [10] W. Greiner, B. Müller, *Relativistic Quantum Mechanics. Wave Equations*, 3rd Edition, Springer-Verlag, Berlin 2000.
- [11] W. Greiner, B. Müller, *Quantum Mechanics. An Introduction*, 4th Edition, Springer-Verlag, Berlin 2001.
- [12] W. Greiner, J. Reinhardt, *Quantum Electrodynamics*, 3rd Edition, Springer-Verlag, Berlin 2003.
- [13] W. Greiner, *Classical Mechanics. Systems of Particles and Hamiltonian Dynamics*, Springer-Verlag, Berlin 2003; 2nd Edition 2010; w serii *Classical Theoretical Physics*.
- [14] W. Greiner, S. Schramm, E. Stein, *Quantum Chromodynamics*, 3rd Edition, Springer-Verlag, Berlin 2007.



-
- [15] J. H. Wilkinson, *Algebraic Eigenvalue Problem*, Oxford University Press, Oxford 1965.
- [16] J. H. Wilkinson, C. Reinsch, *Handbook for Automatic Computation*, vol. 2, Linear Algebra, Springer-Verlag, Berlin 1971.
- [17] S. D. Conte, C. de Boor, *Elementary Numerical Analysis. An Algorithmic Approach*, The McGraw-Hill Companies, 1980.
- [18] D. Potter, *Metody obliczeniowe fizyki*, PWN, Warszawa 1982.
- [19] S. E. Koonin, *Computational Physics*, The Benjamin/Cummings Publishing Company, Inc., Amsterdam 1986.
- [20] D. Heermann, *Podstawy symulacji komputerowych w fizyce*, WN-T, Warszawa 1997; tytuł oryginału w j. ang. *Computer Simulation Methods in Theoretical Physics*, 2nd Edition, Springer-Verlag, Berlin 1990.
- [21] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. I, Finite Difference Methods (Part 1), Amsterdam 1990.
- [22] J. C. Nash, *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*, Adam Hilger, New York 1990.
- [23] R. L. Graham, D. E. Knuth, O. Patashnik, *Concrete Mathematics*, 6th Edition, Addison-Wesley, NY 1990.
- [24] S. E. Koonin, D. C. Meredith, *Computational Physics. Fortran Version*, Addison-Wesley Publishing Company, Inc., Amsterdam 1990.
- [25] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press Series in Algorithms and Architectures for Advanced Scientific Computing, Manchester 1991.
- [26] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. II, Finite Element Methods, Amsterdam 1991.
- [27] D. Kincaid, W. Cheney, *Numerical Analysis. Mathematics of Scientific Computing*, Brooks/Cole Publishing Company, Belmont, California 1991; 3rd Edition, Wadsworth Group, a division of Thompson Learning, Inc., 2002; istnieje tłumaczenie na j. polski, *Analiza numeryczna*, w przekładzie i pod redakcją S. Paszkowskiego, WN-T, Warszawa 2006.
- [28] K. S. Kunz, R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*, CRC Press, London 1993.
- [29] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 2nd Edition, Springer-Verlag, New York 1993.
- [30] P. G. Ciarlet, J. L. Lions Eds., *Handbook of Numerical Analysis*, Elsevier Science, vol. III, Techniques of Scientific Computing (Part 1), Amsterdam 1994.
- [31] P. L. DeVries, *A First Course in Computational Physics*, A John Wiley & Sons, New York 1994.



-
- [32] G. H. Golub, Ch. F. Van Loan, *Matrix Computation*, 3rd Edition, The Johns Hopkins Press, London 1996.
- [33] P. G. Ciarlet, J. L. Lions Eds., *Handbook of Numerical Analysis*, Elsevier Science, vol. IV, Finite Element Methods (Part 2), Numerical Methods for Solids (Part 2) Amsterdam 1996.
- [34] N. J. Giordano, *Computational Physics*, Prentice-Hall, New Jersey 1997; N. J. Giordano, H. Nakanishi, *Computational Physics*, Prentice-Hall, 2nd Edition, New Jersey 2005.
- [35] M. T. Heath, *Scientific Computing. An Introductory Survey*, The McGraw-Hill Companies, 1997.
- [36] L. S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, *ScaLAPACK Users’ Guide*, SIAM, Philadelphia 1997.
- [37] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [38] W. Press, S. H. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in Fortran. The art of Scientific Computing. Volume 1 of Fortran Numerical Recipes*, 2nd Edition, Cambridge University Press, Cambridge 1997.
- [39] P. G. Ciarlet Ed., *Handbook of Numerical Analysis, Elsevier Science, vol. V, Techniques of Scientific Computing (Part 2)*, Amsterdam 1997.
- [40] P. G. Ciarlet Ed., *Handbook of Numerical Analysis, Elsevier Science, vol. VI, Numerical Methods for Solids (Part 3), Numerical Methods for Solids (Part 1)*, Amsterdam 1998.
- [41] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *Lapack Users’ Guide*, 3rd Edition, SIAM, Philadelphia 1999.
- [42] J. M. Thijssen, *Computational Physics*, Cambridge University Press, Cambridge 1999.
- [43] M. M. Woolfson, G. J. Pert, *An Introduction to Computer Simulation*, Oxford University Press, New York, 1999.
- [44] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Methods*, IEEE Press Series on RF and Microwave Technology, New York 2000.
- [45] M. V. K. Chari, S. J. Salon, *Numerical Methods in Electromagnetism*, Academic Press, London 2000.
- [46] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. VII, Solution of Equation in R_n (Part 3), Techniques of Scientific Computing (Part 3), Amsterdam 2000.
- [47] D. P. Landau, K. Binder, *A Guide to Monte Carlo Simulation in Statistical Physics*, Cambridge University Press, Cambridge 2000.
-



- [48] Journal of Computational and Applied Mathematics, vol. 123, 2000, vol. III: Linear algebra, Eigenvalue computation in the 20th century, zawartość: Y. Saad, H.A. van der Vorst, *Iterative solution of linear systems in the 20th century*, pages 1-33; G.H. Golub, H.A. van der Vorst, *Eigenvalue computation in the 20th century*, pages 35-65; D.S. Watkins, *QR-like algorithms for eigenvalue problems*, pages 67-83; Ch. F. Van Loan, *The ubiquitous Kronecker product*, pages 85-100; R.B. Morgan, *Preconditioning eigenvalues and some comparison of solvers*, pages 101-115; B.N. Parlett, *For tridiagonals T replace T with LDL^T* , pages 117-130; I.C.F. Ipsen, *An overview of relative $\sin \Theta$ theorems for invariant subspace of complex matrices*, pages 131-153; A. sameh, Z. Tong, *The trace minimization method for the symmetric generalized eigenvalue problem*, pages 115-175; A. Hadjidimos, *Successive overrelaxation (SOR) and related methods*, pages 177-199; A. Frommer, D.B. Szyld, *On asynchronous iterations*, pages 201-216; D. Calvetti, L. Reichel, *Iterative methods for large continuation problems*, pages 217-240; C. Brezinski, M. Redivo-Zaglia, H. Sadok, *The matrix and polynomial approaches to Lanczos-type algorithms*, pages 241-260; M. Eiermann, O. Ernst, O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, pages 261-292; R. Bridson, W-P. Tang, *Refining an approximate inverse*, pages 293-306; J. Koning, G. Rodrigue, D. White, *Scalable preconditioned conjugate gradient inversion of vector finite element mass matrices*, pages 307-321; T.F. Chan, W.L. Wan, *Robust multigrid methods for nonsmooth coefficient elliptic linear systems*, pages 323-352; G. Poole, L. Neal, *The Rook's pivoting strategy*, pages 353-369; V. Mehrmann, H. Xu, *Numerical methods in control*, pages 371-394; R.W. Freund, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, pages 395-421; D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, *Tikhonov regularization and the L-curve for large discrete ill-posed problems*, pages 423-446; D.P. O'Leary, *Symbiosis between linear algebra and optimization*, pages 447-465; R.J. Plemmons, V.P. Pauca, *Some computational problems arising in adaptive optics imaging systems*, pages 467-487; J.J. Dongarra, V. Eijkhout, *Numerical linear algebra algorithms and software*, pages 489-514; I.S. Duff, *The impact of high-performance computing in the solution of linear systems: trends and problems*, pages 515-530.
- [49] Journal of Computational and Applied Mathematics, vol. 125, 2000, Numerical Analysis 2000, Vol. IV: Optimization and Nonlinear Equations; znakomity zbiór artykułów wybitnych specjalistów poświęconych metodom numerycznym rozwiązywania problemów zdefiniowanych tytułem tomu.
- [50] Journal of Computational and Applied Mathematics, vol. 125, 2000, Numerical Analysis 2000, Vol. VI: Ordinary Differential Equations and Integral Equations; znakomity zbiór artykułów wybitnych specjalistów poświęconych metodom numerycznym rozwiązywania problemów zdefiniowanych w tytule.
- [51] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia 2000.
- [52] M. E. J. Newman, G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press, Oxford 2001.
- [53] Journal of Computational and Applied Mathematics, vol. 127, No 1-2, 2001, Quadrature and Orthogonal Polynomials.
- [54] Journal of Computational and Applied Mathematics, vol. 128, 2001, Numerical Analysis 2000, Vol. VII: Partial Difference Equations, wybór artykułów wybitnych specjalistów poświęconych metodom numerycznym rozwiązywania problemów zdefiniowanych w tytule.



- [55] G. W. Steward, *Matrix Algorithms, Volume II: Eigensystems*, SIAM, Philadelphia 2001.
- [56] J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, 2nd Edition, Marcel Dekker Inc., New York 2001.
- [57] M. N. O Sadiku, *Numerical Techniques in Electromagnetics*, CRC Press, London 2001; K. Kawano, T. Kitoh, *Introduction to Optical Waveguide Analysis. Solving Maxwell's Equations and Schrödinger Equation*, A John Wiley & Sons, New York 2001.
- [58] L. Komzsik, *The Lanczos Methods. Evolution and Application*, SIAM, Philadelphia 2001.
- [59] J. K. Cullum, R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, vol. I: Theory, Classics in Applied Mathematics series*, SIAM, Philadelphia 2002.
- [60] M. Matyka, *Symulacje komputerowe w fizyce*, Helion, Gliwice 2002.
- [61] W. Salejda, M.H. Tyc, M. Just, *Algebraiczne metody rozwiązywania równania Schrödingera*, PWN, Warszawa 2002.
- [62] S. Mauch, *Introduction to Methods of Applied Mathematics or Advanced Mathematical Methods for Scientists and Engineers*, <http://ebookbrowse.com/sean-mauch-introduction-to-methods-of-applied-mathematics-pdf-d57631942>, 2002.
- [63] D. S. Watkins, *Fundamentals of Matrix Computations*, A John Wiley & Sons, Wiley-Interscience, Hoboken 2002.
- [64] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. VIII, Solution of Equations in n (Part 4), Techniques of Scientific Computing (Part4), Numerical Methods for Fluids (Part 2), Amsterdam 2002.
- [65] N. J. Higham, *Accuracy and Stability of Numerical Algorithms, 2nd Edition*, SIAM, Philadelphia 2002.
- [66] R. Fitzpatrick, *Computational Physics: An introductory course*, <http://farside.ph.utexas.edu/teaching/329/lectures/lectures.html>; wersja TEX-owa 2002.
- [67] E. Süli, D. F. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, Cambridge 2003.
- [68] P. G. Ciarlet Ed., *Handbook of Numerical Analysis, Elsevier Science, vol. IX, Numerical Methods for Fluids (Part 3)*, Amsterdam 2003.
- [69] M. Hjorth-Jensen, *Computational Physics*, University of Oslo, Oslo 2003.
- [70] P. G. Ciarlet Ed., *Handbook of Numerical Analysis, Elsevier Science, vol. X, Computational Chemistry*, Amsterdam 2003.
- [71] P. G. Ciarlet Ed., *Handbook of Numerical Analysis, Elsevier Science, vol. XI, Foundations of Computational Mathematics*, Amsterdam 2003.



- [72] S. E. Lyshevski, *Engineering and Scientific Computations Using MATLAB®*, A John Wiley & Sons, Wiley-Interscience, Hoboken 2003.
- [73] P. Monk, *Finite Element Methods for Maxwell's Equations*, Clarendon Press, Oxford 2003.
- [74] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. XII, Computational Models for the Human Body, Amsterdam 2004.
- [75] R. M. Martin, *Electronic Structure. Basic Theory and Practical Methods*, Cambridge University Press, Cambridge 2004.
- [76] A. Taflove, *Computational Electrodynamics. The Finite-Difference Time-Domain Method*, Artech House, Inc., 3rd Edition, London 2005; pierwsze wydanie książki pochodzi z 1995 r.
- [77] W. Y. Yang, W. Cao, T-S. Chung, J. Morris, *Applied Numerical Methods Using Matlab®*, A John Wiley & Sons, Wiley-Interscience, Hoboken 2005.
- [78] M-C. T. Calderer, E.M. Terentjev, Eds., *Modeling of Soft Matter*, Springer Science+Business Media, Inc, New York 2005.
- [79] J. Kiusalaas, *Numerical Methods in Engineering with MATLAB®*, Cambridge University Press, Cambridge 2005.
- [80] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. XIII, Numerical Methods in Electrodynamics, Amsterdam 2005.
- [81] D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems*, Springer-Verlag, Berlin 2005.
- [82] P. Harrison, *Quantum Wells, Wires and Dots. Theoretical and Computational Physics of Semiconductor Nanostructures*, 2nd Edition, A John Wiley & Sons, Wiley-Interscience, Hoboken 2005.
- [83] C. T. Lachowicz, *Matlab®, Scilab®, Maxima®. Opis i przykłady zastosowań.*, Wydawnictwo Politechniki Opolskiej, Opole 2005.
- [84] R.H. Enns, *Computer Algebra Recipes for Mathematical Physics*, Birkhäuser, Boston 2005.
- [85] A. Klein, A. Godunov, *Introductory Computational Physics*, Cambridge University Press, Cambridge 2006.
- [86] T. Pang, *An Introduction to Computational Physics*, 2nd Edition, Cambridge University Press, Cambridge 2006; istnieje tłumaczenie na j. polski pierwszego wydania *Metody obliczeniowe w fizyce*, PWN, Warszawa 2001.
- [87] E. Mulas, *Przykłady symulacji komputerowej w fizyce*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2006.
- [88] M. Trott, *The Mathematica Guide Book for Numerics*, Springer Science+Business Media, Inc, New York 2006.
- [89] K. F. Riley, M. P. Hobson, S. J. Bence, *Mathematical Methods for Physics and Engineering*, Cambridge University Press, Cambridge 2006.



- [90] M. S. Gockenbach, *Understanding and Implementing the Finite Element Method*, SIAM, Philadelphia 2006.
- [91] D. T. Nguyen, *Finite Element Methods: Parallel-Sparse Statics and Eigen-Solutions*, Springer Science+Business Media, Inc, New York 2006.
- [92] S. C. Chapra, R. P. Canale, *Numerical Methods for Engineers*, Mc Graw Hill, Boston 2006.
- [93] W-H. Steeb, *Problems and Solutions in Introductory and Advanced Matrix Calculus*, World Scientific Publishing Co., Singapore 2006.
- [94] I. Danaila, P. Joly, S. M. Kaber, M. Postel, *An Introduction to Scientific Computing: Twelve Computational Projects Solved with Matlab®*, Springer Science+Business Media, Berlin 2007.
- [95] M. K. Jain, S. R. K. Iyengar, R. K. Jain, *Numerical Methods*, New Age International (P) Limited, Publishers, New Delhi 2007.
- [96] D. W. Jordan, P. Smith, *Nonlinear Ordinary Differential Equations*, Oxford University Press, New York, 2007.
- [97] D. S. Watkins, *The Matrix Eigenvalue Problem. GR and Krylov Subspace Methods*, SIAM, Philadelphia 2007.
- [98] R. H. Landau, M. J. Paez, C. C. Bordeianu, *Computational Physics. Problem Solving with Computers*, 2nd Edition, Wiley-VCH Verlag GmbH & Co KGaA, Weinheim 2007.
- [99] T. S. Shores, *Applied Linear Algebra and Matrix Analysis*, Springer Science+Business Media, Inc, New York 2007.
- [100] R. H. Landau, M. J. Paez, C. C. Bordeianu, *A Survey of Computational Physics*, Princeton University Press, Princeton 2008.
- [101] G. Dahlquist, A. Björk, *Methods in Scientific Computing*, vol. 1, SIAM, Philadelphia 2008; pierwsze wydanie pochodzi z 1974 r.
- [102] P. T. Tam, *A Physicist's Guide to Mathematica®*, 2nd Edition, Academic Press Elsevier, London 2008.
- [103] H. Fehske, R. Schneider, A. Wei e Eds., *Computational Many-Particle Physics*, seria Lect. Notes Phys. 739, Springer, Berlin 2008.
- [104] G. O'Regan, *A Brief History of Computing*, Springer Verlag, London Limited 2008.
- [105] J. Sanchez, M. P. Canton, *Software Solutions for Engineers and Scientists*, CRC Press, Boca Raton 2008.
- [106] Y. Hao, R. Mittra, *FDTD Modeling of Metamaterials*, Artech House, Inc, Norwood 2009.
- [107] T. A. Driscoll, *Learning MATLAB®*, SIAM, Philadelphia 2009.
- [108] D. P. Landau, K. Binder, *A Guide to Monte Carlo Simulation in Statistical Physics*, 3rd Edition, Cambridge University Press, Cambridge 2009.



- [109] D. P. O’Leary, *Scientific Computing with case studies*, SIAM, Philadelphia 2009.
- [110] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. XIV, Modeling (Part 1), Nonlinear Methods (Part 2), Turbulence (Part 3), Data Assimilation (Part 4), Analysis (Part 5), Amsterdam 2009.
- [111] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. XV, Part 1: Mathematical Models, Part 2: Computational Methods, Part 3: Applications, Amsterdam 2009.
- [112] G. H. Gonnet, R. Scholl, *Scientific Computation*, Cambridge University Press, Cambridge 2009.
- [113] D. Marx, J. Hutteer, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*, Cambridge University Press, Cambridge 2009.
- [114] P. O. J. Scherer, *Computational Physics. Simulation of Classical and Quantum Systems*, Springer-Verlag, Berlin 2010.
- [115] W. Dos Passos, *Numerical Methods, Algorithms and Tools in C#*, CRC Press, London 2010.
- [116] F. Olafsen Ed., *Experimental and Computational Techniques in Soft Condensed Matter Physics*, Cambridge University Press, Cambridge 2010.
- [117] V. Eijkhout, E. Chow, R. van de Geijn, *Introduction to High-Performance Scientific Computing*, 1st Edition, 2011; <http://www.multicoreinfo.com/2011/01/free-book-introduction-to-high-performance-scientific-computing/>
- [118] R. L. Burden, J.D. Faiers, *Numerical Analysis*, 9th Edition, <http://www.as.yzu.edu/~faires/Numerical-Analysis/>; Brooks Cole, Cengage Learning, Boston 2011; wcześniejsze wydania 8th w 2005 i 3rd w 2003.
- [119] N. Boccara, *Modeling Complex Systems*, 2nd Edition, Springer Science+Business Media, Inc, New York 2010.
- [120] K. Naone, K. Teranishi, J. Cavazos, R. Suda, *Software Automatic Tuning. From Concepts to State-of-the Art Results*, Springer Science+Business Media, Inc, New York 2010.
- [121] S. Mark. S. Mordechai Eds., *Application of Monte Carlo Method in Science and Engineering*, InTech, Rijeka 2011.
- [122] J. Kurzak, D. A. Bader, J. Dongarra, *Scientific Computing with Multicore and Accelerators*, CRC Press, London 2011.
- [123] S. Koranne, *Handbook of Open Source Tools*, Springer Science+Business Media, Inc, New York 2011.
- [124] P. Stpiczyński, M. Brzuszek. *Podstawy programowania obliczeń równoległych*, UMCS, Wydział Matematyki, Fizyki i Informatyki, Instytut Informatyki, Lublin 2011.
- [125] P. G. Ciarlet Ed., *Handbook of Numerical Analysis*, Elsevier Science, vol. XVI, Numerical Methods for Non-Newtonian Fluids, Amsterdam 2011.



- [126] G. W. Griffiths, W. E. Schiesser, *Traveling Wave Analysis of Partial Differential Equations. Numerical and Analytical Methods with MATLAB® and Maple™*, to be published, Academic Press, Elsevier, Amsterdam 2012.
- [127] Wolfram Mathematica®8, <http://www.wolfram.com/mathematica/>.
- [128] Maple™, <http://www.maplesoft.com/products/maple/>
- [129] Matlab®, http://www.mathworks.com/products/matlab/?s_cid=global_nav
- [130] Maxima, a Computer Algebra System, <http://maxima.sourceforge.net/>
- [131] MathCad™, <http://www.ptc.com/products/mathcad/>
- [132] Journal of Computational Physics, http://www.elsevier.com/wps/find/journaldescription.cws_home/622866/description#description.
- [133] Computer Physics Communication, http://www.elsevier.com/wps/find/journaldescription.cws_home/706710/description#description.
- [134] B. E. A. Saleh, M. C. Teich, *Fundamentals of Photonics*, A John Wiley & Sons, New York 1991.
- [135] P. N. Prasad, *Nanophotonics*, A John Wiley & Sons, New York 2004.
- [136] F. Zolla, G. Renversez, A. Nicolet, B. Kuhlmeiy, S. Guenneau, D. Felbacq, *Foundation of Photonic Crystal Fibers*, World Scientific Publishing Co., Singapore 2005.
- [137] K. Sakoda, *Optical Properties of Photonic Crystals*, Springer-Verlag, Berlin 2005.
- [138] H. Rigneault, J-M Lourtioz, C. Delalande, A. Levenson, Eds., *Nonophotonics*, ISTE Ltd, London 2006.
- [139] K. Zhang, D. Li, *Electromagnetic Theory for Microwaves and Optoelectronics*, Springer-Verlag, Berlin 2008.
- [140] D. Fleisch, *A Student's Guide to Maxwell Equations*, Cambridge University Press, Cambridge 2008.
- [141] S. V. Gaponenko, *Introduction to Nanophotonics*, Cambridge University Press, Cambridge 2010.
- [142] Z. M. Wang, A. N. Eds., *Nanoscale Photonics and Optoelectronics*, Springer Science+Business Media, Inc, New York 2010.
- [143] J-M. Jin, *Theory and Computation of Electromagnetic Fields*, A John Wiley & Sons, New York 2010.
- [144] John D. Joannopoulos, Steven G. Johnson, Joshua N. Winn, and Robert D. Meade, *Photonic Crystals: Molding the Flow of Light*, 2nd Edition, książka dostępna na stronie <http://ab-initio.mit.edu/book/>; Princeton University Press, Princeton 2008.
- [145] G. V. Eleftheriades, K. G. Balmain, Eds.,; *Negative-refraction metamaterials. Fundamentals Principle and Applications*, IEEE Press 2005.



- [146] N. Engheta, R. W. Ziolkowski, Eds., *Metamaterials. Physics and Engineering Explorations*, IEEE Press 2006.
- [147] Ch. Caloz, T. Itoh, *Electromagnetic Metamaterials: Transmission Line Theory and Microwave Applications*, Wiley and Sons, New York 2006.
- [148] A. K. Sarychev, V. M. Shalaev, *Electrodynamics of Metamaterials*, World Scientific Publishing Co., Singapore 2007.
- [149] C. M. Krowne, Y. Zhang, Eds., *Physics of Negative Refraction and Negative Index Materials*, Springer-Verlag, Berlin 2007
- [150] F. Capolino Ed., *Metamaterials Handbook. Theory and Phenomena of Metamaterials*, CRC Press, 2008.
- [151] R. Marques, F. Martin, M. Sorolla, *Metamaterials with Negative Parameters. Theory, Design, and Microwave Applications*, A John Wiley & Sons, Wiley-Interscience, Hoboken 2008.
- [152] P. Markos, C. M. Soukoulis, *Wave Propagation. From Electrons to Photonic Crystals and Left-Handed Materials*, Princeton University Press, Princeton 2008.
- [153] S. A. Ramakrishna, T. M. Grzegorzcyk, *Physics and Applications of Negative Refractive Index Materials*, SPIE Press and CRC Press 2009.
- [154] T. J. Cui, D. R. Smith, R. Liu, Eds., *Metamaterials. Theory, Design and Applications*, Springer-Verlag, Berlin 2009.
- [155] L. Solymar, E. Shamonina, *Waves in Metamaterials*, Oxford University Press, Oxford 2009.
- [156] W. Cui, V. Shalaev, *Optical Metamaterials*, Springer-Verlag, Berlin 2010.



Rozdział 1.

Podstawy symulacji komputerowych metodą dynamiki molekularnej

Wstęp

Celem niniejszego skryptu jest zapoznanie Czytelnika z podstawowymi zagadnieniami związanymi z dynamiką molekularną (ang. „*molecular dynamics*”, MD). Jest to, oprócz metody Monte Carlo (MC), jedna z podstawowych technik badawczych w fizyce komputerowej. Tekst ten nie pretenduje do bycia głównym materiałem dydaktycznym z zakresu dynamiki molekularnej, za takie można uznać książki [1], [2], ale stanowi raczej wyjściowy materiał, po poznaniu którego Czytelnikowi będzie łatwiej rozpocząć przygodę z fizyką komputerową.

Skrypt ten został podzielony na cztery główne części, w których to przedstawione zostaną podstawowe algorytmy, wielkości zredukowane, dwuwymiarowy układ Lennarda-Jonesa oraz symulacje w układzie NVE .

Od Czytelnika wymaga się znajomości podstawowych technik programowania w jednym z języków, n.p. C/C++, Fortran 77/95, Pascal. W tekście zostały umieszczone przykładowe algorytmy zapisane w pseudokodzie. Celem takiego postępowania jest uniezależnienie implementacji danego algorytmu od używanego przez Czytelnika języka programowania. Rezygnacja z prezentacji algorytmów w konkretnym języku, n.p. Fortran 77, ma, w odczuciu Autora, ułatwić odbiór tekstu tym, którzy z tym językiem się nie spotkali w swojej dotychczasowej karierze programistycznej.

1.1. Podstawowe algorytmy

1.1.1. Całkowanie równań ruchu Newtona

Podstawowym zadaniem do rozwiązania przy korzystaniu z symulacji komputerowych metodą dynamiki molekularnej jest rozwiązanie równań ruchu Newtona. W ogólnym przypadku są one następującej postaci:

$$m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = \vec{F}_i(t), \quad (1.1)$$

gdzie indeks $i = 1, 2, \dots, N$ numeruje cząstki; m_i , $\vec{r}_i(t)$, $\vec{F}_i(t)$ to odpowiednio – masa i -tej cząstki, jej wektor wodzący oraz całkowita siła działająca na nią.



Równoważny sposób zapisu równania (1.1) to układ dwóch równań różniczkowych zwyczajnych pierwszego stopnia:

$$\frac{d\vec{r}_i(t)}{dt} = \vec{v}_i(t), \quad (1.2)$$

$$m_i \frac{d\vec{v}_i(t)}{dt} = \vec{F}_i(t). \quad (1.3)$$

Algorytm Eulera

Najprostszym algorytmem stosowanym w dynamice molekularnej jest algorytm Eulera. Korzystając z definicji pochodnej jako ilorazu różnicowego możemy napisać równania (1.2)-(1.3) w postaci:

$$v_i(t) = \lim_{h \rightarrow 0} \frac{x_i(t+h) - x_i(t)}{h}, \quad (1.4)$$

$$F_i(t) = m_i \lim_{h \rightarrow 0} \frac{v_i(t+h) - v_i(t)}{h}, \quad (1.5)$$

gdzie dla prostoty zapisu rozpatrujemy przypadek jednowymiarowy: $x_i(t) \equiv \vec{r}_i(t)$, $v_i(t) \equiv \vec{v}_i(t)$. Dodatkowo, zakładając, że h ma pewną skończoną małą wartość powyższe wzory można przedstawić jako:

$$v_i(t) \approx \frac{x_i(t+h) - x_i(t)}{h}, \quad (1.6)$$

$$F_i(t) \approx m_i \frac{v_i(t+h) - v_i(t)}{h}. \quad (1.7)$$

Stąd, rozwiązując oba równania ze względu na $x_i(t+h)$ oraz $v_i(t+h)$ otrzymujemy:

$$x_i(t+h) \approx x_i(t) + v_i(t)h, \quad (1.8)$$

$$v_i(t+h) \approx v_i(t) + \frac{F_i(t)}{m_i}h. \quad (1.9)$$

Algorytm Verleta (postać położeniowa)

Drugą metodą rozwiązywania równań ruchu Newtona jest skorzystanie z algorytmu Verleta w postaci położeniowej. Jego wyprowadzenie przedstawiamy poniżej.

Podobnie jak w Rozdziale 1.1.1. rozpatrywać będziemy przypadek jednowymiarowy. Rozwiniemy w szereg Taylora funkcję $x_i(t)$ dla dwóch chwil czasu: $t+h$ oraz $t-h$:

$$\begin{aligned} x_i(t+h) &= x_i(t) + \frac{1}{1!} \left. \frac{dx_i(\tau)}{d\tau} \right|_{\tau=t} h + \frac{1}{2!} \left. \frac{d^2x_i(\tau)}{d\tau^2} \right|_{\tau=t} h^2 + \frac{1}{3!} \left. \frac{d^3x_i(\tau)}{d\tau^3} \right|_{\tau=t} h^3 + \dots \\ &= x_i(t) + v_i(t)h + \frac{F_i(t)}{2m_i}h^2 + \frac{1}{3!} \left. \frac{d^3x_i(\tau)}{d\tau^3} \right|_{\tau=t} h^3 + \dots \end{aligned} \quad (1.10)$$

$$\begin{aligned} x_i(t-h) &= x_i(t) + \frac{1}{1!} \left. \frac{dx_i(\tau)}{d\tau} \right|_{\tau=t} (-h) + \frac{1}{2!} \left. \frac{d^2x_i(\tau)}{d\tau^2} \right|_{\tau=t} (-h)^2 + \frac{1}{3!} \left. \frac{d^3x_i(\tau)}{d\tau^3} \right|_{\tau=t} (-h)^3 + \dots \\ &= x_i(t) - v_i(t)h + \frac{F_i(t)}{2m_i}h^2 - \frac{1}{3!} \left. \frac{d^3x_i(\tau)}{d\tau^3} \right|_{\tau=t} h^3 + \dots \end{aligned} \quad (1.11)$$

Celem wyznaczenie $x_i(t+h)$ dodajmy do stronami siebie równania (1.10) oraz (1.11):

$$x_i(t+h) + x_i(t-h) = 2x_i(t) + \frac{F_i(t)}{m_i}h^2 + \dots + \mathcal{O}(h^4) \quad (1.12)$$

Rozwiązując równanie (1.12) względem $x_i(t+h)$ otrzymujemy:

$$x_i(t+h) \approx x_i(t-h) + 2x_i(t) + \frac{F_i(t)}{m_i}h^2. \quad (1.13)$$

Podobnie postępujemy wyznaczając $v_i(t)$, tym razem jednak odejmiemy od siebie równania (1.10) oraz (1.11):

$$x_i(t+h) - x_i(t-h) = 2v_i(t)h + \dots + \mathcal{O}(h^3) \quad (1.14)$$

Rozwiązując równanie (1.14) względem $v_i(t)$ otrzymujemy:

$$v_i(t) \approx \frac{x_i(t+h) - x_i(t-h)}{2h}. \quad (1.15)$$

Zwróćmy uwagę, że algorytm Verleta w postaci położeniowej nie pozawala nam na jednoczesne wyznaczenie położenia oraz prędkości w chwili czasu $t+h$.

Algorytm Verleta (postać prędkościowa)

Kolejną, bardziej dokładną, metodą rozwiązywania równań ruchu Newtona jest skorzystanie z algorytmu Verleta w postaci prędkościowej.

Podobnie jak w Rozdziale 1.1.1. rozwiniemy $x_i(t)$ w szereg Taylora:

$$\begin{aligned} x_i(t+h) &= x_i(t) + \frac{1}{1!} \left. \frac{dx_i(\tau)}{d\tau} \right|_{\tau=t} h + \frac{1}{2!} \left. \frac{d^2x_i(\tau)}{d\tau^2} \right|_{\tau=t} h^2 + \frac{1}{3!} \left. \frac{d^3x_i(\tau)}{d\tau^3} \right|_{\tau=t} h^3 + \dots \\ &\approx x_i(t) + v_i(t)h + \frac{F_i(t)}{2m_i}h^2. \end{aligned} \quad (1.16)$$

W przypadku prędkości $v_i(t)$ otrzymujemy:

$$\begin{aligned} v_i(t+h) &= v_i(t) + \frac{1}{1!} \left. \frac{dv_i(\tau)}{d\tau} \right|_{\tau=t} h + \frac{1}{2!} \left. \frac{d^2v_i(\tau)}{d\tau^2} \right|_{\tau=t} h^2 + \dots \\ &\approx v_i(t) + \frac{F_i(t)}{2m_i}h + \frac{1}{2!} \left. \frac{d^2v_i(\tau)}{d\tau^2} \right|_{\tau=t} h^2 \end{aligned} \quad (1.17)$$

W równaniu (1.17) pozostaje nadal nieznanne wyrażenie postaci $\frac{d^2v_i(t)}{dt^2} \equiv \ddot{v}_i(t)$. Z drugiej jednak strony:

$$\dot{v}_i(t+h) = \dot{v}_i(t) + \frac{1}{1!} \frac{d}{dt} \dot{v}_i(t)h + \dots = \dot{v}_i(t) + \ddot{v}_i(t)h + \dots \quad (1.18)$$

Podstawiając $\ddot{v}_i(t) = \frac{\dot{v}_i(t+h) - \dot{v}_i(t)}{h}$ do równania (1.17) otrzymujemy ostatecznie:

$$v_i(t+h) \approx v_i(t) + \frac{F_i(t+h) + F_i(t)}{2m_i}h. \quad (1.19)$$

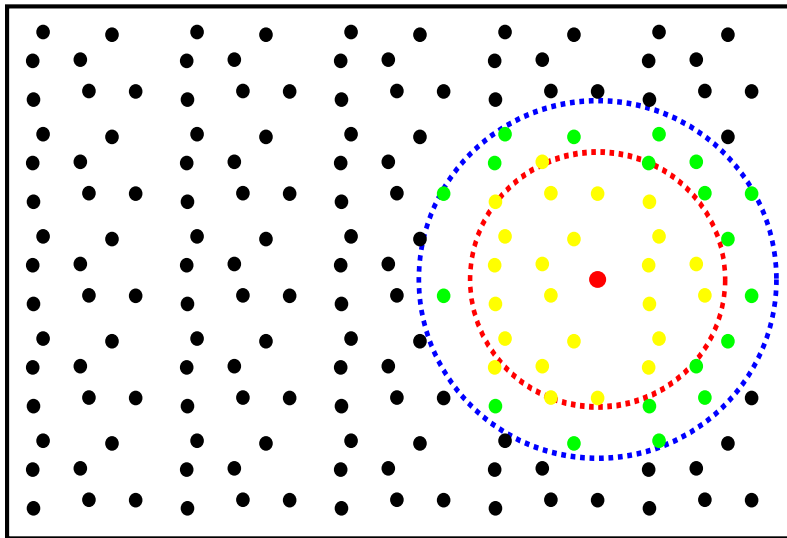
Równania (1.16) oraz (1.19) są podstawą algorytmu Verleta w postaci prędkościowej.

1.1.2. Oddziaływania krótkozasięgowe

W przypadku, gdy w badanym układzie występują jedynie oddziaływania krótkozasięgowe wskazane jest uwzględnienie tego faktu w programie. W tym celu stosuje się dwie techniki: (a) listy Verleta; (b) listę cel.

Listy Verleta

Każdej z N cząstek w układzie przyporządkowuje się dwuwymiarową tablicę $A(i, k)$, która zawiera informację o liczbie cząstek znajdujących się w odległości mniejszej od r_o od cząstki i oraz ich numerach. W symulacjach komputerowych każdej cząstce można nadać unikalny numer. Parametr $r_o > r_c$ określa maksymalną odległość, w której poszukiwani są sąsiedzi cząstki i ; zaś r_c – wartość promienia odcięcia, powyżej której potencjał $U(r)$ przyjmuje wartość zero. Brak warunku $r_o > r_c$ powoduje, że symulacje są niestabilne. Dodatkowo tablica $A(i, j)$ powinna być okresowo aktualizowana, tak aby stale zawierała prawidłowe wartości. Działanie list Verleta zostało przedstawione na Rys. 1.1.



Rysunek 1.1. Lista Verleta: cząstka czerwona oddziałuje tylko z cząstkami żółtymi.

Listy cel

Inną konstrukcją stosowaną w celu przyspieszenia obliczeń są listy cel. Ich zasada działania jest następująca (Rys. 1.2.):

1. pudełko symulacyjne dzielone jest na mniejsze fragmenty (cele) o liniowych rozmiarach r'_o ;
2. cząstki z celi (n, l) , gdzie n oraz l oznaczają odpowiednio kolumnę i wiersz, oddziałują z cząstkami znajdującymi się w tej samej celi oraz w celach sąsiednich, czyli $(n \pm 1, l \pm 1)$.

1.1.3. Periodyczne warunki brzegowe

W ramach eksperymentu komputerowego możliwe (na chwilę obecną) jest badanie układu $N \approx 10^3 \div 10^6$ atomów. Liczba ta jest znacznie mniejsza niż, charakterystyczna

Algorytm 1 Wyznaczania list Verleta

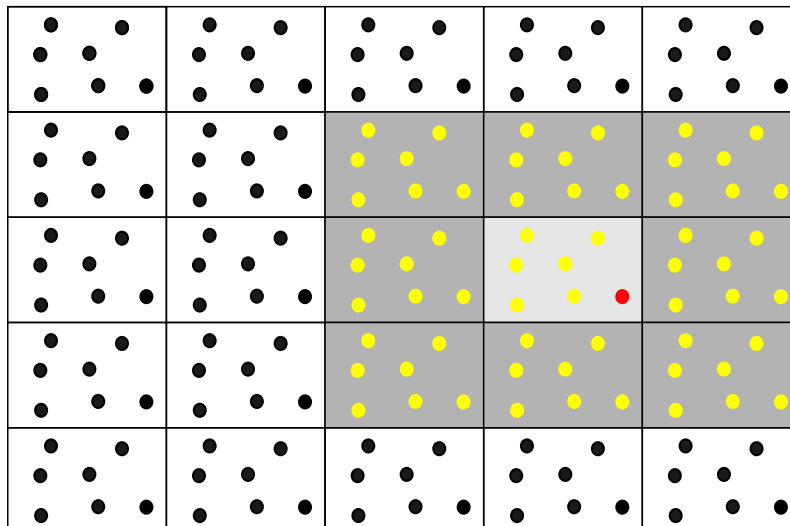
```

1: procedure VERLET-LIST( $\vec{R}$ ,  $r_o$ )
2:    $n \leftarrow \text{length}[\vec{R}]$  ▷ Liczba cząstek
3:   for  $i \leftarrow 1$  to  $n$  do
4:      $L_{i,0} \leftarrow 0$  ▷ zerowanie listy najbliższych sąsiadów
5:   end for

6:   for  $i \leftarrow 1$  to  $n - 1$  do
7:     for  $j \leftarrow i + 1$  to  $n$  do
8:        $|\vec{r}_{i,j}| \leftarrow |\vec{R}_i - \vec{R}_j|$  ▷ wyznaczanie odległości pomiędzy  $i$  a  $j$ 
9:       if  $|\vec{r}_{i,j}| < r_o$  then
10:         $k \leftarrow L_{i,0} + 1$  ▷ aktualizacja listy dla  $i$ 
11:         $L_{i,0} \leftarrow k$ 
12:         $L_{i,k} \leftarrow j$ 
13:         $k \leftarrow L_{j,0} + 1$  ▷ aktualizacja listy dla  $j$ 
14:         $L_{j,0} \leftarrow k$ 
15:         $L_{j,k} \leftarrow i$ 
16:      end if
17:    end for
18:  end for

19:  return  $L$ 
20: end procedure

```



Rysunek 1.2. Lista cel: cząstka czerwona oddziałuje tylko z cząstkami żółtymi ze swojej celi oraz cel sąsiednich.



Algorytm 2 Wyznaczania list cel

```
1: procedure CELL-LIST( $\vec{R}$ ,  $L_x$ ,  $L_y$ ,  $r_o$ )
2:    $n \leftarrow \text{length}[R]$  ▷ Liczba cząstek

3:    $N_x \leftarrow \lfloor L_x/r_o \rfloor$  ▷ Wyznaczanie maksymalnej liczby cel
4:    $N_y \leftarrow \lfloor L_y/r_o \rfloor$ 

5:    $r_{o,x} \leftarrow L_x/N_x$  ▷ Wielkość pojedynczej celi wzdłuż osi  $x$  i  $y$ 
6:    $r_{o,y} \leftarrow L_y/N_y$ 

7:   for  $i \leftarrow 1$  to  $n$  do ▷ zerowanie listy cel #1
8:      $C_i \leftarrow 0$ 
9:   end for

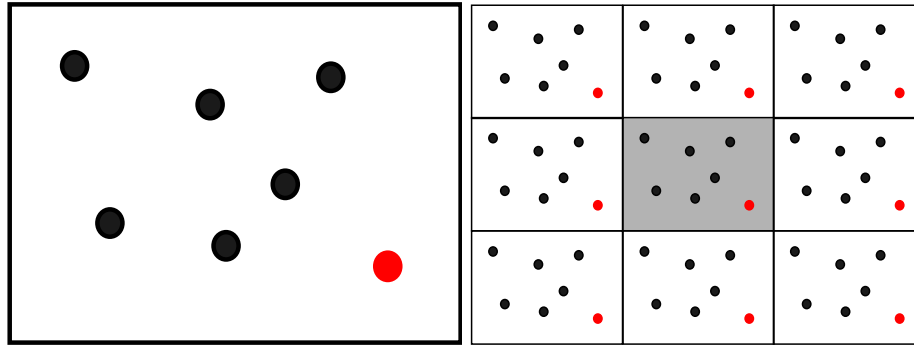
10:  for  $i \leftarrow 0$  to  $N_x N_y - 1$  do ▷ zerowanie listy cel #2
11:     $L_{i,0} \leftarrow 0$ 
12:  end for

13:  for  $i \leftarrow 1$  to  $n$  do
14:     $n_x \leftarrow \lfloor R_{i,x}/r_{o,x} \rfloor$  ▷ Wyznaczanie współrzędnych celi
15:     $n_y \leftarrow \lfloor R_{i,y}/r_{o,y} \rfloor$ 
16:     $j \leftarrow n_y L_x + n_x$  ▷ Uaktualnienie listy cel
17:     $k \leftarrow L_{j,0} + 1$ 
18:     $L_{j,k} \leftarrow i$ 
19:     $C_i \leftarrow j$ 
20:  end for

21:  return  $[L, C]$ 
22: end procedure
```

dla układów rzeczywistych, liczba Avogadro ($N_A \approx 6.02 \times 10^{23}$). W przypadku symulacji układów skończonych ważną rolę odgrywa brzeg układu.

Dla pewnych układów istnienie brzegu jest niepożądane. Można to rozwiązać stosując tzw. okresowe warunki brzegowe, których schematyczna reprezentacja przedstawiona jest na rys. 1.3.



Rysunek 1.3. Okresowe warunki brzegowe: (a) układ symulowany; (b) układ powielony.

W ramach symulacji można to sobie wyobrazić jako powielenie układu wyjściowego wzdłuż każdej z osi x , y .

W czasie symulacji może nastąpić taka sytuacja, że cząstka przekroczy granicę pomiędzy dwiema kopiami układu wyjściowego, np. po stronie lewej. W tym momencie po stronie prawej pojawi się jej kopia pochodząca z układu znajdującego się po prawej stronie.

1.1.4. Zasada minimalnego obrazu

W tym rozdziale skupimy się jak w sposób efektywny wyznaczyć oddziaływanie pomiędzy cząstkami.

Załóżmy, że oddziaływanie zależy tylko od położenia par cząstek: i -tej oraz j -tej. Formalnie energia potencjalna U_i związana z oddziaływaniem cząstki i -tej zależy od położenia wszystkich pozostałych $N - 1$ cząstek, czyli:

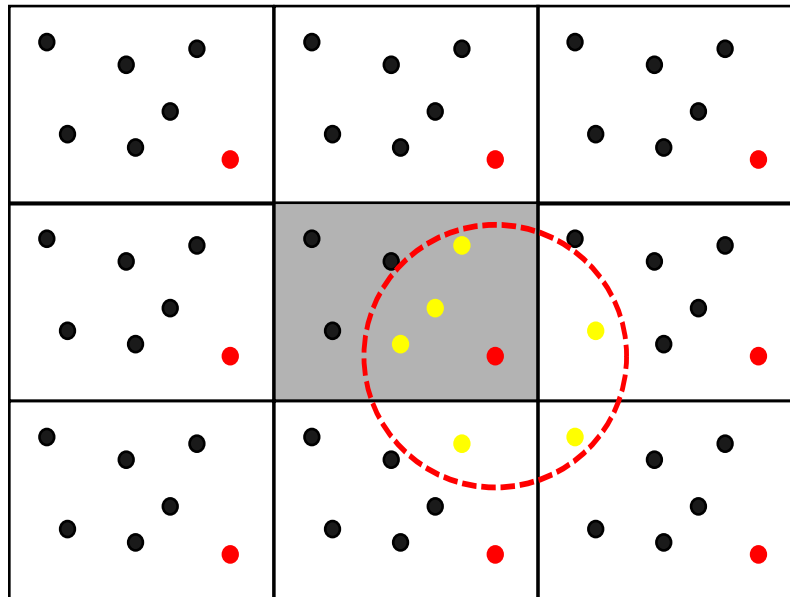
$$U_i = \sum_{j \neq i} U(\vec{r}_{ij}). \quad (1.20)$$

Uwzględniając powielenie układu wzdłuż każdej z osi (okresowe warunki brzegowe), wyznaczenie np. energii potencjalnej układu byłoby niemożliwe z uwagi na sumowanie po nieskończeniu wielu parach pomiędzy cząstką i -tą oraz cząstką j -tą i jej kopiami w układach powielonych.

Rozwiązaniem problemu, który został zasygnalizowany w poprzednim paragrafie, jest wykorzystanie zasady minimalnego obrazu (ang. *minimal image convention*). Według niej cząstka i -ta oddziałuje tylko z cząstkami znajdującymi się w pewnej odległości $r < r_c$, gdzie r_c oznacza wartość promienia odcięcia, powyżej której potencjał $U(r)$ przyjmuje wartość 0:

$$U_c(r) = \begin{cases} U(r) & \text{dla } r < r_c, \\ 0 & \text{dla } r \geq r_c. \end{cases} \quad (1.21)$$

Schematycznie jest to przedstawione na rys. 1.4.



Rysunek 1.4. Zasada minimalnego obrazu.

Dla tak zdefiniowanego potencjału, cząstka i -ta (czerwona) oddziałuje tylko i wyłącznie z tymi cząstkami, bądź ich kopiami z układów sąsiednich (żółte), które znajdują się w odległości $r < r_c$ od cząstki czerwonej.

1.2. Wielkości zredukowane

W przypadku programów do dynamiki molekularnej albo Monte Carlo należy zwrócić w szczególności uwagę na to, aby wyniki uzyskane tym sposobem były obarczone jak najmniejszym błędem. Jednym z ich źródeł mogą być błędy związane z zaokrągleniem liczb rzeczywistych przez procesor. O ile działania na dwóch liczb całkowitych nie niesie ze sobą zagrożenia (za wyjątkiem możliwości wystąpienia błędu przepełnienia), to operacje na liczbach rzeczywistych są bardziej podatne na błędy. Jest to związane z faktem, iż w komputerze wszystkie zmienne mają z góry określoną wielkość pamięci, np. 1, 2, 4, 8 czy 16 bajtów. Związany jest z tym ściśle zbiór możliwych i rozróżnialnych wartości liczb rzeczywistych, które można przechowywać w komputerze; a także precyzja (liczba cyfr znaczących) z jaką są one przechowywane. Więcej informacji na temat reprezentacji liczb rzeczywistych i ich arytmetyki jest dostępna w [3].

Algorytm używający liczb rzeczywistych będzie działał w sposób optymalny, gdy liczby, na których będzie operował, będą tego samego rzędu. Jest zatem stosowne, aby tak wybrać jednostki, aby wszelkie liczby były (w typowych przypadkach) rzędu 1.

1.2.1. Położenie, pole powierzchni oraz gęstość

Jeżeli w badanym układzie występuje pewna charakterystyczna skala wielkości, np. 1 \AA , to wygodnie jest przedstawiać położenie w tych jednostkach zamiast w obowiązujących w układzie SI metrach. Związek pomiędzy długością bezwymiarową oraz rzeczywistą jest następujący:

$$L' \equiv L/\lambda, \quad (1.22)$$

gdzie λ oznacza wybraną skalę wielkości.



Parametr λ definiuje także związek pomiędzy polem powierzchni S' , objętością V' oraz gęstością ρ' :

$$S' \equiv S/\lambda^2, \quad (1.23)$$

$$V' \equiv V/\lambda^3, \quad (1.24)$$

$$\rho' \equiv \lambda^3 \rho = \frac{N}{V'}, \quad (1.25)$$

gdzie wielkość ρ' nazywana jest „number density”.

1.2.2. Energia potencjalna

Energia potencjalna układu N oddziałujących cząstek jest opisana wzorem:

$$U(\{\vec{r}_i\}) = \sum_i u_1(\vec{r}_i) + \sum_i \sum_{i<j} u_2(\vec{r}_i, \vec{r}_j) + \sum_i \sum_{i<j} \sum_{i<j<k} u_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) + \dots, \quad (1.26)$$

gdzie u_1 opisuje oddziaływanie z polem zewnętrznym (np. polem elektrycznym $\vec{E}(\vec{r})$), u_2 – oddziaływanie dwucząstkowe, u_3 – oddziaływanie trójcząstkowe, itd. Zazwyczaj w równaniu (1.26) opuszcza się wzajemne oddziaływania więcej niż dwóch cząstek ($u_3 = u_4 = \dots = 0$):

$$U(\{\vec{r}_i\}) = \sum_i u_1(\vec{r}_i) + \sum_i \sum_{i<j} u_2(\vec{r}_i, \vec{r}_j). \quad (1.27)$$

W przypadku braku pola zewnętrznego wzór (1.27) ulega dalszemu uproszczeniu:

$$U(\{\vec{r}_i\}) = \sum_i \sum_{i<j} u_2(\vec{r}_i, \vec{r}_j). \quad (1.28)$$

Jeżeli w badanym układzie można zdefiniować pewną charakterystyczną wartość energii ϵ , np. 1 eV, to wygodnie jest używać wielkości bezwymiarowych. Wtedy energia potencjalna U w jednostkach ϵ przyjmie wartość:

$$U' \equiv U/\epsilon. \quad (1.29)$$

1.2.3. Czas

Do tej pory zdefiniowaliśmy takie wielkości bezwymiarowe jak położenie (L'), energię potencjalną (U') i energię kinetyczną (K'). W tym Rozdziale pokażemy jak wyprowadzić ogólne wyrażenie na czas bezwymiarowy.

Dla przypomnienia, w układzie SI długość λ mierzona jest w metrach (m), masa μ – kilogramach (kg) energia ϵ – dżulach ($J = N m = kg m^2 s^{-2}$). Te trzy wielkości zależą wyłącznie od jednostek podstawowych: metra, kilograma oraz sekundy. Czy istnieje zatem taka kombinacja λ , ϵ oraz μ , która w wyniku da nam wielkość τ o wymiarze czasu?

Rozpatrzmy zatem następującą wielkość:

$$[\lambda]^\alpha \cdot [\epsilon]^\beta \cdot [\mu]^\gamma = [\tau], \quad (1.30)$$

gdzie zapis $[x]$ oznacza jednostkę wielkości x . Zatem równanie (1.30) przyjmuje postać:

$$kg^{\alpha+\beta} \cdot m^{\gamma+2\alpha} \cdot s^{-2\alpha} = s^1. \quad (1.31)$$



Równanie (1.31) można przedstawić jako układ trzech równań z trzema niewiadomymi:

$$\alpha + \beta = 0, \quad (1.32)$$

$$\gamma + 2\alpha = 0, \quad (1.33)$$

$$-2\alpha = 1. \quad (1.34)$$

Rozwiązaniem równań (1.32)-(1.34) są: $\alpha = -\frac{1}{2}$, $\beta = \frac{1}{2}$, $\gamma = 1$. Podstawiając te wartości do wzoru (1.30) otrzymujemy definicję jednostki czasu jako funkcji położenia, masy oraz energii:

$$\tau = \sqrt{\frac{\mu\lambda^2}{\epsilon}}. \quad (1.35)$$

Tak zdefiniowany czas τ pozwoli nam wprowadzić czas bezwymiarowy t' :

$$t' \equiv t/\tau. \quad (1.36)$$

1.2.4. Energia kinetyczna

Energia kinetyczna K układu N cząstek związana jest z masą cząstek oraz ich prędkościami w następujący sposób:

$$K = \sum_i \frac{1}{2} m_i |\vec{v}_i|^2. \quad (1.37)$$

Wyprowadźmy teraz postać równania (1.37) w jednostkach bezwymiarowych. Z definicji prędkość jest określona jako:

$$\vec{v}_i = \frac{d\vec{r}_i}{dt} = \frac{d(\lambda\vec{r}'_i)}{d(\tau t')} = \frac{\lambda}{\tau} \frac{d\vec{r}'_i}{dt'} = \frac{\lambda}{\tau} \vec{v}'_i = \sqrt{\frac{\epsilon}{\mu}} \vec{v}'_i, \quad (1.38)$$

gdzie \vec{v}'_i jest prędkością bezwymiarową.

Podstawiając wyrażenie $\sqrt{\frac{\epsilon}{\mu}} \vec{v}'_i$ do wzoru (1.37) otrzymujemy

$$K = \sum_i \frac{1}{2} m_i |\vec{v}_i|^2 = \sum_i \frac{1}{2} m_i \left| \sqrt{\frac{\epsilon}{\mu}} \vec{v}'_i \right|^2 = \sum_i \frac{1}{2} m_i \frac{\epsilon}{\mu} |\vec{v}'_i|^2 = \epsilon \sum_i \frac{1}{2} \frac{m_i}{\mu} |\vec{v}'_i|^2 = \epsilon \sum_i \frac{1}{2} m'_i |\vec{v}'_i|^2, \quad (1.39)$$

gdzie m'_i jest masą w jednostkach μ . Równanie (1.39) można zapisać w skróconej formie jako:

$$K' \equiv K/\epsilon. \quad (1.40)$$

Powyższe równanie jest niczym innym jak definicją energii kinetycznej w jednostkach bezwymiarowych.

1.2.5. Temperatura

Temperatura układu jest powiązana, na podstawie zasady ekwipartycji energii, z energią kinetyczną układu. Związek ten ma postać:

$$\frac{d}{2} N_f k_B T = K, \quad (1.41)$$



gdzie d jest wymiarem przestrzeni, $k_B = 1.38 \cdot 10^{-23}$ J/K – stałą Boltzmanna, K – całkowitą energią kinetyczną, N_f – liczbą stopni swobody układu. Wobec tego:

$$T = \frac{2}{dN_f k_B} K = \frac{2}{dN_f k_B} \sum_i \frac{1}{2} m_i |\vec{r}_i^2| = \frac{1}{dN_f k_B} \sum_i m_i |\vec{r}_i^2|. \quad (1.42)$$

Korzystając z zależności (1.39) powyższy wzór możemy zapisać jako:

$$T = \frac{1}{dN_f k_B} \epsilon K = \frac{\epsilon}{k_B} \left(\frac{1}{dN_f} K' \right) = \frac{\epsilon}{k_B} T' \quad (1.43)$$

Podobnie jak w przypadku energii potencjalnej i kinetycznej wprowadza się pojęcie temperatury zredukowanej:

$$T' \equiv \frac{k_B T}{\epsilon}. \quad (1.44)$$

1.2.6. Ciśnienie

Celem zdefiniowania ciśnienia p' w jednostkach bezwymiarowych przeprowadzimy analogiczne rozumowanie jak w Rozdziale 1.2.3. Rozpatrzmy zatem następującą wielkość:

$$[\lambda]^\alpha \cdot [\epsilon]^\beta \cdot [\mu]^\gamma = [p_0]. \quad (1.45)$$

Zatem równanie (1.45) przyjmuje postać:

$$\text{kg}^{\beta+\gamma} \cdot \text{m}^{\alpha+2\beta} \cdot \text{s}^{-2\beta} = \text{kg}^1 \cdot \text{m}^{-1} \cdot \text{s}^{-2}. \quad (1.46)$$

Równanie (1.46) można przedstawić jako układ trzech równań z trzema niewiadomymi:

$$\beta + \gamma = 1, \quad (1.47)$$

$$\alpha + 2\beta = -1, \quad (1.48)$$

$$-2\beta = -2. \quad (1.49)$$

Rozwiązaniem równań (1.47)-(1.49) są: $\alpha = -3$, $\beta = 1$, $\gamma = 0$. Podstawiając te wartości do wzoru (1.45) otrzymujemy definicję jednostki ciśnienia jako funkcji położenia, masy oraz energii:

$$p_0 = \epsilon / \lambda^3. \quad (1.50)$$

Tak zdefiniowany ciśnienie p_0 pozwoli nam wprowadzić ciśnienie zredukowane p' :

$$p' \equiv \frac{\lambda^3}{\epsilon} p. \quad (1.51)$$

Wyznaczanie ciśnienia na podstawie wiriału

W Rozdziale 1.2.6. przedstawiliśmy jak wprowadzić pojęcie ciśnienia w jednostkach bezwymiarowych. Nic jednak nie było powiedziane jak wyznaczyć jego wartość. Zostanie to uczynione w tym Rozdziale.



Jedną z metod wyznaczenia ciśnienia jest skorzystanie z twierdzenia o wirale. W mechanice pojęcie wiriału wyprowadza się na przykład na podstawie równania (1.1):

$$\vec{F}_i(t) = m_i \frac{d^2 \vec{r}(t)}{dt^2} \quad (1.52)$$

$$\vec{F}_i(t) \vec{r}_i(t) = m_i \frac{d^2 \vec{r}(t)}{dt^2} \cdot \vec{r}(t) \quad (1.53)$$

$$= m_i \left[\frac{d}{dt} \left(\vec{r}_i(t) \frac{d\vec{r}_i(t)}{dt} \right) - \left(\frac{d\vec{r}_i(t)}{dt} \right)^2 \right] \quad (1.54)$$

$$= m_i \frac{d}{dt} \left(\vec{r}_i(t) \frac{d\vec{r}_i(t)}{dt} \right) - 2K_i \quad (1.55)$$

Jeżeli wysumujemy teraz równanie (1.55) po wszystkich N cząstkach oraz uśrednimy je po czasie to otrzymamy:

$$\left\langle \sum_i \vec{F}_i(t) \vec{r}_i(t) \right\rangle = \sum_{i=1}^N m_i \left[\frac{1}{\tau} \int_{t=0}^{\tau} \frac{d}{dt} \left(\vec{r}_i(t) \frac{d\vec{r}_i(t)}{dt} \right) dt \right] - 2\langle K \rangle \quad (1.56)$$

$$= \sum_{i=1}^N m_i \left[\frac{1}{\tau} \int_{t=0}^{\tau} \frac{dG(t)}{dt} dt \right] - 2\langle K \rangle \quad (1.57)$$

Całka po prawej stronie znaku równości wynosi zero, jeżeli spełniony jest warunek $|G(t)| < \infty$:

$$\lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{t=0}^{\tau} \frac{dG(t)}{dt} dt = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} G(t) \Big|_{t=0}^{\tau} = \lim_{\tau \rightarrow \infty} \frac{G(\tau) - G(0)}{\tau} = 0. \quad (1.58)$$

Ostatecznie więc:

$$\left\langle \sum_i \vec{F}_i(t) \vec{r}_i(t) \right\rangle = -2\langle K \rangle. \quad (1.59)$$

Wielkość $\left\langle \sum_i \vec{F}_i(t) \vec{r}_i(t) \right\rangle$ nazywana jest wiralem. $\vec{F}_i(t)$ jest całkowitą siłą działającą na i -tą cząstkę i może być zapisana jako suma sił wewnętrznych $F_i^{(w)}(t)$ oraz zewnętrznych $F_i^{(z)}(t)$. Tym samym wiral przyjmie postać:

$$\left\langle \sum_i \vec{F}_i^{(w)}(t) \vec{r}_i(t) \right\rangle + \left\langle \sum_i \vec{F}_i^{(z)}(t) \vec{r}_i(t) \right\rangle = -2\langle K \rangle. \quad (1.60)$$

W równaniu (1.60) nieznaną wartością jest zewnętrzna składowa wiriału:

$$\left\langle \sum_i \vec{F}_i^{(z)}(t) \vec{r}_i(t) \right\rangle = - \left\langle \sum_i \vec{F}_i^{(w)}(t) \vec{r}_i(t) \right\rangle - 2\langle K \rangle. \quad (1.61)$$

Jednak siłę zewnętrzną $\vec{F}_i^{(z)}(t)$ działającą na i -tą cząstkę można powiązać z ciśnieniem p :

$$\vec{F}_i^{(z)}(t) = -pd\vec{S}, \quad (1.62)$$

gdzie $d\vec{S}$ jest elementem powierzchni. Zatem równanie (1.61) przyjmie postać:

$$\left\langle \sum_i pd\vec{S}_i \vec{r}_i(t) \right\rangle = \left\langle \sum_i \vec{F}_i^{(w)}(t) \vec{r}_i(t) \right\rangle + 2\langle K \rangle. \quad (1.63)$$



Korzystając z twierdzenia Gaussa-Ostrogradskiego i całkując (1.63) po objętości otrzymujemy:

$$pV = N_f k_B T + \frac{1}{d} \left\langle \sum_i \vec{F}_i^{(w)}(t) \vec{r}_i(t) \right\rangle, \quad (1.64)$$

gdzie d oznacza wymiar przestrzeni. Równanie (1.64) zwane jest wirialnym równaniem stanu. Dla układu nieoddziałujących cząstek przechodzi ono w równanie gazu doskonałego $pV = Nk_B T$. Pamiętając, że $\vec{F}_i(t)$ jest całkowitą siłą działającą na i -tą cząstkę wyrażenie $\sum_i \vec{F}_i(t) \vec{r}_i(t)$ można zapisać jako:

$$\sum_i \vec{F}_i \vec{r}_i = \sum_i \sum_{j < i} \vec{F}_{ij} \vec{r}_i + \sum_i \sum_{j > i} \vec{F}_{ij} \vec{r}_i \quad (1.65)$$

$$= \sum_i \sum_{j < i} \vec{F}_{ij} \vec{r}_i - \sum_i \sum_{j > i} \vec{F}_{ji} \vec{r}_i, \quad (1.66)$$

gdzie dla uproszczenia zapisu $\vec{F}_i(t) \equiv \vec{F}_i$ oraz $\vec{r}_i(t) \equiv \vec{r}_i$. Wyrażenie $\sum_i \sum_{j > i} \langle \dots \rangle$ we wzorze (1.66) można przedstawić w równoważnej postaci:

$$\sum_i \sum_{j > i} \vec{F}_{ji} \vec{r}_i = \sum_i \sum_{i < j} \vec{F}_{ij} \vec{r}_j, \quad (1.67)$$

co po podstawieniu do wzoru (1.66) daje:

$$\sum_i \vec{F}_i \vec{r}_i = \sum_i \sum_{j < i} \vec{F}_{ij} (\vec{r}_i - \vec{r}_j) = \sum_i \sum_{j < i} \vec{F}_{ij} \vec{r}_{ij}. \quad (1.68)$$

Ostatecznie zatem równanie (1.61) przyjmie postać:

$$pV = N_f k_B T + \frac{1}{d} \left\langle \sum_i \sum_{i < j} \vec{F}_{ij}^{(w)}(t) \vec{r}_{ij}(t) \right\rangle. \quad (1.69)$$

1.3. Dwuwymiarowy układ Lennarda-Jonesa

1.3.1. Potencjał Lennarda-Jonesa

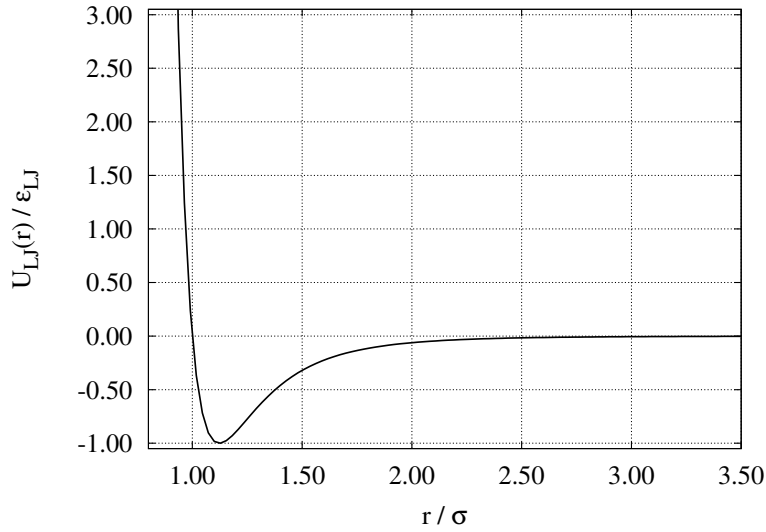
Potencjał Lennarda-Jonesa, stosowany głównie do modelowania gazów szlachetnych takich jak n.p. argon, ma postać:

$$U_{LJ}(r) = 4\epsilon_{LJ} \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (1.70)$$

gdzie $r \equiv |\vec{r}_{ij}| = |\vec{r}_i - \vec{r}_j|$ oznacza wektor wodzący pomiędzy i -tą cząstką a j -tą, ϵ_{LJ} – maksymalną głębokość potencjału, σ – „średnicę” atomów. Zależność potencjału Lennarda-Jonesa od odległości pomiędzy atomami jest przedstawiona na Rys. 1.5.

Ze względu na postać oddziaływania naturalny jest wybór takich jednostek, w których odległość jest liczona w jednostkach σ a energia – ϵ_{LJ} . Wobec tego, parametry λ oraz ϵ z Rozdziału 1.2. przyjmą następujące wartości – $\lambda \equiv \sigma$, $\epsilon \equiv \epsilon_{LJ}$ ¹.

¹Dla układu dwuwymiarowego wielkość ρ' jest zdefiniowana jako $\rho' \equiv \sigma^2 \rho$.



Rysunek 1.5. Potencjał Lennarda-Jonesa jako funkcja odległości pomiędzy atomami.

1.3.2. Bezwymiarowe równanie ruchu Newtona

Na cząstkę poruszającą się w potencjale $U(\vec{r})$, działa siła $\vec{F}(\vec{r})$ zdefiniowana jako:

$$\vec{F}(\vec{r}) = -\nabla_{\vec{r}}U(\vec{r}), \quad (1.71)$$

gdzie $\nabla_{\vec{r}}$ jest operatorem różniczkowym postaci:

$$\nabla_{\vec{r}} \equiv \frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j} + \frac{\partial}{\partial z} \hat{k} = \frac{\partial}{\partial \vec{r}}, \quad (1.72)$$

gdzie \hat{i} , \hat{j} , \hat{k} oznaczają, odpowiednio, wersory wzdłuż osi x , y oraz z .

Równania (1.1) oraz (1.71) pozwalają zatem napisać równanie ruchu Newtona:

$$\vec{F}(\vec{r}) = -\frac{dU(r)}{dr} \cdot \frac{\partial r}{\partial \vec{r}} = -\frac{dU(r)}{dr} \frac{\vec{r}}{r} = \frac{48\epsilon_{LJ}}{\sigma} \left[\left(\frac{\sigma}{r}\right)^{13} - \frac{1}{2} \left(\frac{\sigma}{r}\right)^7 \right] \frac{\vec{r}}{r} \quad (1.73)$$

$$= \frac{48\epsilon_{LJ}}{\sigma^2} \left[\left(\frac{\sigma}{r}\right)^{14} - \frac{1}{2} \left(\frac{\sigma}{r}\right)^8 \right] \vec{r} \quad (1.74)$$

Na podstawie równania (1.74) wyrażenia na siłę $\vec{F}(\vec{r}')$ oraz przyspieszenie \vec{a}' w postaci bezwymiarowej przyjmują postać:

$$\vec{F}' = \frac{48\epsilon_{LJ}}{\sigma^2} \left[\left(\frac{1}{r'}\right)^{14} - \frac{1}{2} \left(\frac{1}{r'}\right)^8 \right] \vec{r}', \quad (1.75)$$

$$\vec{a}' \equiv \frac{F'}{m} = \frac{48\epsilon_{LJ}}{m\sigma^2} \left[\left(\frac{1}{r'}\right)^{14} - \frac{1}{2} \left(\frac{1}{r'}\right)^8 \right] \vec{r}', \quad (1.76)$$

Stały czynnik $\frac{48\epsilon_{LJ}}{m\sigma^2}$ występujący w równaniu (1.76) służy do definicji jednostki czasu (patrz równania (1.12), (1.16)):

$$\tau \equiv \sqrt{\frac{m\sigma^2}{48\epsilon_{LJ}}}. \quad (1.77)$$

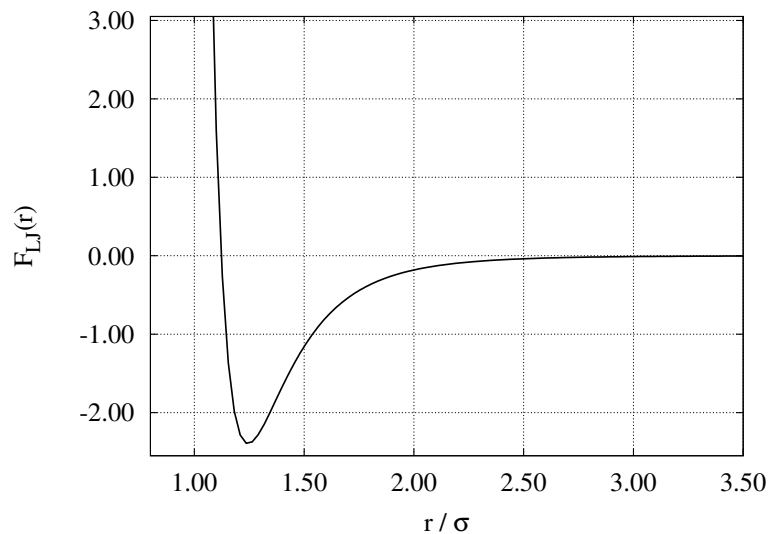
Porównując wzór (1.77) ze wzorem (1.35) można stwierdzić, że dla potencjału Lennarda-Jonesa stała ϵ (jednostka energii) z równanie (1.35) przyjmie wartość:

$$\epsilon \equiv 48\epsilon_{LJ}. \quad (1.78)$$

Definicja jednostki czasu τ powoduje, że zależność pomiędzy energią kinetyczną K (w jednostkach rzeczywistych) oraz K' (w jednostkach bezwymiarowych) ma postać (patrz równanie (1.39)):

$$K = \epsilon_{LJ} \sum_i 24m'_i |\vec{v}'_i|^2 = \epsilon_{LJ} K'. \quad (1.79)$$

Zależność siły $\vec{F}(\vec{r})$ od odległości pomiędzy atomami jest przedstawiona na Rys. 1.6.



Rysunek 1.6. Siła działająca na cząstkę w potencjale Lennarda-Jonesa jako funkcja odległości pomiędzy atomami.

1.4. Symulacje w układzie NVE

W Rozdziale tym przedstawimy techniczne zagadnienia związane z symulacjami metodą dynamiki molekularnej. W szczególności przedstawione zostaną podstawowe informacje na temat wyboru konfiguracji początkowej, metod skalowania prędkości (dla potrzeb równowagowania układu na początku symulacji) oraz modelowy program dynamiki molekularnej.

1.4.1. Warunki początkowe

Wybór warunków początkowych jest niezwykle ważny z punktu widzenia prowadzenia badań zarówno metodami dynamiki molekularnej jak i Monte Carlo.

Warunki początkowe dla każdej symulacji były następujące:

- atomy znajdowały się w węzłach sieci trójkątnej,
- składowe prędkości \vec{v}_i były losowane z rozkładu normalnego:

$$\vec{v}_i \propto \mathcal{N}(0, \sigma_v). \quad (1.80)$$

– całkowity pęd układu wynosił 0:

$$\sum_{i=1}^N m_i \vec{v}_i = 0. \quad (1.81)$$

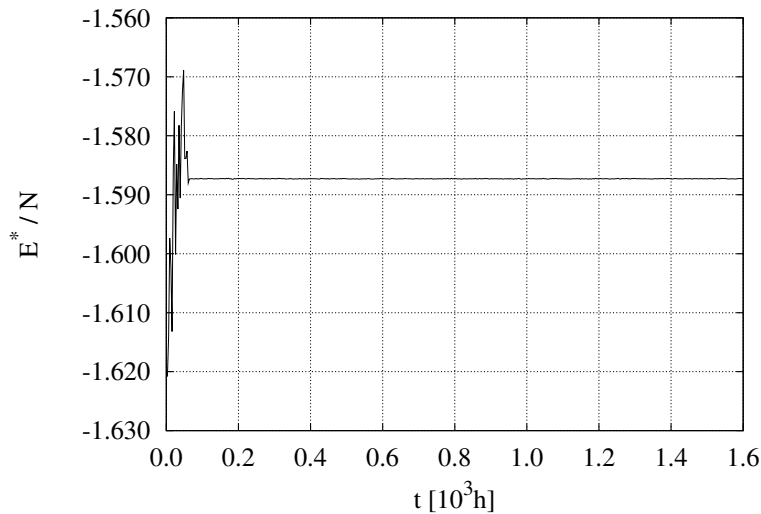
– prędkości \vec{v}_i były tak skalowane, aby temperatura T' układu odpowiadała zadanej temperaturze początkowej T_0^* :

$$\vec{v}_i \rightarrow \vec{v}_i \sqrt{T_0^*/T'}. \quad (1.82)$$

1.4.2. Skalowanie prędkości

Wybrane prędkości początkowe nie pochodzą, na ogół, z rozkładu Maxwella-Boltzmannna. Aby zatem pozwolić układowi dojść do stanu równowagi termodynamicznej na początku symulacji stosuje się jedną z technik skalowania prędkości. Aby przeprowadzać symulacje przy stałej temperaturze T^* można wykorzystać metody takie jak: (a) skalowanie prędkości; (b) termostat Hoovera; (c) termostat Andersena. Techniki te omówienie są w [1, 2]. W niniejszej pracy wykorzystano najprostszą technikę, polegającą na skalowaniu prędkości, gdyż przeprowadzane symulacje odbywały się z daleka od punktu potrójnego.

Na rys. 1.7. przedstawiono zależność energii kinetycznej od czasu dla układu $N = 4096$ cząstek w $T' = 0.70$ oraz $\rho' = 0.78$. Widoczne dla czasów $t > 64h$ linia prosta odpowiada symulacjom bez skalowania prędkości.



Rysunek 1.7. Zależność energii całkowitej E od czasu dla układu $N = 4096$ atomów w $T' = 0.70$ oraz $\rho' = 0.78$.

1.4.3. Modelowy program dynamiki molekularnej

W Rozdziale tym zaprezentujemy programy do symulacji metodą dynamiki molekularnej korzystające do rozwiązywania układu równań ruchu Newtona algorytmy: Newtona oraz Verleta w postaci położeniowej i prędkościowej.

Specjalnie zrezygnowano z użycia konkretnego języka programowania (np. C++) na rzecz pseudokodu, aby ułatwić Czytelnikowi skupienie się na problemie (tu: programie do symulacji) niż na jego konkretnej implementacji.



Algorytm Eulera

Przedstawiony poniżej program korzysta do rozwiązywania układu równań ruchu Newtona ze wzorów (1.8) i (1.9).

Algorytm 3 Modelowy program dynamiki molekularnej: algorytm Eulera

```

1: procedure MOLECULAR-DYNAMICS-EULER
2:    $(N, K, h, r_c, r_o) \leftarrow \text{READ-ENVIRONMENT}(\text{„MD.ini”})$ 
3:    $(\vec{R}, \vec{V}, m) \leftarrow \text{INITIAL-CONFIGURATION}(N, \text{„MD.dat”})$ 
4:    $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}, r_c, r_o)$ 

5:   for  $k \leftarrow 1$  to  $K$  do           ▷ Główna pętla programu: wyznaczanie kolejnych konfiguracji
6:      $t_k \leftarrow k h$ 

7:      $\vec{F} \leftarrow \text{FORCES}(\vec{R})$ 
8:      $\vec{R} \leftarrow \vec{R} + \vec{V} h$            ▷ Wyznaczenie nowych położeń (1.8)
9:      $\vec{V} \leftarrow \vec{V} + \frac{1}{m} \vec{F} h$        ▷ Wyznaczenie nowych prędkości (1.9)

10:    if isScaleVelocities = True then   ▷ Skalowanie prędkości na początku symulacji
11:       $\vec{V} \leftarrow \text{SCALE-VELOCITIES}(\vec{V})$ 
12:    end if

13:     $(T', p', E'_p, E'_k, E'_c) \leftarrow \text{THERMODYNAMICS}(\vec{R}, \vec{V})$ 
14:     $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}, r_c, r_o)$ 

15:    SAVE-CONFIGURATION( $\vec{R}, \vec{V}$ )
16:  end for

17: end procedure

```

Algorytm Verleta w postaci położeniowej

Przedstawiony poniżej program korzysta do rozwiązywania układu równań ruchu Newtona ze wzorów (1.13) i (1.14).

Algorytm Verleta w postaci prędkościowej

Przedstawiony poniżej program korzysta do rozwiązywania układu równań ruchu Newtona ze wzorów (1.16) i (1.19).

1.5. Zakończenie

Przedstawione w poprzednich Rozdziałach informacje na temat symulacji metodą dynamiki molekularnej stanowią jedynie podstawę do dalszego zgłębiania przez Czytelnika tej tematyki. Wspomniane we *Wstępie* pozycje literaturowe [1] oraz [2] są doskonałym kompendium wiedzy w tym zakresie.

Godnymi polecenia są także [4] i [5] przedstawiające typowe problemy oraz ich rozwiązania pojawiające się w przypadku obliczeń numerycznych. Pozycje [6] [7] są zaś źródłem informacji o statystyce i jej praktycznym zastosowaniu.



Algorytm 4 Modelowy program dynamiki molekularnej: algorytm Verleta w postaci położeniowej

```
1: procedure MOLECULAR-DYNAMICS-VERLET-R
2:    $(N, K, h, r_c, r_o) \leftarrow \text{READ-ENVIRONMENT}(\text{„MD.ini”})$ 
3:    $(\vec{R}_{-1}, \vec{R}_0, m) \leftarrow \text{INITIAL-CONFIGURATION}(N, \text{„MD.dat”})$ 
4:    $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}_0, r_c, r_o)$ 

5:   for  $k \leftarrow 1$  to  $K$  do           ▷ Główna pętla programu: wyznaczanie kolejnych konfiguracji
6:      $t_k \leftarrow k h$ 

7:      $\vec{F} \leftarrow \text{FORCES}(\vec{R}_0)$ 
8:      $\vec{R}_{+1} \leftarrow \vec{R}_{-1} + 2\vec{R}_0 + \frac{\vec{F}}{m} h^2$            ▷ Wyznaczenie nowych położenia (1.13)
9:      $\vec{V}_0 \leftarrow \frac{1}{2h} (\vec{R}_{+1} - \vec{R}_{-1})$            ▷ Wyznaczenie nowych prędkości (1.14)

10:    if isScaleVelocities = True then           ▷ Skalowanie prędkości na początku symulacji
11:       $\vec{V} \leftarrow \text{SCALE-VELOCITIES}(\vec{V})$ 
12:    end if

13:     $(T', p', E'_p, E'_k, E'_c) \leftarrow \text{THERMODYNAMICS}(\vec{R}_0, \vec{V})$ 
14:     $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}_0, r_c, r_o)$ 

15:    SAVE-CONFIGURATION( $\vec{R}_0, \vec{V}$ )

16:     $(\vec{R}_{-1}, \vec{R}_0) \leftarrow (\vec{R}_0, \vec{R}_{+1})$ 
17:  end for

18: end procedure
```



Algorytm 5 Modelowy program dynamiki molekularnej: algorytm Verleta w postaci prędkościowej

```
1: procedure MOLECULAR-DYNAMICS-VERLET-V
2:    $(N, K, h, r_c, r_o) \leftarrow \text{READ-ENVIRONMENT}(\text{„MD.ini”})$ 
3:    $(\vec{R}, \vec{V}, m) \leftarrow \text{INITIAL-CONFIGURATION}(N, \text{„MD.dat”})$ 
4:    $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}, r_c, r_o)$ 
5:    $\vec{F} \leftarrow \text{FORCES}(\vec{R})$ 

6:   for  $k \leftarrow 1$  to  $K$  do           ▷ Główna pętla programu: wyznaczanie kolejnych konfiguracji
7:      $t_k \leftarrow k h$ 

8:     if isScaleVelocities = True then   ▷ Skalowanie prędkości na początku symulacji
9:        $\vec{V} \leftarrow \text{SCALE-VELOCITIES}(\vec{V})$ 
10:    end if

11:     $\vec{R} \leftarrow \vec{R} + \vec{V} h + \frac{1}{2} \frac{\vec{F}}{m} h^2$            ▷ Wyznaczenie nowych położeń (1.16)
12:     $\vec{V} \leftarrow \vec{V} + \frac{1}{2} \frac{\vec{F}}{m} h$            ▷ Wyznaczenie nowych prędkości (1.19)

13:     $\vec{F} \leftarrow \text{FORCES}(\vec{R})$ 

14:     $\vec{V} \leftarrow \vec{V} + \frac{1}{2} \frac{\vec{F}}{m} h$            ▷ Wyznaczenie nowych prędkości (1.19)

15:     $(T', p', E'_p, E'_k, E'_c) \leftarrow \text{THERMODYNAMICS}(\vec{R}, \vec{V})$ 
16:     $L \leftarrow \text{UPDATE-VERLET-LIST}(\vec{R}, r_c, r_o)$ 
17:     $\text{SAVE-CONFIGURATION}(\vec{R}, \vec{V})$ 
18:  end for

19: end procedure
```



Zadania do wykonania

1. Wykonać symulacje układu $N = 1024, 1600, 2500$ cząstek dla temperatury zredukowanej $T' = 0.80$. Nanieść na jeden wykres zależność energii potencjalnej na cząstkę (U'/N) od czasu. Jako oddziaływanie przyjąć potencjał Lennarda-Jonesa.
2. Dla ustalonej temperatury zredukowanej $T' = 0.70$ znaleźć obszary odpowiadające (a) kryształowi, (b) obszarowi dwufazowemu, (c) cieczy.
3. Na podstawie zależności energii całkowitej E' znaleźć położenie równowagowe atomów dla przypadku najgęstszego upakowania w dwóch wymiarach (sieć trójkątna).
4. Dla wybranej temperatury i gęstości zbadać stabilność numeryczną wybranego algorytmu całkowania równań ruchu Newtona od (a) wartości parametru h , (b) wyboru wartości parametrów r_o i r_c .



Bibliografia

- [1] D. Frenkel, B. Smit; *Understanding Molecular Simulation. From Algorithms to Application*, 2nd ed.; Academic Press, 2002.
- [2] M. P. Allen, D. J. Tildesley; *Computer Simulation of Liquids*; Clarendon Press, Oxford 1987.
- [3] *754-2008 IEEE Standard for Floating-Point Arithmetic*, ISBN 978-0-7381-5753-5, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4610935
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery; *Numerical Recipes in Fortran 77. The Art of Scientific Computing*, 2nd ed.; Cambridge University Press.
- [5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery; *Numerical Recipes in Fortran 90. The Art of Parallel Scientific Computing*, 2nd ed.; Cambridge University Press.
- [6] R. Nowak; *Statystyka dla fizyków*; Wydawnictwo Naukowe PWN, Warszawa 2002.
- [7] *R: A Language and Environment for Statistical Computing*, R Development Core Team, R Foundation for Statistical Computing, Wiedeń, Austria 2006, ISBN 3-900051-07-0.



TA STRONA JEST INTENCJONALNIE PUSTA



Rozdział 2.

Metoda Monte Carlo (MC)

Opis termodynamicznych i strukturalnych własności układów z hamiltonianami modelowymi uzyskuje się stosując metody mechaniki statystycznej. Podejście czysto analityczne napotyka na duże trudności zarówno natury technicznej jak i koncepcyjnej, dotyczącej zakresu stosowalności używanych przybliżeń.

Metody symulacji komputerowych, w szczególności metoda Monte Carlo [1–5] oraz metoda dynamiki molekularnej [1], oferują inne, komplementarne podejście do badania termodynamiki i struktury układów wielu cząstek. W niniejszym opracowaniu zajmujemy się szczególną realizacją ogólnej strategii Monte Carlo, polegającej na estymowaniu parametrów pewnej hipotetycznej populacji na podstawie próby tej populacji uzyskanej przy użyciu losowej sekwencji liczb [4].

2.1. Statystyczny opis układu

W niniejszym opracowaniu rozpatrzmy jedynie układ w zespole kanonicznym [6]. Oznaczmy różne możliwe mikroskopowe konfiguracje (mikrostan) układu indeksem a . Zbiór mikrostanów może być, zależnie od układu, albo ciągły, albo dyskretny. Niech E_a oznacza energię mikrostanu oznaczonego indeksem a . Prawdopodobieństwo p_a , że układ znajduje się w mikro stanie a jest równe

$$p_a = \frac{1}{Z} e^{-\frac{E_a}{k_B T}}, \quad (2.1)$$

gdzie k_B oznacza stałą Boltzmanna, T temperaturę bezwzględną, a

$$Z = \sum_a e^{-\frac{E_a}{k_B T}} \quad (2.2)$$

nosi nazwę sumy statystycznej. Wzór (2.1) przedstawia rozkład Gibbsa prawdopodobieństwa. Możemy dzięki niemu obliczyć średnią wartość po zespole kanonicznym \bar{X} dowolnej wielkości X związanej z układem pod warunkiem, że:

- znamy mikrostan a układu;
- znamy energię E_a każdego mikrostanu;
- znamy wartość X_a , którą przybiera X w każdym z mikrostanów a ;
- potrafimy wykonać sumowanie

$$\bar{X} = \sum_a p_a X_a = \frac{1}{Z} \sum_a X_a e^{-\beta E_a}. \quad (2.3)$$



2.2. Metoda Monte Carlo - koncepcje

Obliczenie sumy (2.3) jest z reguły niemożliwe. Metoda Monte Carlo polega na generowaniu konfiguracji (stanów a) i obliczaniu dla nich wartości interesujących nas wielkości X . Następnie obliczamy średnią \bar{X} po wygenerowanych N konfiguracjach:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i \quad (2.4)$$

Nie jest to dokładnie procedura omówiona w części 2.1. Zamiast wszystkich możliwych mikrostanów a mamy zbiór mikrostanów częściej lub rzadziej występujących. Do generowania mikrostanów używa się procesu Markowa. Proces Markowa jest regułą zawierającą element losowy, który pozwala otrzymać nową konfigurację układu z aktualnie istniejącej. Jest ważne, że ta reguła uwzględnia tylko obecny stan układu i nie wymaga wiedzy o stanach poprzednich. Jest ona wyrażona przez zadanie zbioru prawdopodobieństw: z każdą możliwą parą stanów a i a' związane jest prawdopodobieństwo $P(a \rightarrow a')$ przejścia ze stanu a do stanu a' w następnym kroku. Te prawdopodobieństwa podlegają ograniczeniu, wyrażającemu fakt, że układ zawsze przechodzi do jakiegoś stanu:

$$\sum_{a'} P(a \rightarrow a') = 1. \quad (2.5)$$

Metoda MC generuje łańcuch Markowa - sekwencję stanów generowanych przez proces Markowa. Prawdopodobieństwa znajdowania się układu w mikrostanach p_a opisuje równanie master:

$$\frac{dp_a(t)}{dt} = - \sum_{a'} P(a \rightarrow a') p_a(t) + \sum_{a'} P(a' \rightarrow a) p_{a'}(t). \quad (2.6)$$

Aby częstotliwości pojawiania się każdego mikrostanu a zmierzały asymptotycznie do rozkładu stacjonarnego (związanego z rozwiązaniem stacjonarnym równania Master), związanego z gibbsowskim rozkładem prawdopodobieństwa, muszą być spełnione następujące warunki związane z prawdopodobieństwami przejścia $P(a \rightarrow a')$.

1. Dla danego punktu początkowego musi istnieć możliwość osiągnięcia przez układ dowolnej innej konfiguracji. Żądanie to jest często zwane zasadą dostępności.
2. Prawdopodobieństwa przejścia muszą spełniać warunek

$$\sum_{a'} p_a P(a \rightarrow a') = \sum_{a'} P(a' \rightarrow a) p_{a'}, \quad (2.7)$$

lub tzw. warunek równowagi szczegółowej (mocniejszy od poprzedniego – niekonieczny, ale wystarczający):

$$p_a P(a \rightarrow a') = p_{a'} P(a' \rightarrow a). \quad (2.8)$$

2.2.1. Algorytm Metropolis

Jednym z algorytmów realizujących metodę Monte Carlo dla zespołu kanonicznego jest algorytm Metropolis [7]:

1. określenie konfiguracji początkowej a
2. wygenerowanie nowej konfiguracji a'



3. obliczenie różnicy energii $E_{a'} - E_a$
4. jeśli $E_{a'} - E_a < 0$ to nowa konfiguracja zostaje dołączona do łańcucha Markowa; powrót do kroku 2
5. obliczenie $w = \exp[-(E_{a'} - E_a)/k_B T]$
6. wygenerowanie liczby losowej R z przedziału $[0, 1]$
7. jeżeli $R \leq \exp[-(E_{a'} - E_a)/k_B T]$ to nowa konfiguracja (a') jest dołączana do łańcucha Markowa; powrót do kroku 2
8. w przeciwnym wypadku stara konfiguracja (a) jest dołączana do łańcucha Markowa; powrót do kroku 2

Widać stąd, że zawsze akceptujemy nową konfigurację, jeśli jej energia jest mniejsza lub równa energii poprzedniej konfiguracji. Nową konfigurację, która zwiększa energię układu, akceptujemy tylko z pewnym prawdopodobieństwem. W ten sposób minimalizowana jest nie energia, lecz inny potencjał termodynamiczny, np. energia swobodna.

2.3. Dwuwymiarowy model Isinga

2.3.1. Opis modelu

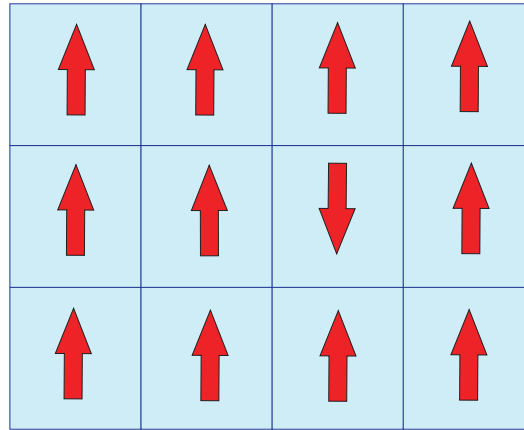
Model został zaproponowany przez W. Lenza w 1920 roku jako prosty model ferromagnetyka. Atrakcyjność modelu jest związana z prostotą jego opisu i obserwowanym w modelowym układzie przejściem fazowym. Analityczne rozwiązania znane są dla wybranych przypadków: modelu jednowymiarowego (E. Ising, 1925), modelu dwuwymiarowego bez pola (L. Onsager, 1944) i modelu dwuwymiarowego z polem (A. B. Zamolodchikov, 1989).

Każdemu węzłowi dwuwymiarowej sieci (spinowi) przyporządkowana jest liczba s , która może być równa $+1$ lub -1 (rys. 2.1.). Hamiltonian układu ma postać

$$H = \frac{1}{2} J \sum_{\langle ij \rangle} s_i s_j + h \sum_i s_i, \quad (2.9)$$

gdzie indeksy numerują węzły sieci (pierwsze sumowanie przebiega po wszystkich parach sąsiadujących ze sobą węzłów), h jest natężeniem zewnętrznego, jednorodnego pola. Będziemy zajmować się symulacją układu bez pola. Dla $J > 0$ w niskich temperaturach preferowane będą (ze względu na niższą energię) te same wartości s w sąsiednich węzłach – mamy wtedy do czynienia z układem ferromagnetycznym. Dla $J < 0$ mamy układ antyferromagnetyczny, z preferowanymi różnymi wartościami s w sąsiednich węzłach. Układ w niskich temperaturach wykazuje tendencję do uporządkowania (w zależności od wyboru J – ferromagnetycznego lub antyferromagnetycznego). Wraz ze wzrostem temperatury układ będzie tracił uporządkowanie. Po przekroczeniu pewnej temperatury T_c ($T_c k_B / J = 2 / \ln(1 + \sqrt{2})$) układ znajdzie się w fazie paramagnetycznej, w której te same i przeciwne wartości s w sąsiednich węzłach występują z równym prawdopodobieństwem.

Przedstawmy krótko wielkości fizyczne charakteryzujące nasz układ.



Rysunek 2.1. Schemat układu modelowego. Na kwadratowej sieci umieszczono strzałki skierowane w górę (dla $s = +1$) lub w dół (dla $s = -1$).

- **Namagnesowanie** $M = -\frac{\partial}{\partial B} \left(\frac{F}{k_B T} \right)$, gdzie F oznacza energię swobodną Helmholtza, obliczane (na jeden węzeł) jako suma wszystkich wartości s układu podzielona przez liczbę węzłów sieci:

$$m = \frac{1}{N} \sum_{i=1}^N s_i, \quad (2.10)$$

gdzie indeks i numeruje wszystkie węzły sieci (spiny), $N = L \times L$ - liczba spinów w układzie.

- **Podatność magnetyczna** (na jeden węzeł) obliczana z wariancji magnetyzacji:

$$\chi = \frac{1}{N k_B T} (\langle M^2 \rangle - \langle M \rangle^2), \quad (2.11)$$

gdzie $M = \sum_{i=1}^N s_i$ (czyli $M = Nm$), lub

$$\chi = \frac{N}{k_B T} (\langle m^2 \rangle - \langle m \rangle^2). \quad (2.12)$$

- **Pojemność cieplna** $C = \frac{\partial E}{\partial T}$ obliczana (na jeden węzeł) z fluktuacji energii układu:

$$C = \frac{1}{N k_B T^2} (\langle E^2 \rangle - \langle E \rangle^2). \quad (2.13)$$

- **Wykładniki krytyczne** dla długości korelacji ξ , magnetyzacji, podatności magnetycznej i pojemności cieplnej zdefiniowane są w następujący sposób:

$$\xi \propto |t|^{-\nu}, \quad (2.14)$$

$$\langle m \rangle \propto |t|^\beta, \quad (2.15)$$

$$\chi \propto |t|^{-\gamma}, \quad (2.16)$$

$$C \propto |t|^{-\alpha}, \quad (2.17)$$

gdzie $t = 1 - \frac{T}{T_c}$. Dla dwuwymiarowego modelu Isinga wykładniki mają następujące wartości: $\nu = 1$, $\beta = 1/8$, $\gamma = 7/4$, $\alpha = 0$.



- **Kumulant Bindera** czwartego stopnia związany z parametrem porządku zdefiniowany jest następującym równaniem

$$U_L = 1 - \frac{\langle M^4 \rangle_L}{3 \langle M^2 \rangle_L^2}. \quad (2.18)$$

Wielkość ta pozwala na wyznaczenie temperatury krytycznej bez znajomości wartości wykładników krytycznych.

2.3.2. Symulacja

Rozpatrywany układ składa się z dwuwymiarowej, kwadratowej sieci o rozmiarach $L \times L$. Zatem informację o przypisanej każdemu węzłowi liczbie s możemy przechowywać w dwuwymiarowej tablicy $S(L,L)$. Energia oddziaływania między sąsiednimi węzłami opisana jest przez hamiltonian (2.9). Stosujemy okresowe warunki brzegowe, tzn. węzły z ostatniej kolumny układu sąsiadują z odpowiednimi węzłami z pierwszej kolumny. Analogicznie węzły z ostatniego wiersza sąsiadują z odpowiednimi węzłami z wiersza pierwszego. W tej sytuacji w celu odwołania się do najbliższych sąsiadów danego węzła wygodnie jest zastosować tzw. tablicę najbliższych sąsiadów.

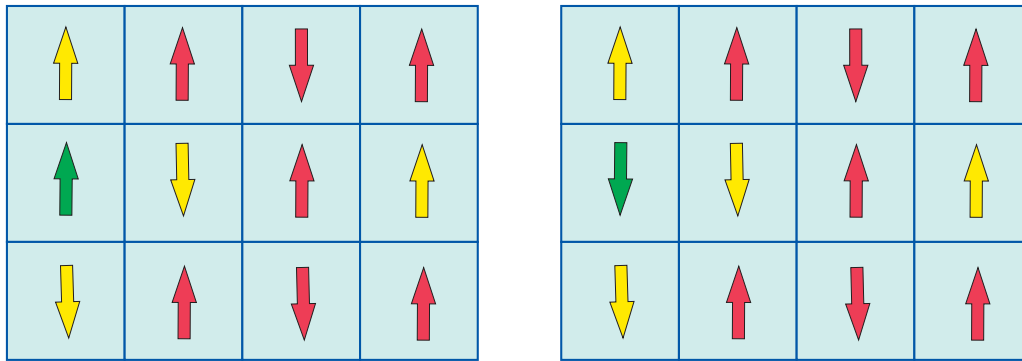
```
DO I=1,L
  NI(I)=I+1
  PI(I)=I-1
ENDDO NI(L)=1 PI(1)=L
```

Teraz, żeby odwołać się do liczby s sąsiada, np. z prawej strony węzła o współrzędnych (I, J) , wystarczy posłużyć się naszą tablicą i napisać $S(NI(I),J)$.

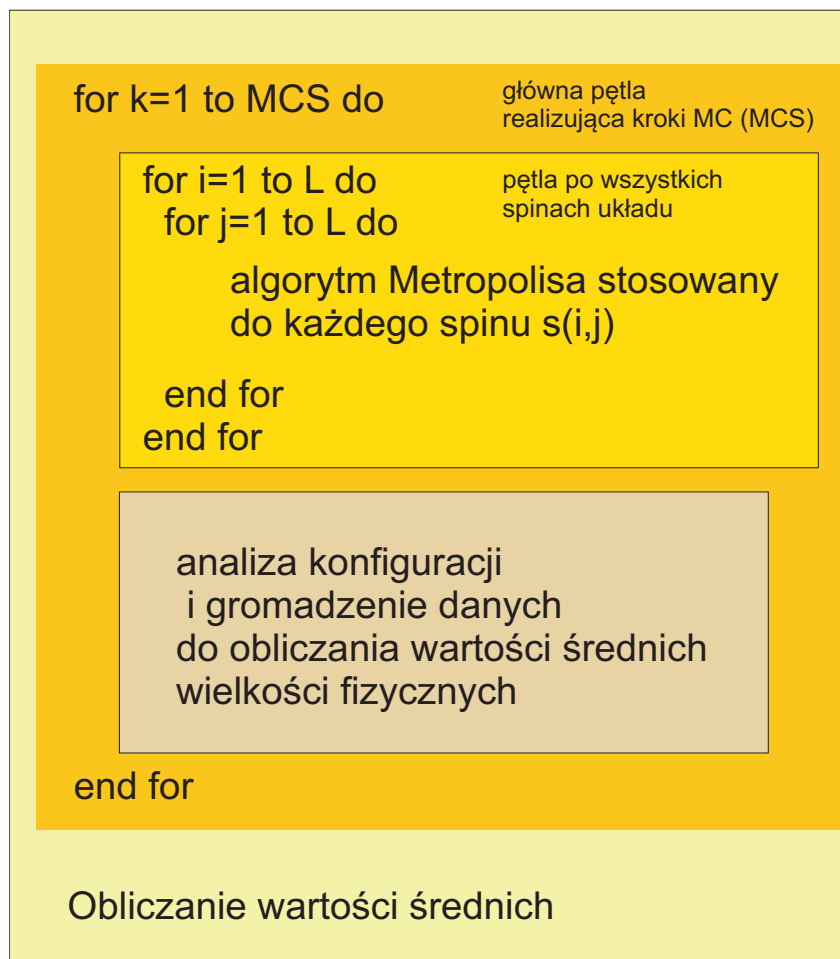
Do generowania konfiguracji układu w zespole kanonicznym stosować będziemy algorytm Metropolis'a. Po wybraniu konfiguracji początkowej (np. losowej – przypisujemy każdemu węzłowi liczbę s równą, z prawdopodobieństwem $1/2$, $+1$ lub -1) postępujemy wg. następującego schematu:

1. wygeneruj próbną konfigurację (za $S(I, J)$ dla wybranego węzła podstaw $-S(I, J)$) (rys. 2.2.);
2. oblicz zmianę energii ΔE ;
3. oblicz $w = \min\{1, e^{-\Delta E/(k_B T)}\}$
4. wylosuj R z rozkładu jednostajnego na przedziale $[0,1]$;
5. jeżeli $R \leq w$, to zaakceptuj jako nową konfigurację konfigurację próbną, w przeciwnym razie weź jako nową konfigurację konfigurację starą.

Tę procedurę należy przeprowadzić dla każdego węzła sieci. Schemat programu przedstawiono na Rys. 2.3. Za każdym razem dostajemy nową, kolejną konfigurację, która różni się, bądź nie, od starej (wyjściowej). Przeprowadzenie tej procedury dla kompletu $N = L \times L$ węzłów nazwiemy krokiem MC (MCS). Propozycja nowej konfiguracji polega na zmianie stanu danego węzła na przeciwny, dlatego zmiana energii ΔE ($\Delta E = -2 * S(I, J) * [S(I + 1, J) + S(I - 1, J) + S(I, J + 1) + S(I, J - 1)]$) towarzysząca takiej operacji może przyjmować tylko 5 wartości: $-8J, -4J, 0, 4J, 8J$ – w zależności od stanów najbliższych sąsiadów. Zamiast obliczać za każdym razem czynnik Boltzmann'a $e^{-\Delta E/(k_B T)}$ warto umieścić w tablicy jego wartości dla każdej możliwej kombinacji stanów naszego węzła i czterech najbliższych sąsiadów. W obliczeniach stosujemy wielkości



Rysunek 2.2. Schemat generowania próbnej konfiguracji, różniącej się od konfiguracji wyjściowej (poprzedniej) orientacją jednego, wybranego spinu (zaznaczonego na zielono). Kolorem żółtym zaznaczono najbliższych sąsiadów wybranego spinu. Zastosowano okresowe warunki brzegowe. Po lewej – konfiguracja wyjściowa, po prawej – próbna.



Rysunek 2.3. Schemat programu komputerowego do symulacji układu spinów.

zredukowane $T^* = Tk_B/J$ i $E^* = E/J$. Parametrami programu będzie rozmiar liniowy układu L i temperatura T^* . Wartość temperatury krytycznej dwuwymiarowego modelu



Isinga jest znana i wynosi $T_c^* = k_B T_c / J = 2.269\dots$

2.3.3. Obliczanie wartości średnich

W celu obliczenia wartości średniej interesującej nas wielkości fizycznej musimy dysponować odpowiednią liczbą konfiguracji układu znajdującego się w stanie równowagi termodynamicznej (przykładowe konfiguracje pokazano na rys. 2.4.). Konfiguracje na początku symulacji nie reprezentują na ogół układu w równowadze, bo nie reprezentuje go konfiguracja początkowa. Pewna liczba kroków MC jest potrzebna do uzyskania stanu równowagi (tzw. termalizacji układu). Konfiguracje uzyskane w toku termalizacji układu nie nadają się do obliczania wartości średnich. Ponadto kolejne konfiguracje są generowane na podstawie poprzednich, prze co są do nich podobne. Do obliczania wartości średnich potrzebne są konfiguracje nieskorelowane, co w praktyce osiąga się przez branie pod uwagę nie wszystkich konfiguracji, ale np. co 100 kroków MC.

Do zilustrowania przejścia fazowego w naszym układzie zbadamy w pierwszej kolejności zależność parametru porządku, tj. namagnesowania m , od temperatury (zbadanie zależności temperaturowej pojemności cieplnej i podatności magnetycznej pozostawiono czytelnikowi jako ćwiczenie). Żeby obliczyć średnią (po zespole) wartość namagnesowania $\langle m \rangle$ musimy obliczyć wartości m dla pewnej liczby konfiguracji, a następnie obliczyć średnią arytmetyczną. Na rys. 2.5. przedstawiono zmiany namagnesowania m w toku symulacji dla temperatury poniżej T_c . Widać tu, że mamy do czynienia z namagnesowaniem spontanicznym dwukrotnie zdegenerowanym (układ z tym samym prawdopodobieństwem jest uporządkowany ze stanami $s = +1$ (wtedy $m = 1$) i $s = -1$ ($m = -1$)). Aby, uśredniając m po takich konfiguracjach układu uporządkowanego, otrzymana wartość $\langle m \rangle$ odzwierciedlała uporządkowanie, musimy brać do średniej wartości bezwzględne m :

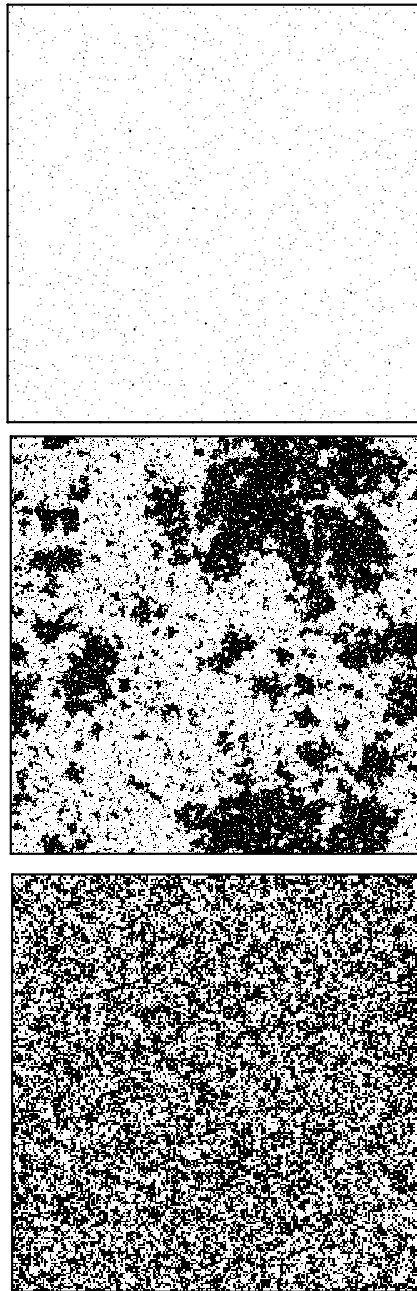
$$\langle m \rangle = \frac{1}{k} \sum_k |m_k|, \quad (2.19)$$

gdzie k numeruje konfiguracje układu, które uwzględniamy przy obliczaniu średniej. Na rys. 2.6. pokazano zależność od temperatury (w jednostkach zredukowanych) średniego namagnesowania dla różnych rozmiarów układu.

Symulacja układów skończonych, gdzie mamy do czynienia z ograniczeniem zasięgu korelacji do rozmiaru liniowego układu, prowadzi do wyników, które nie odzwierciedlają dokładnie własności układów makroskopowych. Jeśli długość korelacji charakterystyczna dla układu modelowego jest większa od rozmiaru zastosowanego w symulacjach, to wyniki otrzymane z symulacji będą zależeć od rozmiaru. Widać to na wykresie 2.6. (i wykresach sporządzonych przez czytelnika w ramach ćwiczenia), gdzie w pobliżu punktu krytycznego otrzymane krzywe mocno zależą od rozmiaru. Efekt rozmiaru ma także wpływ na podatność magnetyczną i pojemność cieplną. W granicy termodynamicznej podatność magnetyczna gwałtownie rośnie w temperaturze bliskiej T_c . Takie narastanie jest, ze względu na skończoną długość korelacji, ograniczone w układach o skończonych rozmiarach. Generowane za pomocą symulacji zależności mogą jednak dążyć do pojedynczej krzywej, gdyż wartości wielkości fizycznych w pobliżu punktu krytycznego ulegają skalowaniu. Skalowanie magnetyzacji z rozmiarem układu przebiega w następujący sposób:

$$m L^{\beta/\nu} = f[(T - T_c/T_c) L^{1/\nu}]. \quad (2.20)$$

Przykład efektu rozmiaru ze skalowaniem właśnie dla magnetyzacji przedstawiono na rys. 2.7. Używając znanych wartości odpowiednich wykładników krytycznych dla dwuwymiarowego modelu Isinga ($\beta = 1/8$, $\nu = 1$), przeskalowano wartości magnetyzacji i otrzymano – w pobliżu punktu krytycznego – jedną krzywą.

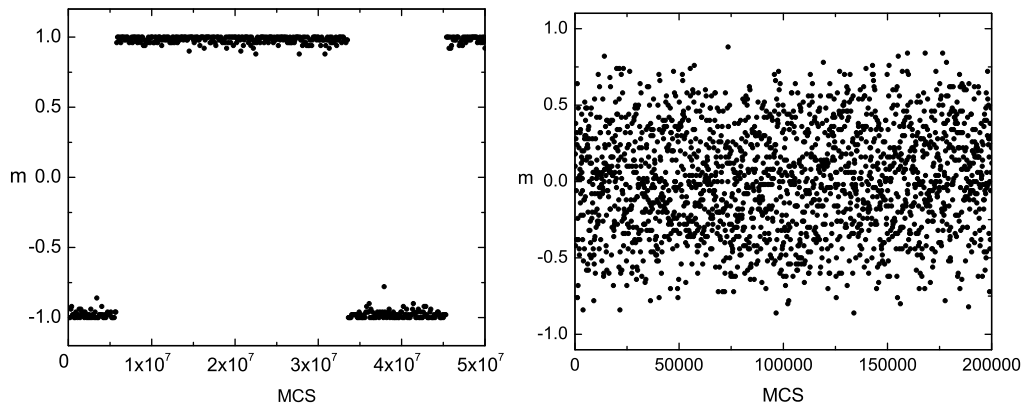


Rysunek 2.4. Przykładowe konfiguracje pochodzące z symulacji MC układu 500×500 węzłów – dla niskiej temperatury (u góry), dla temperatury bliskiej wartości krytycznej (w środku), dla wysokiej temperatury (na dole). Czarnym kolorem oznaczono węzły sieci dla których $s = +1$, białym $s = -1$. Typowe rozmiary czarnych obszarów są miarą długości korelacji ξ w układzie.

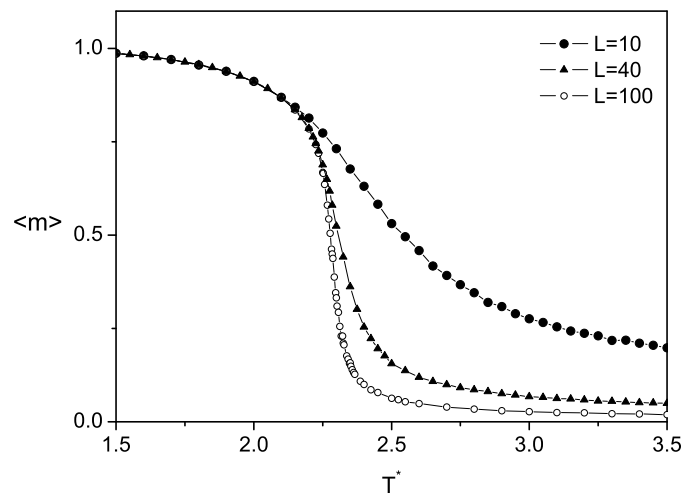
Do wyznaczania temperatury krytycznej wykorzystuje się własności kumulantów czwartego stopnia (2.18). Wartości kumulantów U_L dla różnych rozmiarów L układu są równe w punkcie przemiany fazowej.

$$\left. \frac{U_{L_1}}{U_{L_2}} \right|_{T=T_c} = 1 \quad (2.21)$$

Na rys. 2.8. pokazano przecięcie krzywych zależności $U_L(T)$ obliczonych dla układów o róż-

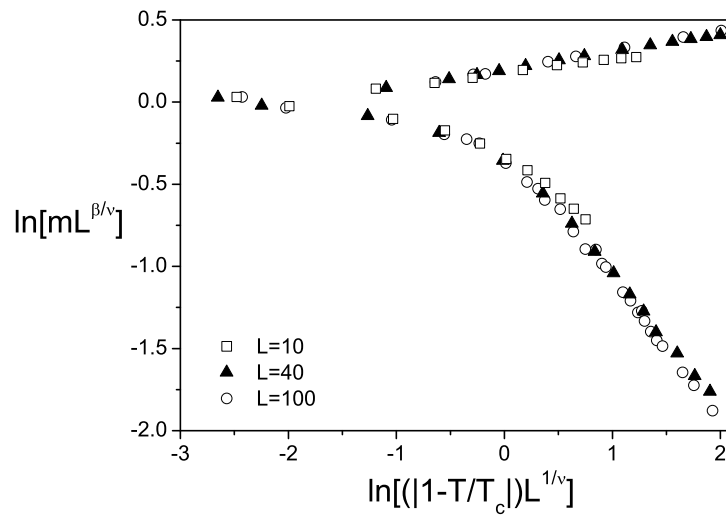


Rysunek 2.5. Namagnesowanie dla kolejnych kroków symulacji, dla układu $N = 10 \times 10$ węzłów i temperatury $T^* = 1.7$ (a) i $T^* = 3.0$ (b).

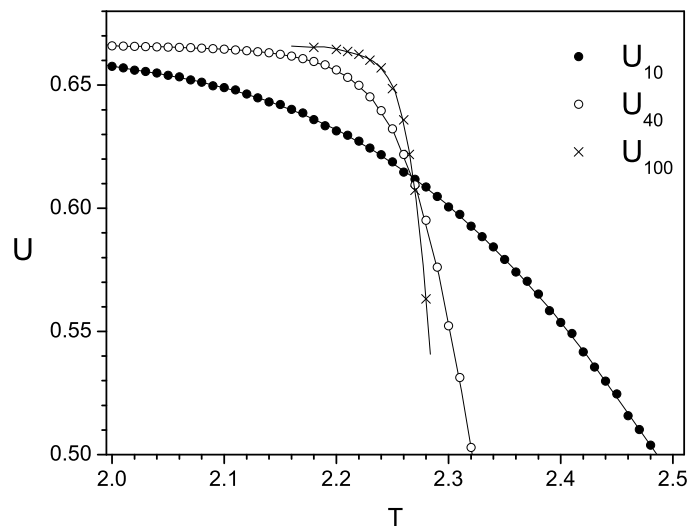


Rysunek 2.6. Zależność średniego namagnesowania od temperatury dla różnych rozmiarów układu. Symulowano $5.5 \cdot 10^5$ kroków MC.

nych rozmiarach. Widać tam, że temperatura przecięcia $T \approx T_c$.



Rysunek 2.7. Przeskalowana zgodnie ze schematem z równania (2.20) zależność średniego namagnesowania od temperatury dla różnych rozmiarów układu. Widać tu, że punkty otrzymane dla różnych rozmiarów układu, w pobliżu punktu krytycznego, układają się na tej samej krzywej. Symulowano $5.5 \cdot 10^5$ kroków MC.



Rysunek 2.8. Zależność kumulantów Bindera od temperatury dla różnych rozmiarów układu. Symulowano $5.5 \cdot 10^5$ kroków MC.



2.3.4. Zadania do wykonania w ramach laboratorium

1. Obserwacja układu i jego parametrów w toku symulacji MC
 - Zaobserwować przejścia układu między stanami metastabilnymi ze średnią magnetyzacją $+1$ i -1 .
 - Zaobserwować konfiguracje typowe dla temperatur: niskiej, wysokiej i w pobliżu T_c .
 - Zaobserwować zachowanie wybranych parametrów układu podczas termalizacji.
2. Obliczanie metodą MC średnich wartości wybranych wielkości fizycznych
 - Zbadać zależność magnetyzacji, podatności magnetycznej, energii i pojemności cieplnej od temperatury, dla różnych rozmiarów układu.
 - Zbadać zależność wariancji pojemności cieplnej od temperatury (np. metodą bootstrap).
3. Sprawdzenie teorii skalowania
 - Zastosować teorię skalowania do wyników symulacji (przyjąć wartości wykładników krytycznych $\beta = 0.125$, $\nu = 1$).
4. Wyznaczanie temperatury krytycznej
 - Wyznaczyć temperaturę krytyczną stosując metodę kumulantów Bindera.



TA STRONA JEST INTENCJONALNIE PUSTA



Bibliografia

- [1] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford (1987).
- [2] M. E. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Clarendon Press, Oxford (1999).
- [3] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, Cambridge (2000).
- [4] D. W. Hermann, *Computer Simulation Methods in Theoretical Physics*, Springer-Verlag Berlin Heidelberg (1986).
- [5] A. Patrykiewicz, *Wprowadzenie do metody Monte Carlo*, Wydawnictwo UMCS, Lublin (1998).
- [6] K. Huang, *Mechanika Statystyczna*, PWN, Warszawa (1978).
- [7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, J. Chem Phys. **21**, 1087 (1953).



TA STRONA JEST INTENCJONALNIE PUSTA



Rozdział 3.

Rozwiązywanie równań Maxwella metodą FDTD

3.1. Wstęp

Metoda FDTD jest algorytmem numerycznym używanym do rozwiązywania równań Maxwella uwzględniającym zależność pola elektromagnetycznego obserwowanego w dowolnym ośrodku od współrzędnych przestrzennych i czasu. Po raz pierwszy dyskretyzację równań Maxwella i metodę ich rozwiązywania wprost, zaprezentował Yee [1] w 1966 roku. Zaproponowany wówczas algorytm, jest rdzeniem wszystkich obecnie używanych metod numerycznych tej klasy.

W 1975 roku pomysł Yee został przypomniany i uzupełniony przez Taflove'a [2] o kryteria stabilności dla przypadków jedno-, dwu- i trójwymiarowego. Obecnie używaną powszechnie nazwę *Finite Difference Time Domain* algorytm otrzymał w 1980 roku [3]. Od tego czasu, dzięki systematycznemu rozwijaniu, poszerzyły się znacznie jego możliwości. Liczne prace przyczyniły się do dokładnego zbadania właściwości numerycznych i stałego zmniejszania niedokładności oraz coraz wierniejszego modelowania oddziaływania fal elektromagnetycznych z różnymi materiałami.

Podstawowymi zaletami metody FDTD są: jej proste wynikanie z równań Maxwella oraz wysoka dokładność i dobrze zrozumiane źródła błędów. Ponadto FDTD, jako algorytm działający w przestrzeni czasowej, w naturalny sposób traktuje wszelkie procesy związane z włączaniem sygnału oraz nieliniowe odpowiedzi badanych układów. Prócz tego, jedynymi częściami algorytmu określającymi analizowaną strukturę są siatka i parametry materiałowe określone na niej, zatem przejście do innej geometrii problemu nie wymaga, żadnej – potencjalnie złożonej – ingerencji w równania opisujące układ.

Celem opracowania jest zapoznanie czytelnika z podstawami metody skończonych przyrostów czasowych FDTD (Finite Differences Time Domain) [4,5] i umożliwienie wykonania pierwszych własnych obliczeń. Kolejne paragrafy mają na celu przedstawić podstawowy algorytm Yee, numeryczne źródła promieniowania, proste warunki brzegowe oraz sposób modelowania absorpcji. Integralną częścią opracowania są kody programów napisane dla środowiska obliczeniowego MATLAB. W opracowaniu wykorzystane są fragmenty pracy magisterskiej autora [6], która powstała pod opieką dra hab. inż. Włodzimierza Salejdy, prof. PWr. Niniejsze opracowanie może być wykorzystane w realizacji ćwiczenia dydaktycznego.



3.2. Równania Maxwella

W celu wyprowadzenia algorytmu Yee należy rozpatrzeć równania Maxwella, które opisują zjawiska klasycznej elektrodynamiki [7–10]:

$$\begin{aligned}\frac{\partial \vec{B}}{\partial t} &= -\nabla \times \vec{E} - \vec{K}, \\ \frac{\partial \vec{D}}{\partial t} &= \nabla \times \vec{H} - \vec{J}, \\ \nabla \cdot \vec{D} &= \rho_e, \\ \nabla \cdot \vec{B} &= \rho_m,\end{aligned}\tag{3.1}$$

gdzie:

- \vec{E} – natężenie pola elektrycznego [V/m],
- \vec{D} – indukcja elektryczna [C/m²],
- \vec{H} – natężenie pola magnetycznego [A/m],
- \vec{B} – indukcja magnetyczna [Wb/m²],
- \vec{J} – gęstość prądu elektrycznego [A/m²],
- \vec{K} – gęstość prądu magnetycznego [V/m²],
- ρ_e – gęstość ładunku elektrycznego [C/m³],
- ρ_m – gęstość ładunku magnetycznego [Wb/m³].

Podana powyżej postać równań obiega od najczęściej prezentowanej w podręcznikach. Uzasadnieniem rozbieżności jest chęć uzyskania pełnej symetrii między właściwościami elektrycznymi oraz magnetycznymi.

Równanie $\nabla \cdot \vec{D} = \rho_e$ wyraża prawo Gaussa dla pola elektrycznego w postaci różniczkowej. W tym wyrażeniu ρ_e jest gęstością ładunku elektrycznego. Prawo Gaussa dla pola magnetycznego – $\nabla \cdot \vec{B} = 0$ – wyraża nieistnienie ładunku magnetycznego. Żadne dotychczasowe doświadczenie nie wskazuje, aby ładunek magnetyczny istniał, jednakże wprowadzenie ładunku magnetycznego rozumianego jako magnetycznego analogu ładunku elektrycznego pozwala zapisać prawo Gaussa dla pola magnetycznego w postaci, która w pełni odpowiada prawu Gaussa dla pola elektrycznego. Podobnie w prawie Ampera ($\frac{\partial \vec{D}}{\partial t} = \nabla \times \vec{H} - \vec{J}$), występuje wielkość J – gęstość prądu elektrycznego. Jest ona związana z ładunkami elektrycznymi w ruchu. W tradycyjnym zapisie prawa Faradaya brakuje analogicznego członu ($\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}$). Jednakże wprowadzając ładunek magnetyczny musimy wprowadzić także gęstość prądu magnetycznego rozumianego jako przepływ hipotetycznych ładunków magnetycznych.

W przypadku metody FDTD ten formalny zabieg, oprócz uzyskania ładnej postaci równań, ułatwia uwzględnienie zależności względnych przenikalności elektrycznej i magnetycznej od częstotliwości.

Uzupełnieniem równań Maxwella są związki między wektorami indukcji, a odpowiednimi składowymi pola nazywane równaniami materiałowymi¹

$$\begin{aligned}\vec{D} &= \epsilon_r \epsilon_0 \vec{E}, \\ \vec{B} &= \mu_r \mu_0 \vec{H},\end{aligned}\tag{3.2}$$

gdzie:

- $\epsilon_0 = 8.854 \cdot 10^{-12}$ [F/m] – przenikalność elektryczna próżni,

¹Równania te są prawdziwe dla ośrodków jednorodnych i niedyspersyjnych.



- ε_r – względna przenikalność elektryczna ośrodka,
- $\mu_0 = 4\pi \cdot 10^{-7}$ [H/m] – przenikalność magnetyczna próżni,
- μ_r – względna przenikalność magnetyczna ośrodka.

W równaniach tych możemy zastosować przybliżenie pochodnej funkcji ilorazem różnicowym [11]. Przybliżenia tego dokonuje się na siatce Yee, która jest kluczowym elementem algorytmu numerycznego.

3.2.1. Równania Maxwella – trzy wymiary

Równania Maxwella w przestrzeni pozbawionej ładunków przyjmują dla trzech wymiarów postać

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} - \vec{K}, \quad (3.3)$$

$$\frac{\partial \vec{D}}{\partial t} = \nabla \times \vec{H} - \vec{J},$$

$$\nabla \cdot \vec{D} = 0, \quad (3.4)$$

$$\nabla \cdot \vec{B} = 0.$$

Wielkości \vec{J} i \vec{K} mogą być traktowane jako niezależne źródła energii pola elektromagnetycznego, odpowiednio dla składowych \vec{E} i \vec{H} . Jednocześnie możemy uwzględnić absorpcję energii pól i jej zamianę na ciepło poprzez równania:

$$\begin{aligned} \vec{J} &= \vec{J}_z + \sigma_e \vec{E}, \\ \vec{K} &= \vec{K}_z + \sigma_h \vec{H}, \end{aligned} \quad (3.5)$$

gdzie:

- σ_e – przewodność elektryczna [S/m],
- σ_h – równoważnik strat magnetycznych [Ω /m],
- \vec{J}_z – gęstość prądu elektrycznego źródła,
- \vec{K}_z – gęstość prądu magnetycznego źródła.

Po uwzględnieniu równań (3.5) oraz zależności materiałowych (3.2) pierwsze dwa równania (3.3) rozpisane na poszczególne składowe przestrzenne przyjmują (w przypadku materiałów liniowych, izotropowych, niedyspersyjnych i absorbujących) postać:

$$\begin{aligned} \frac{\partial H_x}{\partial t} &= \frac{1}{\mu} \left[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - (K_{z_x} + \sigma_h H_x) \right], \\ \frac{\partial H_y}{\partial t} &= \frac{1}{\mu} \left[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - (K_{z_y} + \sigma_h H_y) \right], \end{aligned} \quad (3.6)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - (K_{z_z} + \sigma_h H_z) \right];$$

$$\begin{aligned} \frac{\partial E_x}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - (J_{z_x} + \sigma_e E_x) \right], \\ \frac{\partial E_y}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - (J_{z_y} + \sigma_e E_y) \right], \\ \frac{\partial E_z}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - (J_{z_z} + \sigma_e E_z) \right]. \end{aligned} \quad (3.7)$$

Układ sześciu sprzężonych cząstkowych równań różniczkowych stanowi podstawę algorytmu FDTD. Nie zakłada on wprost relacji opisanych prawami Gaussa. Rozmieszczenie

przestrzenne wektorów na siatce związane z dyskretyzacją równań musi gwarantować spełnienie praw Gaussa.

3.2.2. Równania Maxwella – dwa wymiary

Zakładając niezmienniczość układu w kierunku osi z , redukujemy równania (3.6) i (3.7) do postaci, w której otrzymujemy dwa niezależne od siebie, układy trzech równań sprzężonych. Jeden z nich odpowiada sytuacji, w której składowe pola magnetycznego są prostopadłe do wyróżnionego kierunku (mod TM_z):

$$\begin{aligned}\frac{\partial H_x}{\partial t} &= \frac{1}{\mu} \left[-\frac{\partial E_z}{\partial y} - (K_{zx} + \sigma_h H_x) \right], \\ \frac{\partial H_y}{\partial t} &= \frac{1}{\mu} \left[\frac{\partial E_z}{\partial x} - (K_{zy} + \sigma_h H_y) \right], \\ \frac{\partial E_z}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - (J_{zz} + \sigma_e E_z) \right].\end{aligned}\quad (3.8)$$

W wypadku modu TE_z prostopadłe do osi z jest pole elektryczne i równania mają następującą postać:

$$\begin{aligned}\frac{\partial E_x}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_z}{\partial y} - (J_{zx} + \sigma_e E_x) \right], \\ \frac{\partial E_y}{\partial t} &= \frac{1}{\varepsilon} \left[-\frac{\partial H_z}{\partial x} - (J_{zy} + \sigma_e E_y) \right], \\ \frac{\partial H_z}{\partial t} &= \frac{1}{\mu} \left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - (K_{zz} + \sigma_h H_z) \right].\end{aligned}\quad (3.9)$$

3.2.3. Równania Maxwella – jeden wymiar

Kontynuując upraszczanie przez założenie niezmienniczości w kierunku osi y otrzymujemy z równania (3.8) dla modu TM_z , układ dwu sprzężonych równań postaci:²

$$\begin{aligned}\frac{\partial H_y}{\partial t} &= \frac{1}{\mu} \left[\frac{\partial E_z}{\partial x} - (K_{zy} + \sigma_h H_y) \right], \\ \frac{\partial E_z}{\partial t} &= \frac{1}{\varepsilon} \left[\frac{\partial H_y}{\partial x} - (J_{zz} + \sigma_e E_z) \right].\end{aligned}\quad (3.10)$$

Analogicznie można postąpić z modem TE_z , co prowadzi do układu równań sprzężonych:³

$$\begin{aligned}\frac{\partial E_y}{\partial t} &= \frac{1}{\varepsilon} \left[-\frac{\partial H_z}{\partial x} - (J_{zy} + \sigma_h E_y) \right], \\ \frac{\partial H_z}{\partial t} &= \frac{1}{\mu} \left[-\frac{\partial E_y}{\partial x} - (K_{zz} + \sigma_e H_z) \right].\end{aligned}\quad (3.11)$$

3.3. Dyskretyzacja pochodnych na siatce Yee

Powyżej przedstawione równania Maxwella są równaniami różniczkowymi cząstkowymi pierwszego stopnia. W celu ich rozwiązania numerycznego musimy operator różniczkowania zastąpić odpowiednim ilorazem różnicowym.

²Dodatkowo zakładamy, że $H_x = 0$ dla $t = 0$ oraz stale $K_{zx} = 0$.

³Dodatkowo zakładamy, że $E_x = 0$ dla $t = 0$ oraz stale $J_{zx} = 0$.



Wielkości fizyczne rozpatrywane tutaj są funkcjami zmiennych przestrzennych (x, y, z) oraz zmiennej czasowej (t) . Wartości będziemy rozpatrywać na dyskretnej siatce przestrzennej⁴ o kroku Δ_s i w dyskretnych chwilach, które dzieli czas Δ_t . Punktem początkowym siatki jest $\vec{r}_0 = (0, 0, 0)$, natomiast chwilą początkową $t_0 = 0$.

Dla uproszczenia przyjmuje się następujące oznaczenia:

- punkt w przestrzeni

$$\vec{r}_{i,j,k} = (i \cdot \Delta_s, j \cdot \Delta_s, k \cdot \Delta_s), \quad (3.12)$$

- chwila czasu

$$t_n = n \cdot \Delta_t, \quad (3.13)$$

- wartość $u(\vec{r}, t)$ w chwili t_n w punkcie $\vec{r}_{i,j,k}$ będziemy zapisywać w postaci

$$u(\vec{r}_{i,j,k}, t_n) = u_{i,j,k}^n. \quad (3.14)$$

W dalszej części opracowania stwierdzenie „chwila n ,” należy rozumieć jako chwila t_n ; natomiast „punkt (i, j, k) ,” jako punkt $\vec{r}_{i,j,k}$.

Przyjęto także konwencję, w której indeksy dolne oznaczają współrzędne przestrzenne, natomiast indeks górny oznacza czas. Stosując te oznaczenia można zapisać następujące wyrażenia:

- dla pochodnych cząstkowych względem zmiennych przestrzennych:

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_{i,j,k}^n &= \frac{u_{i+1/2,j,k}^n - u_{i-1/2,j,k}^n}{\Delta_s} + O[(\Delta_s)^2], \\ \left. \frac{\partial u}{\partial y} \right|_{i,j,k}^n &= \frac{u_{i,j+1/2,k}^n - u_{i,j-1/2,k}^n}{\Delta_s} + O[(\Delta_s)^2], \\ \left. \frac{\partial u}{\partial z} \right|_{i,j,k}^n &= \frac{u_{i,j,k+1/2}^n - u_{i,j,k-1/2}^n}{\Delta_s} + O[(\Delta_s)^2], \end{aligned} \quad (3.15)$$

- dla pochodnej cząstkowej względem czasu

$$\left. \frac{\partial u}{\partial t} \right|_{i,j,k}^n = \frac{u_{i,j,k}^{n+1/2} - u_{i,j,k}^{n-1/2}}{\Delta_t} + O[(\Delta_t)^2], \quad (3.16)$$

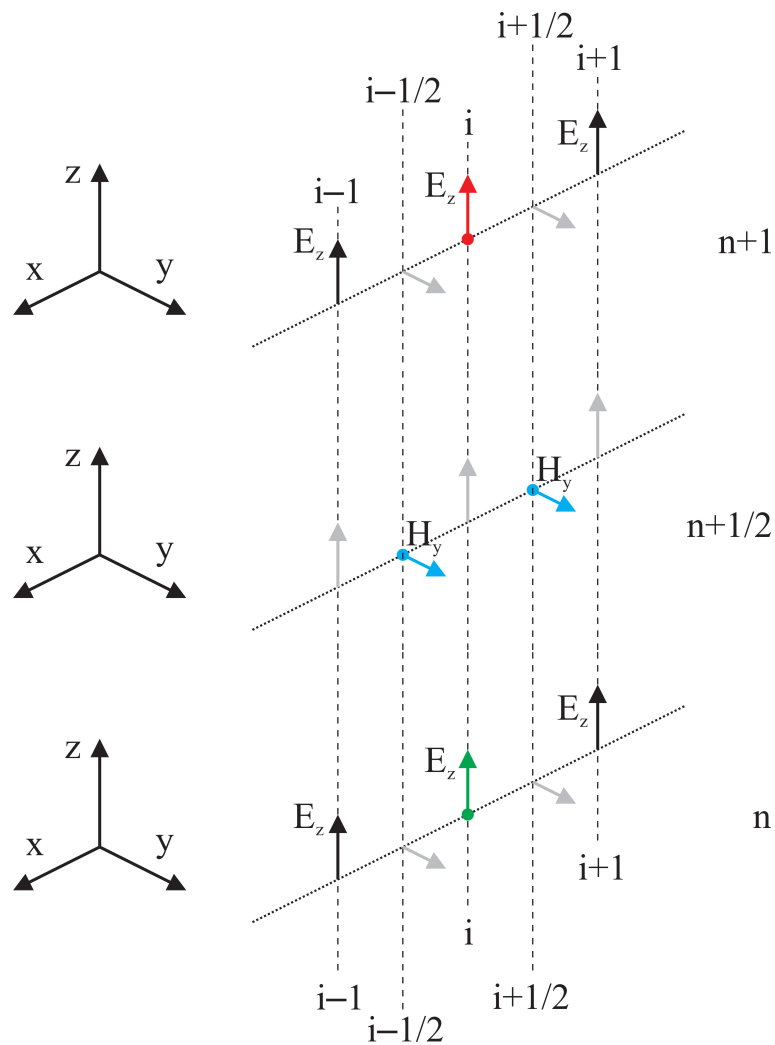
gdzie $O[(\Delta_s)^2]$ i $O[(\Delta_t)^2]$ są resztami zapisanymi przy użyciu skróconej notacji.⁵

3.3.1. Siatka Yee w jednym wymiarze

Cechą charakterystyczną siatki Yee jest rozsuniecie składowych pola elektromagnetycznego. W celu obliczenia wartości pola w danym punkcie korzystamy z jego wartości w tym miejscu w poprzednim kroku czasowym, oraz z wartości dualnego (magnetycznego lub elektrycznego) pola w sąsiednich punktach pół kroku czasowego wcześniej. Rysunek 3.1. przedstawia siatkę Yee w jednym wymiarze w trzech chwilach czasu $(n, n + 1/2, n + 1)$. Zatem mając początkowy rozkład pola możemy obliczać ewolucję czasową jego wartości.

⁴W ogólności krok w różnych kierunkach może być różny $\Delta x \neq \Delta y \neq \Delta z \neq \Delta_s$, lecz w rozpatrywanych przypadkach $\Delta x = \Delta y = \Delta z = \Delta_s$.

⁵Wiodące człony reszt są rzędu $(\Delta_s)^2$ i $(\Delta_t)^2$



Rysunek 3.1. Siatka Yee dla przypadku jednowymiarowego w trzech kolejnych chwilach (n , $n+1/2$, $n+1$). Wartość $E_z|_i^{n+1}$ (kolor czerwony) obliczana jest na podstawie wartości $E_z|_i^n$ (kolor zielony) oraz $H_y|_{i+1/2}^{n+1/2}$ i $H_y|_{i-1/2}^{n+1/2}$ (kolor niebieski).

Zapisując układ równań (3.10) na siatce Yee w postaci dyskretnej otrzymujemy:

$$\begin{aligned} \frac{H_y|_{i+1/2}^{n+1/2} - H_y|_{i+1/2}^{n-1/2}}{\Delta t} &= \frac{1}{\mu|_{i+1/2}} \left[\frac{E_z|_{i+1}^n - E_z|_i^n}{\Delta_s} + \right. \\ &\quad \left. - \sigma_h|_{i+1/2} H_y|_{i+1/2}^n \right], \\ \frac{E_z|_i^{n+1} - E_z|_i^n}{\Delta t} &= \frac{1}{\varepsilon|_i} \left[\frac{H_y|_{i+1/2}^{n+1/2} - H_y|_{i-1/2}^{n+1/2}}{\Delta_s} + \right. \\ &\quad \left. - \sigma_e|_i E_z|_i^{n+1/2} \right]. \end{aligned} \quad (3.17)$$

We wzorach (3.17) pojawiają się wartości pola elektrycznego w punktach o współrzędnych całkowitych (i , $i + 1$) oraz pola magnetycznego w punktach o współrzędnych połówkowych ($i - 1/2$, $i + 1/2$). Aby do podobnej zależności doprowadzić dla zmiennej czasowej zastosować należy następujące uśrednienia:

$$\begin{aligned} H_y|_{i+1/2}^n &= \frac{H_y|_{i+1/2}^{n+1/2} + H_y|_{i+1/2}^{n-1/2}}{2}, \\ E_z|_i^{n+1/2} &= \frac{E_z|_i^{n+1} + E_z|_i^n}{2}. \end{aligned} \quad (3.18)$$

Po podstawieniu zależności (3.18) do (3.17) i prostych przekształceniach otrzymujemy równania:

$$\begin{aligned} H_y|_{i+1/2}^{n+1/2} &= C_{hh}|_{i+1/2} H_y|_{i+1/2}^{n-1/2} + C_{he}|_{i+1/2} (E_z|_{i+1}^n - E_z|_i^n), \\ E_z|_i^{n+1} &= C_{ee}|_i E_z|_i^n + C_{eh}|_i (H_y|_{i+1/2}^{n+1/2} - H_y|_{i-1/2}^{n+1/2}), \end{aligned} \quad (3.19)$$

gdzie:

$$\begin{aligned} C_{hh}|_{i+1/2} &= \left(1 - \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}} \right) / \left(1 + \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}} \right), \\ C_{he}|_{i+1/2} &= \left(\frac{\Delta t}{\mu|_{i+1/2} \Delta_s} \right) / \left(1 + \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}} \right), \\ C_{ee}|_i &= \left(1 - \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i} \right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i} \right), \\ C_{eh}|_i &= \left(\frac{\Delta t}{\varepsilon|_i \Delta_s} \right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i} \right). \end{aligned} \quad (3.20)$$

Ograniczając rozważania do ośrodków nieabsorbujących otrzymujemy następujące wartości współczynników:

$$\begin{aligned} C_{hh}|_{i+1/2} &= 1, \\ C_{he}|_{i+1/2} &= \left(\frac{\Delta t}{\mu|_{i+1/2} \Delta_s} \right), \\ C_{ee}|_i &= 1, \\ C_{eh}|_i &= \left(\frac{\Delta t}{\varepsilon|_i \Delta_s} \right). \end{aligned} \quad (3.21)$$

3.4. Źródła promieniowania

Istotnym elementem modelowania metodą FDTD są źródła promieniowania [4]. Początkowo (w chwili startu obliczeń numerycznych) wartości pola elektromagnetycznego są wyzerowane w całym obszarze obliczeniowym. Dopiero z upływem czasu symulacji (w kolejnych krokach czasowych obliczeń) wprowadza się promieniowanie. Prostymi w implementacji numerycznej są źródła twarde oraz źródła miękkie opisane poniżej.

3.4.1. Źródła twarde

Twarde źródło promieniowania uzyskuje się przez zadawanie wartości pola w wybranym punkcie siatki Yee, w kolejnych krokach czasowych. Przykładowo, aby wprowadzić energię przez twarde źródło, w jednowymiarowej siatce Yee, w punkcie i_s , należy posłużyć się jedną z zależności:

$$\begin{aligned} E_z|_{i_s}^n &= f(n), \\ H_y|_{i_s}^n &= f(n). \end{aligned} \quad (3.22)$$

Funkcja $f(n)$ może być dobierana według potrzeb.⁶ Przykładowymi są:

- fala sinusoidalna o częstotliwości ν i amplitudzie A_0

$$f(n) = A_0 \sin(2\pi\nu n \Delta_t), \quad (3.23)$$

- impuls gaussowski o maksimum w chwili n_0 o wartości A_0 oraz charakterystycznym czasie zaniku n_δ (warunek $n_0 > 3n_\delta$ powinien być zachowany, by zapewnić gładkie włączenie źródła)

$$f(n) = A_0 \exp \left[- \left(\frac{n - n_0}{n_\delta} \right)^2 \right], \quad (3.24)$$

- złożenie impulsu gaussowskiego i fali sinusoidalnej

$$f(n) = A_0 \exp \left[- \left(\frac{n - n_0}{n_\delta} \right)^2 \right] \cdot \sin [2\pi\nu (n - n_0) \Delta_t]. \quad (3.25)$$

Tak zadane źródła wydają się być bardzo dogodne, jednakże posiadają one pewne wady. Jeżeli fala wyemitowana przez takie źródło odbije się od badanej struktury, a następnie powróci do źródła ulegnie ponownemu odbiciu. Twarde źródła zachowują się jak idealne przewodniki elektryczne (lub magnetyczne).

3.4.2. Źródła miękkie

Miękkie źródła promieniowania uzyskuje się przez zadawanie wartości prądu w wybranym punkcie siatki Yee, w kolejnych krokach czasowych. W algorytmie uzyskuje się taki efekt poprzez dodawanie do wielkości obliczonej według algorytmu poprawki związanej z prądem. Przykładowo, aby wprowadzić prąd elektryczny w punkcie i_s , do wielkości $\widetilde{E}_z|_{i_s}^n$ obliczonej według formuły (3.19) dodaje się wartość zależną od czasu.

$$E_z|_{i_s}^n = \widetilde{E}_z|_{i_s}^n + f(n). \quad (3.26)$$

Wszystkie wcześniejsze uwagi dotyczące funkcji $f(n)$ zachowują ważność.

Zaletą tak wprowadzonych źródeł jest ich transparentność. Nie generują one ponownych odbić.

⁶Istotnym jest, dla wyeliminowania wysokoczęstotliwościowych składowych widma, by funkcja $f(n)$ przechodziła gładko od(do) zera przy włączaniu(wyłączaniu).



3.5. Pierwszy program

W rozdziale zostanie przedstawiony prosty program wykorzystujący metodę FDTD. Poniżej znajduje się kod programu wraz ze szczegółowym omówieniem.

Program 1. ftd-01.m

Program jest skryptem napisanym dla środowiska MATLAB. W pierwszej kolejności czyszczona jest zawartość pamięci roboczej oraz podawane są wartości stałych fizycznych.

```
clear all; % czyszczenie pamieci roboczej
2 mu_0 = 1.2566370614359171e-6; % przenikalnosc magnetyczna
e_0 = 8.8541878176203892e-12; % przenikalnosc elektryczna
4 cc = sqrt(1/mu_0/e_0); % predkosc swiatla
```

Przygotowanie pamięci roboczej i inicjalizacja zmiennych zawierających wartości stałych fizycznych

Następnie określone są rozmiary kroku przestrzennego oraz powiązanego z nim kroku czasowego. Współczynnik 1/2 jest związany z kryterium stabilności metody.⁷

```
6 dx = 1; % okreslenie rozmiaru kroku przestrzennego
dt = 0.5*dx/cc; % okreslenie rozmiaru kroku czasowego
```

Określenie rozmiaru kroku czasowego i przestrzennego

Określenie rozmiaru symulacji (liczby punktów siatki przestrzennej i liczby kroków czasowych w symulacji).

```
10 i = 200; % okreslenie rozmiaru siatki
n = 600; % okreslenie dlugosci trwania symulacji
```

Określenie rozmiarów siatki

Podanie parametrów źródła. Rozpatrywana jest propagacja impulsu gaussowskiego określonego równaniem

$$f(n) = A_0 \exp \left[- \left(\frac{n - n_0}{n_\delta} \right)^2 \right]. \quad (3.27)$$

Podawane są zatem parametry n_0 oraz n_δ (spread) oraz położenie źródła.

```
12 n0 = 40; % parametry impulsu początkowego
nd = 12;
14 is = [101]; % położenie źródła
```

Określenie parametrów źródła

Następnie inicjalizowane są:

– wartości pól,

```
18 e = zeros(1,i); % inicjalizacja pola elektrycznego
h = zeros(1,i); % inicjalizacja pola magnetycznego
```

Inicjalizacja wartości pól

⁷Do określenia stabilności metody wykorzystuje się parametr $S = c\Delta_t/\Delta_s$. Dla $S \leq 1$ metoda jest stabilna.



– współrzędne węzłów

```
20 %obliczenie wartosci polozenia wezlow siatki
    xe = [1:i]
22 xh = [1:i]+.5;
```

Obliczenie współrzędnych węzłów

– przenikalności elektryczna i magnetyczna, które w rozważanym przypadku są przenikalnościami dla próżni,

```
24 %definicja parametrow materialowych
    eps = e_0*ones(1,i);
26 mu = mu_0*ones(1,i);
```

Obliczenie przenikalności ośrodków

– współczynniki $C_{eh}|_i$ oraz $C_{he}|_{i+1/2}$ z równania (3.21).

```
28 %obliczenie wspolczynnikow ceh i che
    ceh = (dt./eps/dx);
30 che = (dt./mu/dx);
```

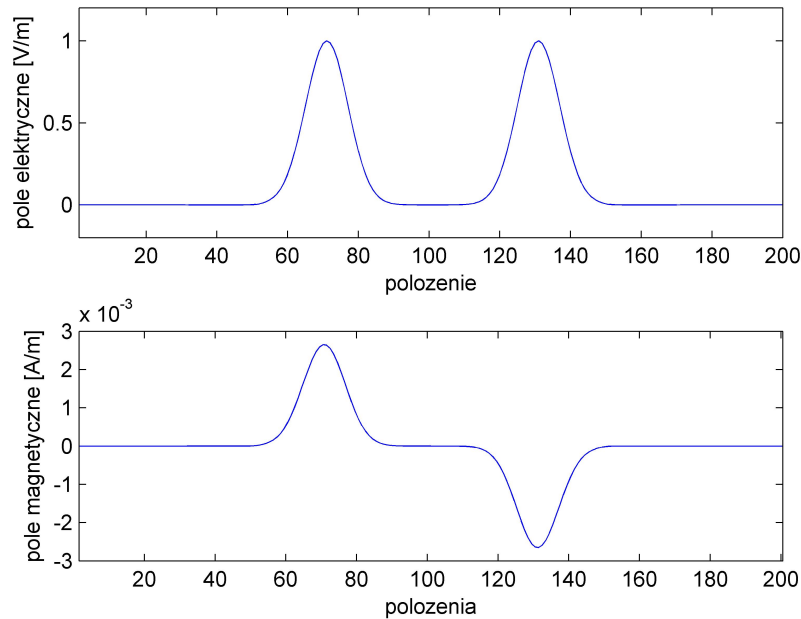
Obliczenie współczynników C_{eh} i C_{he}

Ostatnią czynnością przed przejściem do głównej pętli programu jest inicjalizacja wykresów.

```
figure; %otworzenie okna wykresow
34 % wykreslenie wykresow dla chwili poczatkowej
    subplot(2,1,1); %wykres pola elektrycznego
36 pe = plot(xe, e);
    axis([min(xe), max(xe), -.2, 1.2]);
38 xlabel('polozenie'); %definicja etykiet osi
    ylabel('pole elektryczne [V/m]');
40
    subplot(2,1,2); %wykres pola magnetycznego
42 ph = plot(xh, h);
    axis([min(xh), max(xh), -3e-3, 3e-3]);
44 xlabel('polozenie');
    ylabel('pole magnetyczne [A/m]');
```

Inicjalizacja wykresów

W pętli głównej programu, która jest pętlą `for`, następuje obliczenie wartości pól elektrycznego i magnetycznego w kolejnych chwilach czasu. Wykorzystywany jest indeks `m`, który przebiega wartości od 1 do `n`. Wykorzystana jest wektorowa notacja środowiska Matlab. Linie 51 i 56 programu odpowiadają równaniu (3.19). W linii 53 wprowadzana jest wartość źródła, w chwili czasu `m`. W każdym wykonaniu pętli odświeżane są dane na wykresie.



Rysunek 3.2. Wykres pól elektrycznego i magnetycznego po 100 krokach czasowych uzyskany przy użyciu opisanego programu

```
48 % glowna petla programu
for m=1:n
50   %obliczenie pola elektrycznego w kolejnej chwili czasu
   e(2:i) = e(2:i) + ceh(2:i).*(h(2:i)-h(1:i-1));
52   %wstawienie wartosci pola dla zrodla
   e(is) = exp(-.5*((m-n0)/nd)^2);
54
   %obliczenie pola magnetycznego w kolejnej chwili czasu
56   h(1:i-1) = h(1:i-1) + che(1:i-1).*(e(2:i)-e(1:i-1));
58
   %aktualizacja danych na wykresach
   set(pe,'YData',e);
60   set(ph,'YData',h);
   refresh;
62   pause(0.01)
end
```

Główna pętla programu

Zadanie 1. Uruchom program `fdtd-01.m`. Odtwórz wyniki przedstawione na Rysunku 3.2.

Zadanie 2. Zmodyfikuj program `fdtd-01.m`, aby prześledzić dalszą propagację impulsów. Co się dzieje, gdy impuls osiągnie granice obszaru obliczeniowego?



3.6. Warunki brzegowe

W wielu zagadnieniach rozwiązywanych metodą przyrostów skończonych rozproszona na elementach badanej struktury fala elektromagnetyczna po pewnym czasie symulacji osiąga granice obszaru obliczeniowego. Aby zasymulować jego rozciąganie się w nieskończoność, należy zastosować odpowiednie warunki brzegowe, które pozwolą fali na opuszczenie obszaru obliczeniowego.

Skutecznym rozwiązaniem są zaproponowane przez Berengera [12] w 1994 roku warstwy idealnie dopasowane (PML – perfectly matched layers). Na brzegach obszaru obliczeniowego znajdują się warstwy, które są absorbujące, ale ich właściwości są dobrane tak, by na granicy właściwego obszaru obliczeniowego nie generowało się odbicie (niezależnie od kąta padania).

W opracowaniu przedstawiono absorbujące warunki brzegowe, które są prostym rozwiązaniem sprawdzającym się w obliczeniach jednowymiarowych.

W przedstawionym wcześniej programie symulowana jest propagacja fali elektromagnetycznej w próżni, a rozmiar kroku czasowego Δ_t jest związany z rozmiarem kroku przestrzennym Δ_s formułą:

$$\Delta_t = \frac{1}{2} \frac{\Delta_s}{c}, \quad (3.28)$$

gdzie c jest prędkością światła w próżni.

Front fali elektromagnetycznej poruszającej się z prędkością światła potrzebuje dwóch kroków czasowych, na przebycie jednego kroku przestrzennego. Zatem wartości pól elektrycznego lub magnetycznego w skrajnych punktach obszaru obliczeniowego mogą być wyznaczone w następujący sposób:

$$E_z|_1^{n+2} = E_z|_2^n. \quad (3.29)$$

Wartość pola elektrycznego z punktu sąsiadującego ze skrajnym jest przechowywana przez dwa kroki symulacji w pamięci, a następnie wpisywana jako wartość w polu skrajnym.

We wcześniej przedstawionym programie wprowadzono stosowne modyfikacje, które opisano poniżej.

Program 2. fdtd-02.m

W programie wprowadzono absorbujące warunki brzegowe. W skrajnych węzłach siatki obliczane są wartości:

- pola elektrycznego dla węzła o numerze 1,
- pola magnetycznego dla węzła o numerze $i + 1/2$.

Zatem w pętli głównej programu po obliczeniu aktualnych wartości dla pola elektrycznego dla węzłów o indeksach od 2 do i (linia 55) i po wpisaniu wartości w punkcie, w którym znajduje się źródło (linia 57), aktualizujemy wartość pola elektrycznego w węźle o indeksie 1 (linia 59). Pomocnicze zmienne `etemp1` oraz `etemp2` służą do przechowania odpowiednich wartości przez dwa kroki czasowe. Podobne rozwiązanie zastosowano dla drugiego skrajnego punktu dla pola magnetycznego.

```
52 % glowna petla programu
for m=1:n
54     %obliczenie pola elektrycznego w kolejnej chwili czasu
    e(2:i) = e(2:i) + ceh(2:i).*(h(2:i)-h(1:i-1));
56     %wstawienie wartosci pola dla zrodla
```



```

e(is) = exp(-.5*((m-n0)/nd)^2);
58 %absorbujace warunki brzegowe dla pierwszego wezla siatki
e(1) = etemp2;
60 etemp2 = etemp1;
etemp1 = e(2);
62
%obliczenie pola magnetycznego w kolejnej chwili czasu
64 h(1:i-1) = h(1:i-1) + che(1:i-1).*(e(2:i)-e(1:i-1));
%absorbujace warunki brzegowe dla ostatniego wezla siatki
66 h(i) = htemp2;
htemp2 = htemp1;
68 htemp1 = h(i-1);

70 %aktualizacja danych na wykresach
set(pe,'YData',e);
72 set(ph,'YData',h);
refresh;
74 pause(0.01)
end

```

Główna pętla programu po wprowadzeniu absorbujących warunków brzegowych

Oprócz tej modyfikacji w pętli głównej należy zainicjalizować zmienne pomocnicze `etemp1`, `etemp2`, `htemp1` oraz `htemp2`.

```

e = zeros(1,i); %inicjalizacja pola elektrycznego
18 h = zeros(1,i); %inicjalizacja pola magnetycznego
etemp2 = 0; %inicjalizacja zmiennych pomocniczych
20 etemp1 = 0; %wykorzystywanych w
htemp2 = 0; %absorbujacych warunkach brzegowych
22 htemp1 = 0;

```

Inicjalizacja wartości pól oraz zmiennych pomocniczych

Zadanie 3. Uruchom program `fdtd-02.m`, odtwórz wyniki przedstawione na Rysunku 3.3.

3.7. Propagacja w ośrodku dielektrycznym

W celu opisanie propagacji w ośrodku o zadanej względnej przenikalności elektrycznej należy odpowiednio zmienić wartości współczynników $C_{eh}|_i$ zgodnie z równaniem (3.21).

Program 3. `fdtd-03.m`

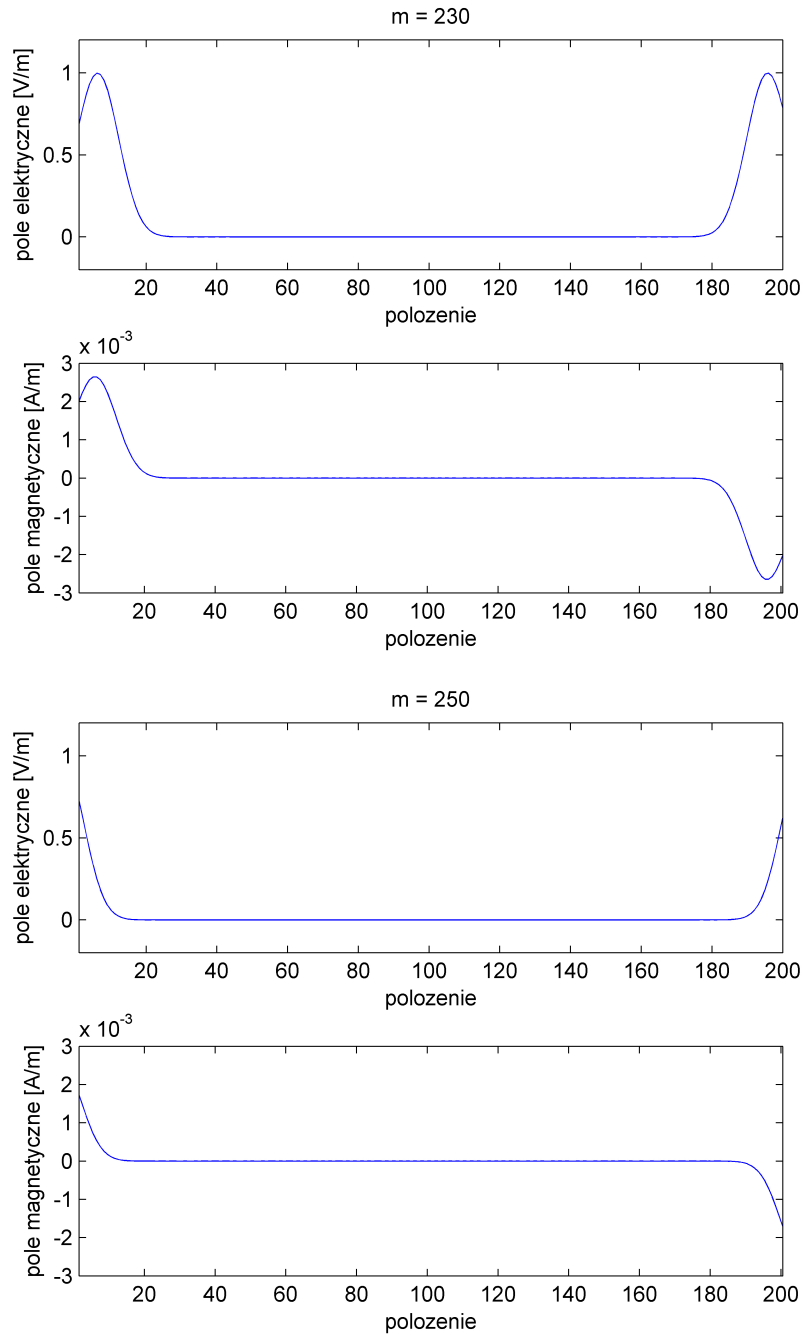
Poniżej przedstawiona modyfikacja pozwala przeprowadzić symulację odbicia rozważanego impulsu gaussowskiego na granicy próżni i ośrodka o względnej przenikalności elektrycznej równej 4.

```

28 %definicja parametrow materialowych
eps = e_0*[1.*(xe<=100) + 4.*(xe>100)];
30 mu = mu_0*ones(1,i);

```

Obliczenie przenikalności ośrodków



Rysunek 3.3. Wykres pól elektrycznego i magnetycznego po 230 i po 250 krokach czasowych uzyskany przy użyciu opisanego programu



Dla punktów siatki, których położenie jest mniejsze od 100, wartość względnej przenikalności elektrycznej jest równa 1, natomiast dla punktów, których położenie jest większe od 100 wartość względnej przenikalności elektrycznej jest równa 4.

Zmienna `eps` jest wykorzystywana do obliczenia współczynników `ceh`.

```
32 %obliczenie wspolczynnikow ceh i che
ceh = (dt./eps/dx);
34 che = (dt./mu/dx);
```

Obliczenie współczynników $C_{eh}|_i$ oraz $C_{he}|_i$

Dodatkowo przesunięto źródło impulsu w symulacji z węzła o numerze 100 do węzła o numerze 5.

```
14 is = [5];           %polozenie zrodla
```

Położenie źródła

Został także zmieniony typ źródła z twardego na miękkie.

```
58 %wstawienie wartosci pola dla zrodla
e(is) = e(is) + exp(-.5*((m-n0)/nd)^2);
```

Miękkie źródło

Impuls po osiągnięciu granicy ulega częściowemu odbiciu. Dane z symulacji przedstawione są na Rysunku 3.4.

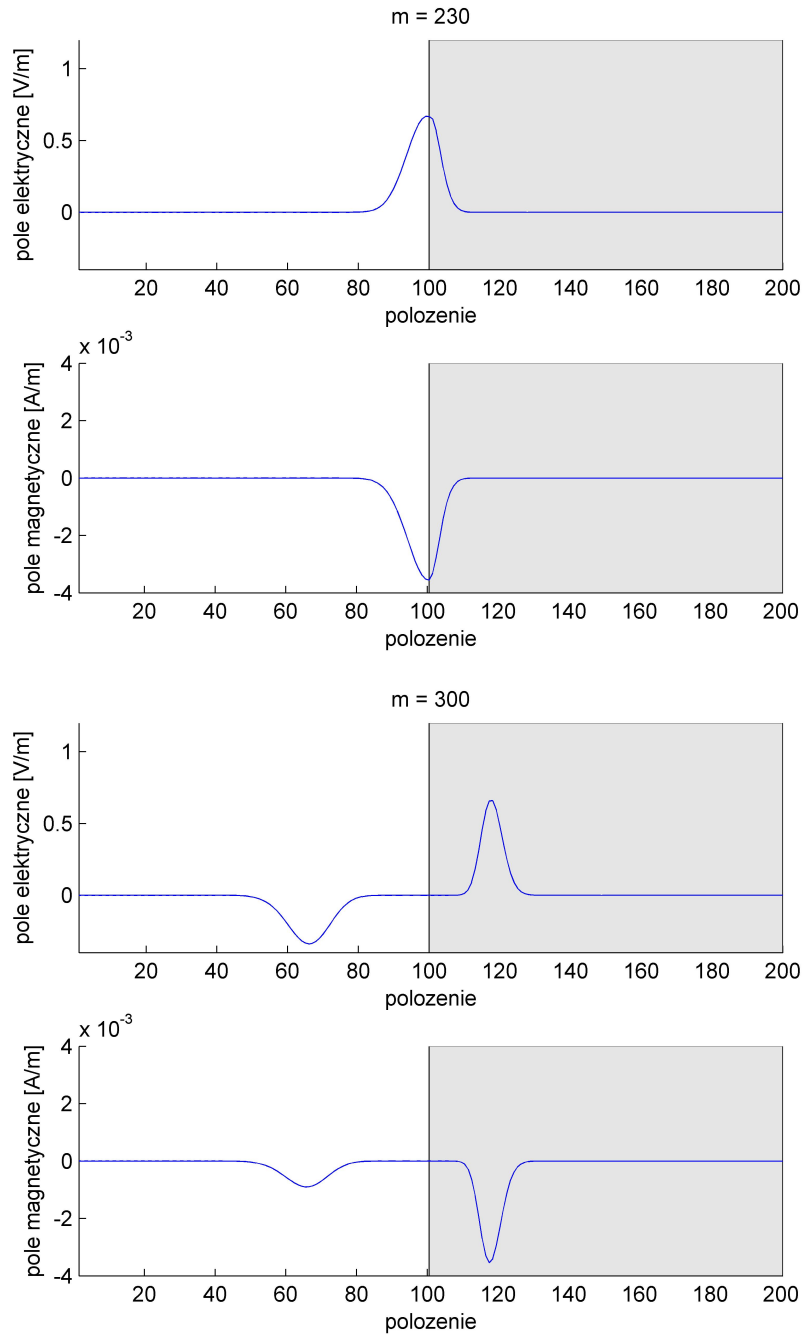
Przedstawiony przykład pokazuje łatwość z jaką do obszaru obliczeniowego w metodzie FDTD można wprowadzić różne obiekty.

Zadanie 4. Uruchom program `fdtd-03.m`, odtwórz wyniki przedstawione na Rysunku 3.4.

3.8. Propagacja w ośrodku absorbującym

W celu opisanie propagacji w ośrodku absorbującym należy powrócić do współczynników wyrażonych równaniami (3.20):

$$\begin{aligned}
 C_{hh}|_{i+1/2} &= \left(1 - \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}}\right) / \left(1 + \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}}\right), \\
 C_{he}|_{i+1/2} &= \left(\frac{\Delta t}{\mu|_{i+1/2} \Delta_s}\right) / \left(1 + \frac{\sigma_h|_{i+1/2} \Delta t}{2 \mu|_{i+1/2}}\right), \\
 C_{ee}|_i &= \left(1 - \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right), \\
 C_{eh}|_i &= \left(\frac{\Delta t}{\varepsilon|_i \Delta_s}\right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right).
 \end{aligned} \tag{3.30}$$



Rysunek 3.4. Wykresy pól elektrycznego i magnetycznego po 230 i po 300 krokach czasowych. Widoczny jest impuls po osiągnięciu granicy oraz po odbiciu. Szarym kolorem zaznaczono ośrodek o względnej przenikalności elektrycznej równej 4.

Po przyjęciu założenia, że $\sigma_h = 0$, współczynniki te mają postać:

$$\begin{aligned} C_{hh}|_{i+1/2} &= 1, \\ C_{he}|_{i+1/2} &= \frac{\Delta t}{\mu|_{i+1/2} \Delta_s}, \\ C_{ee}|_i &= \left(1 - \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right), \\ C_{eh}|_i &= \left(\frac{\Delta t}{\varepsilon|_i \Delta_s}\right) / \left(1 + \frac{\sigma_e|_i \Delta t}{2 \varepsilon|_i}\right). \end{aligned} \quad (3.31)$$

Opisany poniżej program przedstawia oddziaływanie fali sinusoidalnej z ośrodkiem absorbującym.

Program 4. fdtd-04.m

Modyfikacje, jakie zostały wprowadzone to:

- zmniejszenie rozmiaru kroku przestrzennego, a co za tym idzie kroku czasowego;

```
6 dx = 0.01;           %okreslenie rozmiaru kroku przestrzennego
  dt = 0.5*dx/cc;     %okreslenie rozmiaru kroku czasowego
```

Określenie rozmiaru kroku czasowego i przestrzennego

- powiększenie rozmiaru obszaru obliczeniowego i wydłużenie czasu trwania symulacji;

```
10 i = 300;           %okreslenie rozmiaru siatki
    n = 1500;        %okreslenie dlugosci trwania symulacji
```

Określenie rozmiarów siatki

- zmiana funkcji opisującej pracę źródła;

```
12 n0 = 10;          %parametry impulsu początkowego
    is = [5];        %polozenie zrodla
```

Określenie parametrów źródła

```
58 %wstawienie wartosci pola dla zrodla
    e(is) = e(is) + sin(m/n0);
```

Funkcja opisująca źródło

- wprowadzenie do opisu, oprócz przenikalności elektrycznej i magnetycznej, także przewodności elektrycznej,

```
%definicja parametrow materialowych
28 eps = e_0*(1.*(xe<=100) + 4.*(xe>100));
    mu = mu_0*ones(1,i);
30 sigma = 0.*(xe<=100) + 0.04.*(xe>100);
```

Wprowadzenie parametrów materiałowych ośrodków



- zmiana wyrażień na współczynniki $C_{eh}|_i$ oraz $C_{he}|_{i+1/2}$, a także wprowadzenie współczynników $C_{ee}|_i$ zgodnie z równaniem (3.31);

```
32 %obliczenie współczynników ceh i che
   ceh = (dt./eps/dx)./(1+sigma*dt/2./eps);
34 che = (dt./mu/dx);
   cee = (1-sigma*dt/2./eps)./(1+sigma*dt/2./eps);
```

Obliczenie współczynników C_{eh} , C_{he} , C_{ee}

- zmiana osi wykresów;

```
38 figure; %otworzenie okna wykresow
   % wykreslenie wykresow dla chwili początkowej
40 subplot(2,1,1); %wykres pola elektrycznego
   patch([100 300 300 100],[-2 -2 2 2],[.9 .9 .9]), hold on;
42 pe = plot(xe, e); axis([min(xe), max(xe), -2, 2]);
   xlabel('polozenie'); %definicja etykiet osi
44 ylabel('pole elektryczne [V/m]');

46 subplot(2,1,2); %wykres pola magnetycznego
   patch([100 300.5 300.5 100],[-4e-3 -4e-3 4e-3 4e-3],...
48 [.9 .9 .9]), hold on;
   ph = plot(xh, h); axis([min(xh), max(xh), -4e-3, 4e-3]);
50 xlabel('polozenie');
   ylabel('pole magnetyczne [A/m]');
```

Inicjalizacja wykresów pól

- zmiana sposobu obliczenia pola elektrycznego uwzględniająca wprowadzenie współczynników $C_{ee}|_i$.

```
56 %obliczenie pola elektrycznego w kolejnej chwili czasu
   e(2:i) = cee(2:i).*e(2:i)+ceh(2:i).*(h(2:i)-h(1:i-1));
```

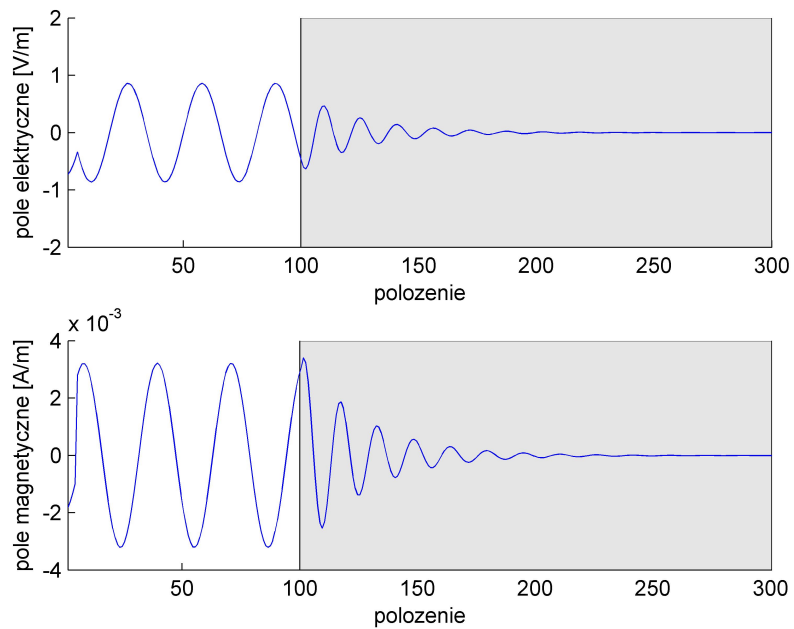
Obliczenia wartości pola elektrycznego

Rysunek 3.5. przedstawia rozkład pól po 1500 krokach czasowych. Zaobserwować można zmniejszanie amplitudy wraz z głębokością na jaką wnika fala.

Zadanie 5. Uruchom program `fdtd-04.m`. Odtwórz wyniki przedstawione na Rysunku 3.5.

Zadanie 6. Wykonaj symulację odbicia sinusoidalnej fali elektromagnetycznej na granicy dwóch ośrodków o różnych współczynnikach załamania. Następnie zaproponuj odpowiednią warstwę antyrefleksyjną i sprawdź jej działanie.

Zadanie 7. Wykonaj symulację odbicia fali elektromagnetycznej od zwierciadła Bragga. W tym celu wprowadź odpowiednią zależność względnej przenikalności elektrycznej od położenia.



Rysunek 3.5. Wykres pól elektrycznego i magnetycznego po 1500 krokach czasowych uzyskany przy użyciu opisanego programu



TA STRONA JEST INTENCJONALNIE PUSTA



Bibliografia

- [1] K. Yee, *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Transactions on Antennas and Propagation, vol. 14, str. 302-307 (1966).
- [2] A. Taflove, M. E. Brodwin, *Numerical Solution of Steady-state Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations*, IEEE Transactions on Microwave Theory and Techniques, vol. 23, nr 8, str. 623-630 (1975).
- [3] A. Taflove, *Application of the finite-difference time-domain method to sinusoidal steady state electromagnetic penetration problems*, IEEE Transactions on Electromagnetic Compatibility, vol. 22, nr 3, str. 191-202 (1980).
- [4] A. Taflove, S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time Domain Method*, Artech House, Inc. (2005).
- [5] D. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, Wiley-IEEE Press (2000).
- [6] K. Tarnowski, *Analiza numeryczna rozkładu i propagacji pola elektromagnetycznego w metamateriałach metodą FDTD*, praca magisterska, Politechnika Wroclawska (2007).
- [7] D. J. Griffiths, *Podstawy elektrodynamiki*, Wydawnictwo Naukowe PWN, Warszawa (2006).
- [8] J. D. Jackson, *Elektrodynamika klasyczna*, Wydawnictwo Naukowe PWN, Warszawa (1982).
- [9] M. Suffczyński, *Elektrodynamika*, Wydawnictwo Naukowe PWN, Warszawa (1980).
- [10] R. P. Feynman, R. B. Leighton, M. Sands, *Feynmana wykłady z fizyki*, Wydawnictwo Naukowe PWN, Warszawa (2001).
- [11] D. Kincaid, W. Cheney, *Analiza numeryczna*, Wydawnictwa Naukowo-Techniczne, Warszawa (2002).
- [12] J. P. Berenger, *A perfectly matched layer for absorption of electromagnetic waves*, Journal of Computational Physics, vol. 114, str. 185-200 (1994).



TA STRONA JEST INTENCJONALNIE PUSTA



Rozdział 4.

Modelowanie właściwości światłowodów metodą elementów skończonych

Celem tego rozdziału jest zapoznanie czytelnika z podstawami metody elementów skończonych (MES, w literaturze anglojęzycznej FEM – finite element method) oraz nauczanie go jej praktycznego wykorzystania. W pierwszej części rozdziału znajduje się wyprowadzenie MES dla skalarnych zagadnień brzegowych w elektromagnetyzmie. Dalej zaprezentowany jest przykład rozwiązania skalarnego równania falowego dla lokalnie izotropowego, niejednorodnego falowodu. W końcowej części znajduje się omówienie kodu źródłowego dla implementacji MES w środowisku MATLAB.

4.1. Wprowadzenie

Metoda elementów skończonych jest techniką numerycznego wyznaczania przybliżonych rozwiązań zagadnień brzegowych. Matematyczne podstawy tej metody opracowane zostały na początku lat czterdziestych XX wieku przez Couranta. Początkowo metoda ta stosowana była do mechanicznych obliczeń strukturalnych, jednak począwszy od 1968 roku zaczęto ją także wykorzystywać do rozwiązywania problemów elektromagnetyzmu [1]. W pierwszym kroku MES [1, 2] przeprowadza się dyskretyzację obszaru, w którym poszukuje się rozwiązania, poprzez podzielenie go na skończoną liczbę prostych geometrycznie podobszarów, które nazywa się elementami skończonymi. Wybór kształtu elementów zależy od rozwiązywanego zagadnienia, dla zagadnień dwuwymiarowych mają najczęściej kształt trójkątów. Następnie wyprowadza się równanie bazowe dla pojedynczego elementu. Potem zapisuje się układ równań dla wszystkich elementów tworzących dziedzinę poszukiwanej funkcji a w ostatnim kroku rozwiązuje się ten układ metodami algebraicznymi, uzyskując wyniki nadające się do wizualizacji i dalszego przetworzenia.

4.2. Sformułowanie problemu

Metoda elementów skończonych wykorzystywana jest w ogólności do rozwiązywania zagadnień wartości brzegowych [2]. Zagadnienie wartości brzegowych to równanie różniczkowe, zwyczajne lub cząstkowe, którego rozwiązywane musi spełniać dodatkowe ograniczenia nazywane warunkami brzegowymi. Przykładem takiego równania jest rozpatrywane w płaszczyźnie xy stacjonarne równanie Helmholtza, które dla funkcji własnych ψ ma postać [1–4]

$$\nabla_{\perp}^2 \psi + k^2 \psi = 0, \quad (4.1)$$



gdzie ∇_{\perp}^2 jest operatorem Laplace'a

$$\nabla_{\perp}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad (4.2)$$

a k^2 jest wartością własną. Równanie Helmholtza opisuje wiele zjawisk fizycznych. W kontekście elektromagnetyzmu, jest to stacjonarne równanie falowe, które może zostać wykorzystane do badania rozkładów pól modów TE (*transverse electric*) i TM (*transverse magnetic*) w falowodach o dowolnym kształcie przekroju poprzecznego [1–4]. W falowodach o jednorodnym rozkładzie współczynnika załamania równanie (4.1) jest spełnione przez składową podłużną (w kierunku propagacji z) pola elektrycznego E_z lub magnetycznego H_z poszczególnych modów, a pierwiastki kwadratowe kolejnych wartości własnych k stanowią długości fali odcięcia tych modów. Dla falowodów niejednorodnych (np. światłowodów planarnych lub światłowodów cylindrycznych w przybliżeniu słabego przewodzenia), równanie (4.1) jest spełnione dla składowych poprzecznych pola - wtedy funkcja ψ oznacza składową poprzeczną pola elektrycznego dla przypadku TE oraz składową poprzeczną pola elektrycznego dla przypadku TM. Pierwiastek wartości własnej k ma w takim przypadku inny sens fizyczny - jest tzw. poprzeczną stałą propagacji k_{\perp} , która związana jest ze stałą propagacji w kierunku podłużnym b oraz rozkładem poprzecznym współczynnika załamania n poniższą zależnością [1–4]

$$k^2 \equiv k_{\perp}^2(x, y) = k_0^2 n^2(x, y) - \beta^2, \quad (4.3)$$

w której k_0 to moduł liczby falowej dla długości fali świetlnej λ_0

$$k_0 = \frac{2\pi}{\lambda_0}. \quad (4.4)$$

Należy w tym miejscu zdać sobie sprawę [2], że dla wielu zagadnień (przykładem jest falowód o niejednorodnym rozkładzie współczynnika załamania) skalarne falowe równanie stanowi tylko przybliżenie bardziej ogólnego wektorowego równania falowego (ponieważ nie uwzględnia m.in. ono sprzężenia pomiędzy polem elektrycznym a magnetycznym, efektów polaryzacyjnych, skoków pola na granicy niejednorodności oraz szeregu innych zjawisk fizycznych opisanych równaniami równania Maxwella) [1, 2]. W dalszej części tego rozdziału to przybliżenie jest uznawane za wystarczające i będzie omawiane w kontekście propagacji modów w falowodach o niejednorodnym rozkładzie współczynnika załamania, czyli równania (4.1) z wartościami własnymi (4.3):

$$\nabla_{\perp}^2 \psi + k_{\perp}^2 \psi = 0. \quad (4.5)$$

Drugą istotną kwestią związaną z zagadnieniami brzegowymi są warunki brzegowe bez których nie da się znaleźć jednoznacznych rozwiązań tego typu problemów. Najczęściej są to tzw. jednorodne warunki brzegowe Dirichleta lub Neumana wymagające, odpowiednio, zerowania się rozwiązania lub pochodnej rozwiązania na brzegu obszaru poszukiwań tego rozwiązania. W omawianym przypadku odpowiednikiem warunku Dirichleta jest tzw. warunek brzegowy „doskonałego przewodnika elektrycznego” PEC (*perfect electric conductor*) dla modów TM. Odpowiednikiem warunku Neumana jest tzw. warunek brzegowy „doskonałego przewodnika magnetycznego” PMC (*perfect magnetic conductor*) dla modów TE. Warunek PEC w zapisie matematycznym przyjmuje postać

$$\hat{n} \times \vec{E} = 0, \quad (4.6)$$



natomiast warunek PMC

$$\hat{n} \times \vec{H} = 0, \quad (4.7)$$

gdzie \vec{E} to wektor natężenia pola elektrycznego, \vec{H} to wektor natężenia pola magnetycznego a \hat{n} to wektor o jednostkowej długości prostopadły do brzegu obszaru ograniczonego danym warunkiem brzegowym. Pierwszy z nich (PEC) powoduje, że z pola elektrycznego o dowolnym kierunku, po „zadziałaniu” warunku brzegowego pozostaje tylko składowa prostopadła (normalna) do brzegu obszaru. Analogiczne jest działanie warunku PMC w odniesieniu do pola magnetycznego. Konsekwencje tych spostrzeżeń są istotne z punktu widzenia MES [2, 4].

W ujęciu klasycznym, zdefiniowane powyżej zagadnienie brzegowe można rozwiązać na kilka sposobów. Dwa z nich, stanowiące podstawę stosownej współcześnie metody elementów skończonych, to metoda wariacyjna Raileigha-Ritza [1–3] oraz metoda residuów ważonych w ujęciu Galerkina [2, 4]. Pierwsza z nich bazuje na przyjęciu dla danego zagadnienia brzegowego takiego funkcjonału, którego minimum jest rozwiązaniem tego zagadnienia [2]. Następnie znajduje się rozwiązanie minimalizując funkcjonał. Uzyskane rozwiązanie jest przybliżeniem w tym sensie, że minimalizacji dokonuje się podstawiając jako próbne funkcje, które stanowią jedynie przybliżenie rozwiązania dokładnego. Przykładowo, przyjmuje się jako przybliżenie funkcję liniową i tak dobiera się jej współczynniki, żeby funkcjonał, którego argumentem jest ta funkcja osiągnął wartość minimalną [2, 3] (przybliżenie polega na tym, że mogą istnieć inne funkcje, dla których osiągnię się lepszą minimalizację funkcjonału). Druga metoda polega z kolei na tym, że konstruuje się residuum ważne dla danego zagadnienia (w uproszczeniu: sumuje się różnice pomiędzy rozwiązaniem przybliżonym, które podstawione do równania (1) daje wartości niezerowe a wartościami rozwiązania dokładnego, czyli zerowymi; różnice te są dodatkowo mnożone przez pewne funkcje wagowe), przy czym jako funkcje wagowe wybiera się takie same funkcje, jakich użyto do aproksymacji rozwiązania. Następnie dobierając odpowiednio współczynniki funkcji wagowych minimalizuje się residuum [2, 4]. W dalszej części przedstawione zostanie rozwiązanie równania (4.5) metodą Galerkina.

Przed przystąpieniem do rozwiązania równania (4.5) metodą elementów skończonych należy wykonać jeszcze jeden krok, tzn. przepisać to równanie z „postaci silnej” (czyli zawierającej drugą pochodną) do „postaci słabej” (czyli wyrażenia całkowo-różniczkowego zawierającego pochodne niższych rzędów). W metodzie Galerkina, przyjmie ono postać

$$\iint_{\Omega} (T_s \nabla_{\perp}^2 \psi + k_{\perp}^2 T_s \psi) ds = 0, \quad (4.8)$$

gdzie T_s oznacza funkcję testową (której postać zostanie wybrana w kolejnym punkcie), a Ω oznacza obszar dla którego szukane jest rozwiązanie (rys. 1 a). Pierwszy składnik równania (4.8) można przepisać zastępując laplasjan dywergencją gradientu

$$\iint_{\Omega} T_s \nabla_{\perp}^2 \psi ds = \iint_{\Omega} T_s (\vec{\nabla}_{\perp} \cdot \vec{\nabla}_{\perp} \psi) ds \quad (4.9)$$

a następnie przy pomocy pierwszej tożsamości Greena (będącej analogiem całkowania przez części) wyrazić go w postaci [4]

$$\iint_{\Omega} T_s (\vec{\nabla}_{\perp} \cdot \vec{\nabla}_{\perp} \psi) ds = - \iint_{\Omega} \vec{\nabla}_{\perp} T_s \cdot \vec{\nabla}_{\perp} \psi ds + \oint_{\Gamma} T_s \frac{\partial \psi}{\partial \hat{n}} dl. \quad (4.10)$$

We wzorze (4.10) pojawia się całkowanie po konturze Γ otaczającym obszar rozwiązania Ω oraz pochodna kierunkowa $\frac{\partial \psi}{\partial \hat{n}}$ po wektorze \hat{n} normalnym do konturu Γ .



Wstawiając z powrotem (4.10) do (4.8) otrzymuje się

$$\iint_{\Omega} \vec{\nabla}_{\perp} T_s \cdot \vec{\nabla}_{\perp} \psi ds - \iint_{\Omega} k_{\perp}^2 T_s \psi ds = \oint_{\Gamma} T_s \frac{\partial \psi}{\partial \hat{n}} dl. \quad (4.11)$$

W tym miejscu należy odwołać się do równań (4.6) i (4.7) definiujących warunki brzegowe w analizowanym zagadnieniu. Ponieważ funkcja testowa ma aproksymować rozwiązanie dokładne, wymaga się od niej (na mocy warunku PEC (6)) wyzerowania się na brzegu Γ obszaru rozwiązania w przypadku TM. Z kolei w przypadku TE, na mocy warunku brzegowego PMC (4.7) pochodna kierunkowa przyjmie wartości zerowe. Dzięki temu można zauważyć, że bez względu na to, czy szukane są rozwiązania TE czy TM prawa strona równania (4.11) jest zawsze równa zero

$$\oint_{\Gamma} T_s \frac{\partial \psi}{\partial \hat{n}} dl = 0. \quad (4.12)$$

Zerowanie się (4.12) oznacza, że warunki brzegowe PEC dla modów TM i PMC dla modów TE są spełnione ze względu na samą postać równania i dlatego też bywają nazywane naturalnymi warunkami brzegowymi. Równanie (4.11) można zatem zapisać jako

$$\iint_{\Omega} \vec{\nabla}_{\perp} T_s \cdot \vec{\nabla}_{\perp} \psi ds = \iint_{\Omega} k_{\perp}^2 T_s \psi ds \quad (4.13)$$

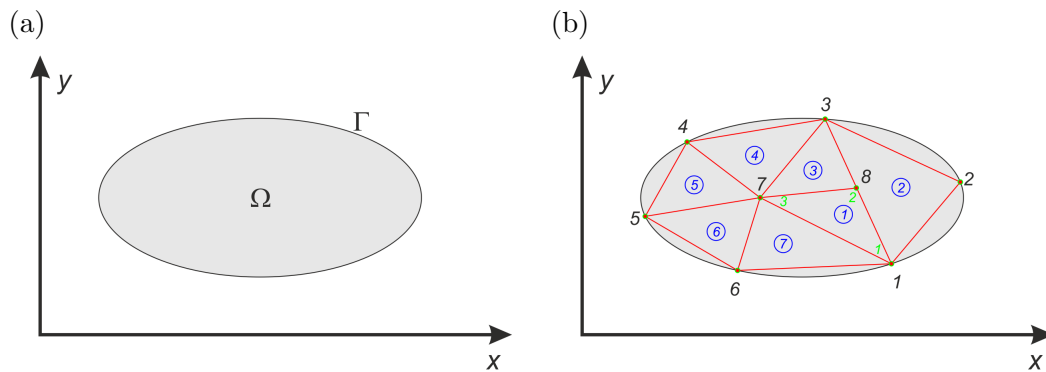
i przystąpić do jego rozwiązywania.

4.3. Rozwiązanie zagadnienia brzegowego metodą MES

Jak wspomniano w punkcie 4.1, rozwiązanie metodą elementów skończonych przebiega w czterech etapach. Są to dyskretyzacja obszaru rozwiązania, wybór funkcji aproksymującej rozwiązanie, zapisanie układu równań dla wszystkich elementów oraz rozwiązanie powstałego układu równań.

4.3.1. Dyskretyzacja

Dyskretyzacja obszaru rozwiązania w metodzie elementów skończonych polega na podziale obszaru rozwiązania na elementy skończone. Element skończony to prosta figura geometryczna, dla której określone zostały węzły oraz funkcje węzłowe (inaczej funkcje interpolacyjne, kształtu lub bazowe) aproksymujące rozkład analizowanej wielkości „nad” elementem [1, 2].



Rysunek 4.1. Obszar rozwiązania Ω zagadnienia brzegowego z konturem Γ przed (a) i po dyskretyzacji (b). Kolorem niebieskim zaznaczono numery elementów, kolorem czarnym globalną numerację węzłów. Kolorem czerwonym zaznaczono przykładową numerację wewnętrzną węzłów dla elementu 1.

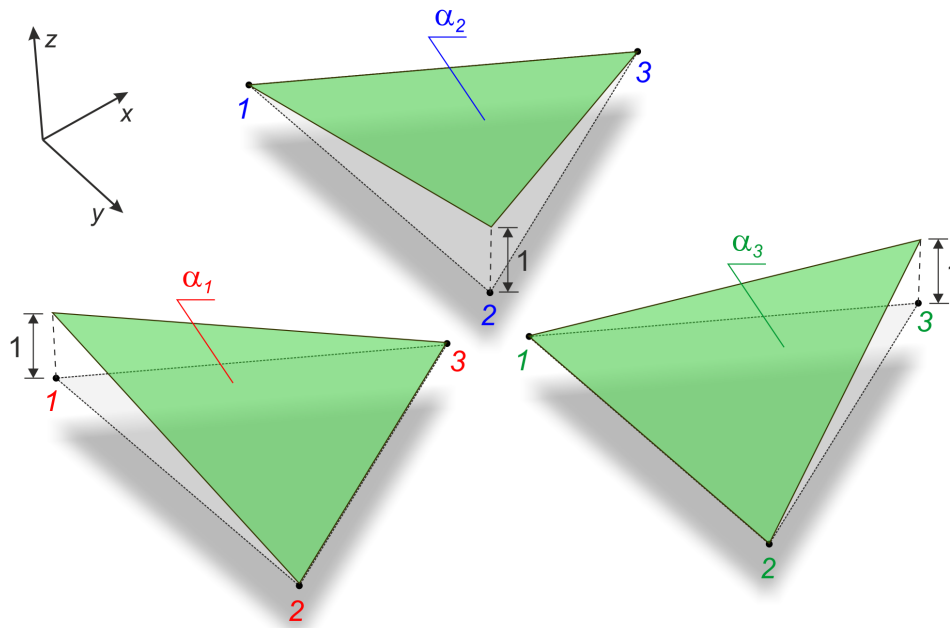
Podczas dyskretyzacji elementowi nadawany jest numer. Numerowane są również węzły należące do danego elementu, przy czym należy rozróżnić numerację lokalną węzłów w obrębie zawierającego je elementu oraz numerację globalną, w obrębie całego rozwiązania, jak zilustrowano to na Rys. 4.1.b.

W zależności od liczby wymiarów zagadnienia wyróżnia się elementy skończone różnego typu. W ogólności:

- element nazywany jest liniowym, jeżeli węzły znajdują się w jego wierzchołkach,
- jeżeli węzłów jest więcej niż wierzchołków, element nazywany jest elementem wyższego rzędu,
- rząd elementu jest zawsze równy rzędowi lokalnej funkcji aproksymującej (tzw. funkcji kształtu lub funkcji bazowej),
- liczba funkcji kształtu dla pojedynczego elementu jest równa liczbie jego węzłów,
- funkcje kształtu są tak dobrane, aby w węzłach które dotyczą ich wartości przyjmowały wartość jeden, a pozostałych węzłach przyjmowały zero (Rys. 4.2.).

W analizowanym przypadku będą stosowane elementy trójkątne liniowe z liniowymi funkcjami wagowymi.

Samo zagadnienie dyskretyzacji obszaru rozwiązania jest zbyt złożone aby omawiać je w ramach tego tekstu. Zamiast tworzyć siatkę samodzielnie, można skorzystać jednego z powszechnie dostępnych generatorów siatek. Można również wykorzystać interaktywne narzędzie *pdetool*, które dostępne jest w środowisku MATLAB z zainstalowanym pakietem dodatkowym *Partial Differential Equation Toolbox*. Przy pomocy tego narzędzia można narysować obszar rozwiązania, nadać mu dodatkowe właściwości (tj. wyznaczyć podobszary o różnym współczynniku załamania) wygenerować siatkę i wyeksportować do plików zmienne zawierające punkty i elementy. Można również wygenerować siatkę z poziomu poleceń konsoli wykorzystując funkcje dostępne w ramach tego pakietu, jak zostanie to zademonstrowane w przykładzie. W notacji stosowanej w środowisku MATLAB tablica punktów zawiera w dwóch wierszach współrzędne odpowiednio x i y kolejnych węzłów (nr kolumny to numer węzła). Tablica elementów zawiera cztery wiersze. Pierwsze trzy kodują globalne numery kolejnych węzłów elementu, czwarty wiersz zawiera numer podobszaru



Rysunek 4.2. Przykładowe funkcje kształtu (bazowe) dla liniowego elementu trójkątnego pierwszego rzędu [1]

do którego należy dany element. Zazwyczaj potrzebna jest też tablica zawierająca numery węzłów brzegowych, dzięki której można w sposób jawny wprowadzić do końcowego układu równań warunki brzegowe, jednak w przytoczonym poniżej przykładzie nie będzie ona wykorzystywana. W narzędziu *pdetool* tablica ta wprawdzie nie jest dostępna bezpośrednio, ale łatwo można ją utworzyć na podstawie analizy tablicy zawierającej krawędzie. Tablica ta w pierwszych dwóch wierszach zawiera numer początkowego i końcowego węzła poszczególnych krawędzi, natomiast w dwóch ostatnich wierszach zawiera numery podobszarów, które dane krawędzie przecinają. Krawędzie brzegowe sąsiadują z podobszarem o indeksie zerowym a dzięki tej informacji (wykorzystując np. polecenia *find* i *unique*) można w prosty sposób uzyskać również numery węzłów brzegowych.

4.3.2. Aproksymacja

Wewnątrz pojedynczego elementu, rozwiązanie dokładne przybliżane jest płaszczyzną (formalnie, należałoby w tym miejscu wyraźnie zaznaczyć różnicę pomiędzy rozwiązaniem dokładnym y a jego przybliżeniem, jednak dla uproszczenia zapisu nie wprowadza się nowego symbolu)

$$\psi = a + bx + cy, \quad (4.14)$$

gdzie a , b i c to nieznanne parametry. Zakładając, że znane są wartości rozwiązania y_1 , y_2 , y_3 w węzłach elementu, równanie tej płaszczyzny możemy znaleźć przy pomocy interpolacji zapisując układ równań

$$\begin{aligned} \psi_1 &= a + bx_1 + cy_1 \\ \psi_2 &= a + bx_2 + cy_2 \\ \psi_3 &= a + bx_3 + cy_3 \end{aligned} \quad (4.15)$$

Układ ten można przepisać w notacji macierzowej

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} \quad (4.16)$$

i a następnie znaleźć jego rozwiązanie

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}. \quad (4.17)$$

Po wstawieniu rozwiązania (4.17) do równania (4.14)

$$\psi = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} \quad (4.18)$$

można zauważyć, że (4.18) da się przepisać w postaci kombinacji liniowej wielomianów a_i

$$\psi = \sum_{i=1}^3 \psi_i \alpha_i(x, y), \quad (4.19)$$

gdzie

$$\alpha_i(x, y) = \frac{1}{2A} (a_i + b_i x + c_i y) \quad (i = 1, 2, 3), \quad (4.20)$$

oraz (A jest polem powierzchni elementu)

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = x_1 b_1 + x_2 b_2 + x_3 b_3. \quad (4.21)$$

$$\begin{aligned} a_i &= x_j y_k - x_k y_j \\ b_i &= y_j - y_k \\ c_i &= x_k - x_j \end{aligned} \quad \text{dla} \quad \begin{cases} i = 1, j = 2, k = 3 \\ i = 2, j = 3, k = 1 \\ i = 3, j = 1, k = 2 \end{cases} \quad (4.22)$$

Wyrażenie (4.22) oznacza po prostu cykliczne przechodzenie indeksów przy obliczaniu współczynników a_i , b_i , c_i w wielomianie a_i dla i -tego węzła. Jak wspomniano wcześniej, przedstawione tu rozwiązanie zagadnienia brzegowego MES przeprowadzane jest z wykorzystaniem metody Galerkina, a w metodzie tej jako funkcje testowe (wagowe) wprowadzone wcześniej wykorzystuje się właśnie wielomiany aproksymujące w wyprowadzonej powyżej postaci

$$T_s = \alpha_j(x, y) \quad (j = 1, 2, 3). \quad (4.23)$$

Dzięki temu lewą stronę (4.13) równania dla pojedynczego elementu (oznaczonego symbolicznie jako obszar Δ) można zapisać jako

$$\iint_{\Delta} (\vec{\nabla}_{\perp} T_s \cdot \vec{\nabla}_{\perp} \psi) dx dy = \sum_{i=1}^3 \psi_i \iint_{\Delta} (\vec{\nabla}_{\perp} \alpha_i \cdot \vec{\nabla}_{\perp} \alpha_j) dx dy \quad (j = 1, 2, 3) \quad (4.24)$$

a prawą w postaci

$$\iint_{\Delta} k_{\perp}^2 (T_s \psi) dx dy = k_{\perp}^2 \sum_{i=1}^3 \psi_i \iint_{\Delta} (\alpha_i \cdot \alpha_j) dx dy \quad (j = 1, 2, 3). \quad (4.25)$$



W równaniu (4.25) poprzeczną stałą propagacji k_{\perp}^2 można wyłączyć przed całkę, ponieważ w obrębie poszczególnych elementów przyjmuje ona wartości stałe (tak jak i współczynnik załamania od którego zależy). Zatem jak widać na podstawie (24, 25) równanie (4.13) zapisane dla pojedynczego elementu przybiera postać

$$\sum_{i=1}^3 \psi_i \iint_{\Delta} (\vec{\nabla}_{\perp} \alpha_i \cdot \vec{\nabla}_{\perp} \alpha_j) dx dy = k_{\perp}^2 \sum_{i=1}^3 \psi_i \iint_{\Delta} (\alpha_i \cdot \alpha_j) dx dy \quad (j = 1, 2, 3), \quad (4.26)$$

a w postaci macierzowej

$$[S_{el}] [\psi] = k_{\perp}^2 [T_{el}] [\psi], \quad (4.27)$$

gdzie

$$[S_{el}] = \iint_{\Delta} (\vec{\nabla}_{\perp} \alpha_i \cdot \vec{\nabla}_{\perp} \alpha_j) dx dy \quad (j = 1, 2, 3) \quad (4.28)$$

oraz

$$[T_{el}] = \iint_{\Delta} (\alpha_i \alpha_j) dx dy \quad (j = 1, 2, 3). \quad (4.29)$$

W zapisie (4.28) i (4.29) indeks *el* oznacza, że dotyczy macierzy wyznaczonych dla pojedynczego elementu. Wiedząc, że dla funkcji bazowych α_i gradient przyjmuje postać

$$\vec{\nabla}_{\perp} \alpha_i = \frac{\partial \alpha_i}{\partial x} \hat{x} + \frac{\partial \alpha_i}{\partial y} \hat{y} = \frac{1}{2A} (b_i \hat{x} + c_i \hat{y}), \quad (4.30)$$

elementy macierzy

$$[\vec{\nabla}_{\perp} \alpha_i \cdot \vec{\nabla}_{\perp} \alpha_j] = \begin{bmatrix} \vec{\nabla}_{\perp} \alpha_1 \cdot \vec{\nabla}_{\perp} \alpha_1 & \vec{\nabla}_{\perp} \alpha_1 \cdot \vec{\nabla}_{\perp} \alpha_2 & \vec{\nabla}_{\perp} \alpha_1 \cdot \vec{\nabla}_{\perp} \alpha_3 \\ \vec{\nabla}_{\perp} \alpha_2 \cdot \vec{\nabla}_{\perp} \alpha_1 & \vec{\nabla}_{\perp} \alpha_2 \cdot \vec{\nabla}_{\perp} \alpha_2 & \vec{\nabla}_{\perp} \alpha_2 \cdot \vec{\nabla}_{\perp} \alpha_3 \\ \vec{\nabla}_{\perp} \alpha_3 \cdot \vec{\nabla}_{\perp} \alpha_1 & \vec{\nabla}_{\perp} \alpha_3 \cdot \vec{\nabla}_{\perp} \alpha_2 & \vec{\nabla}_{\perp} \alpha_3 \cdot \vec{\nabla}_{\perp} \alpha_3 \end{bmatrix} \quad (4.31)$$

da się zapisać bezpośrednio w oparciu o współczynniki funkcji bazowych dla każdego węzła

$$[\vec{\nabla}_{\perp} \alpha_i \cdot \vec{\nabla}_{\perp} \alpha_j] = \frac{1}{4A^2} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix} \quad (4.32)$$

a zatem

$$[S_{el}] = A [\nabla_{\perp} \alpha_i \cdot \nabla_{\perp} \alpha_j]. \quad (4.33)$$

Podobnie macierz (4.29) upraszcza się do postaci [1–4]

$$[T_{el}] = \frac{A}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}. \quad (4.34)$$

Uwzględniając w układzie równań (4.27) postać (4.3) poprzecznej stałej propagacji dla ośrodków niejednorodnych otrzyma się

$$[S_{el}] [\psi] = (k_0^2 n_{el}^2 - \beta^2) [T_{el}] [\psi], \quad (4.35)$$

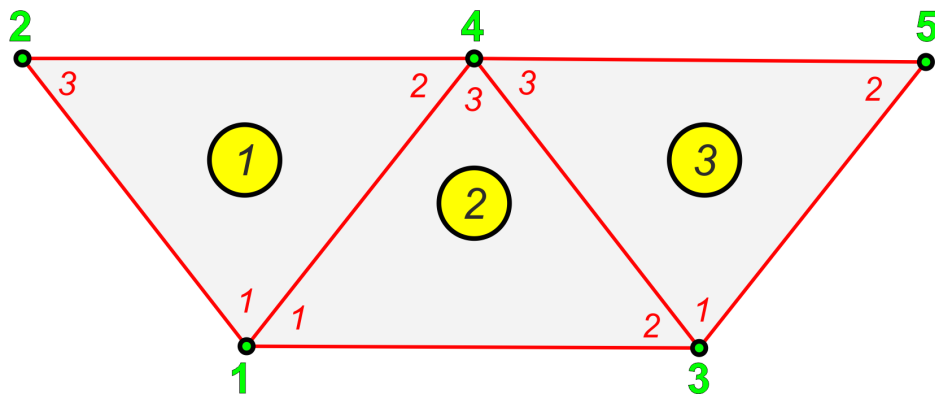
gdzie n_{el} to współczynnik załamania dla danego elementu. Na koniec układ równań (4.35) przekształca się do postaci

$$[C_{el}] [\psi] = \beta^2 [T_{el}] [\psi], \quad (4.36)$$

macierzowego problemu własnego z wartościami własnymi reprezentowanymi przez podłużną stałą propagacji b oraz wektorami własnymi y stanowiącymi przybliżenie (o czym wspomniano wcześniej) rozwiązania dokładnego nad obszarem danego elementu, gdzie

$$[C_{el}] = k_0^2 n_{el}^2 [T_{el}] - [S_{el}]. \quad (4.37)$$

Dzięki takiemu zapisowi (podłużna stała propagacji b nie zmienia się w przekroju poprzecznym falowodu) możliwe będzie ułożenie globalnego układu równań.



Rysunek 4.3. Przykład ilustrujący scalanie układów równań [1]. Węzły lokalne numerowane są przeciwnie do ruchu wskazówek zegara, przy czym wybór pierwszego węzła jest dowolny. Dowolna jest również kolejność numeracji elementów i węzłów w skali globalnej.

4.3.3. Scalenie i rozwiązanie układu równań

Ostatnie dwa etapy metody elementów skończonych to scalenie układów równań dla pojedynczych elementów w układ globalny i rozwiązanie powstałego w ten sposób algebraicznego zagadnienia własnego.

Celem scalania (asemblacji) lokalnych (tj. związanych z pojedynczymi elementami) układów równań do układu globalnego jest zagwarantowanie ciągłości rozwiązania pomiędzy sąsiadującymi elementami oraz uwzględnienia warunków brzegowych (jako zewnętrzne sąsiedztwo elementów znajdujące się na brzegu obszaru obliczeniowego przyjmuje się warunki brzegowe). Aby lepiej zrozumieć proces scalania należy posłużyć się uproszczonym przykładem, zilustrowanym na Rys. 4.3. [1]. Należy zwrócić uwagę, że węzły lokalne numerowane są przeciwnie do ruchu wskazówek zegara (jest to dosyć istotne, ponieważ numeracja zgodna z ruchem wskazówek zegara zmienia znak wyrażenia (21) na ujemny), przy czym wybór pierwszego węzła jest dowolny. Dowolna jest również kolejność numeracji elementów i węzłów w skali globalnej. Wynikają one z wyboru algorytmów triangulacji wykorzystywanych przez generatory siatek.

W rozpatrywanym przykładzie obszar rozwiązania został podzielony na trzy elementy trójkątne leżące pomiędzy pięcioma punktami węzłowymi. W związku z tym macierz współczynników (4.37) ma wymiary 5×5 (dokładniej, liczba równań w układzie jest równa liczbie stopni swobody układu a ta z kolei równa jest liczbie węzłów pomnożonej przez liczbę stopni swobody pojedynczego węzła czyli liczbę niewiadomych do wyznaczenia w tym węźle – w rozpatrywanym przypadku liczba stopni swobody każdego węzła wynosi 1)

$$[C] = \begin{bmatrix} C_{11} & \cdots & C_{15} \\ \vdots & \ddots & \vdots \\ C_{51} & \cdots & C_{55} \end{bmatrix} \quad (4.38)$$

a jej poszczególne elementy traktujemy jako współczynniki sprzężenia pomiędzy poszczególnymi węzłami [1, 2]. Jak wspomniano, w rozpatrywanym przypadku rozwiązanie (tzn. funkcja y) w każdym węźle musi być ciągła. Wykorzystuje się ten fakt do wyznaczenia elementów C_{ij} macierzy współczynników (4.38): sprzężenie i -tego węzła z j -tym wyznacza się jako sumę wkładów pochodzących od wszystkich elementów zawierających węzły i oraz j . Jest proste zadanie ponieważ w obrębie każdego elementu (el to numer elementu)

macierz C_{el} ma również postać

$$C_{el} = \begin{bmatrix} C_{11}^{(el)} & C_{12}^{(el)} & C_{13}^{(el)} \\ C_{21}^{(el)} & C_{22}^{(el)} & C_{23}^{(el)} \\ C_{31}^{(el)} & C_{23}^{(el)} & C_{33}^{(el)} \end{bmatrix} \quad (4.39)$$

uwzględniającą zależności (sprzężenia) pomiędzy poszczególnymi węzłami wewnętrznymi elementu. Zatem jak wynika z Rys. 4.3., współczynnik C_{11} w macierzy (4.38) będzie miał postać

$$C_{11} = C_{11}^{(1)} + C_{11}^{(2)}, \quad (4.40)$$

ponieważ dotyczy węzła o numerze 1, który jest węzłem wspólnym dla elementów 1 i 2 (a w numeracji wewnętrznej obydwu elementów węzeł ten również ma numer 1). Węzeł 2 należy tylko do elementu pierwszego i ma lokalny (w obrębie tego elementu) numer 3, zatem

$$C_{22} = C_{33}^{(1)}. \quad (4.41)$$

Węzeł 4 należy do elementów 1, 2 i 3, stąd element C_{44} macierzy (4.38) przyjmie postać

$$C_{44} = C_{22}^{(1)} + C_{33}^{(2)} + C_{33}^{(3)}. \quad (4.42)$$

Sprzężenie węzła 1 z węzłem 4 dzięki temu, że oba węzły są wspólne dla elementów 1 i 2, zostanie zapisane jako

$$C_{14} = C_{12}^{(1)} + C_{13}^{(2)}. \quad (4.43)$$

a węzła 4 z 1

$$C_{41} = C_{21}^{(1)} + C_{31}^{(2)}. \quad (4.44)$$

Widać w tym miejscu, że zachodzi relacja

$$C_{14} = C_{41} \quad (4.45)$$

ponieważ macierze sprzężeń lokalnych C_{el} są symetryczne - wynika to ze wzorów (31, 34 37). Ponieważ istnieją też węzły, które nie mają bezpośredniego sprzężenia, część elementów macierzy (4.38) przyjmie po prostu wartość zerową, przykładowo

$$C_{23} = C_{32} = 0. \quad (4.46)$$

Ostatecznie macierz (4.38) po uwzględnieniu wszystkich zależności przyjmie postać

$$C = \begin{bmatrix} C_{11}^{(1)} + C_{11}^{(2)} & C_{13}^{(1)} & C_{12}^{(2)} & C_{12}^{(1)} + C_{13}^{(2)} & 0 \\ C_{31}^{(1)} & C_{33}^{(1)} & 0 & C_{32}^{(1)} & 0 \\ C_{21}^{(2)} & 0 & C_{22}^{(2)} + C_{11}^{(3)} & C_{23}^{(2)} + C_{13}^{(3)} & C_{12}^{(3)} \\ C_{21}^{(1)} + C_{31}^{(2)} & C_{23}^{(1)} & C_{32}^{(2)} + C_{31}^{(3)} & C_{22}^{(1)} + C_{33}^{(2)} + C_{33}^{(3)} & C_{32}^{(3)} \\ 0 & 0 & C_{21}^{(3)} & C_{23}^{(3)} & C_{22}^{(3)} \end{bmatrix}. \quad (4.47)$$

Analogicznie wyprowadza się macierz \mathbf{T} po prawej stronie równania (4.36) co prowadzi do globalnego układu równań

$$[C][\psi] = \beta^2 [T][\psi]. \quad (4.48)$$

Można zauważyć, że gdy liczba elementów będzie rosła, sytuacja zapisana wzorem (4.46), czyli brak sprzężenia pomiędzy węzłami będzie coraz częstsza. Doprowadzi to do



sytuacji, w której macierze w równaniu (4.48) będą tzw. macierzami rzadkimi. Ponieważ MATLAB [5] umożliwia prowadzenie obliczeń na tego typu macierzach, zostanie to wykorzystane w kodzie programu zaproponowanego poniżej.

W tym miejscu można przystąpić do rozwiązania algebraicznego problemu własnego (4.48), co jest celem wszystkich kroków wykonanych w ramach opisanego algorytmu MES. Samo rozwiązanie, podobnie jak generowanie siatki, nie stanowi tematu tego rozdziału. Podobnie jak w przypadku siatek oraz podczas późniejszej wizualizacji wyników, do rozwiązania algebraicznego zagadnienia własnego zostaną wykorzystane funkcje środowiska MATLAB.

Na zakończenie warto przytoczyć jeszcze dwie użyteczne formuły

$$\frac{d\psi}{dx} = \frac{1}{2A} \sum_{i=1}^3 \psi_i b_i, \quad (4.49)$$

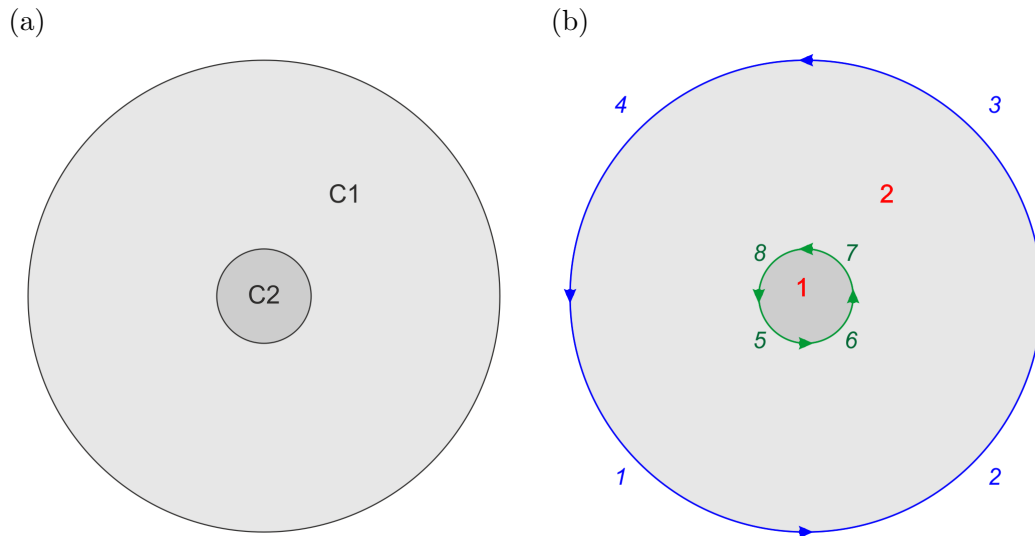
$$\frac{d\psi}{dy} = \frac{1}{2A} \sum_{i=1}^3 \psi_i c_i, \quad (4.50)$$

które umożliwiają wyznaczenie pochodnych cząstkowych potencjału y po rozwiązaniu zagadnienia własnego (4.48).

4.4. Przykładowy program

Poniższy przykład dotyczy rozwiązania skalarnego równania falowego (4.5) dla modów TE w światłowodzie o przekroju cylindrycznym. Funkcja y w równaniu (4.5) to składowa poprzeczna pola elektrycznego. Ze względu na nieskończoną symetrię obrotową przekroju poprzecznego (czyli brak wyróżnionego kierunku) światłowodu, nie da się dokładnie określić zorientowania kierunku tej składowej względem przyjętego układu współrzędnych kartezjańskich. Innymi słowy, jeśli przyjmie się oznaczenie, że jest to składowa E_y , układ „sam zadecyduje” jaki kierunek będzie miała ta składowa. Dodatkowo należy podkreślić, że przytoczony przykład nie wymaga jawnej implementacji warunków brzegowych – jak wspomniano wcześniej dla przypadku TE warunki brzegowe PMC są spełnione w sposób naturalny (w przypadku warunków TM należałoby usunąć te wiersze i kolumny macierzy układu (48) które odpowiadają węzłom brzegowym). Poniżej zamieszczono przykład implementacji tego zadania.

W pierwszym kroku definiowana jest geometria problemu. Światłowód cylindryczny (tzn. jego przekrój poprzeczny) można przedstawić jako dwa współśrodkowe okręgi $C1$ i $C2$ o różnych promieniach. Aby wykorzystać do obliczeń pakiet *Partial Differential Equation Toolbox* należy dostosować rozwiązywany problem do przyjętej w ramach tego pakietu nomenklatury (jest to tzw. model CSG - *constructive solid geometry* [5]), mianowicie trzeba zdefiniować trzy zmienne: gd , ns oraz sf a następnie wykorzystać je jako argumenty funkcji *decsf*. Zmienna gd (od *geometry description matrix*) zawiera w każdej kolumnie informacje o poszczególnych elementach geometrii w modelu CSG. W analizowanym przypadku elementy to dwa okręgi $C1$ (płaszczyzna o promieniu 20 mm) i $C2$ (rdzeń o promieniu 4 mm). Okrąg kodowany jest flagą o wartości 1 w pierwszym wierszu, w kolejnych wierszach znajdują się współrzędne x i y środka okręgu i jego promień. Elementy CSG muszą posiadać nazwy. Nazwy kodowane są w zmiennej ns (od *name space matrix*), której poszczególne kolumny odpowiadają kolumnom zmiennej gd i zawierają łańcuchy znaków (czyli ‘C1’ i ‘C2’) transponowane do postaci kolumnowej i przekonwertowane na typ double (czyli wynikiem będzie macierz [67, 67; 49, 50]).

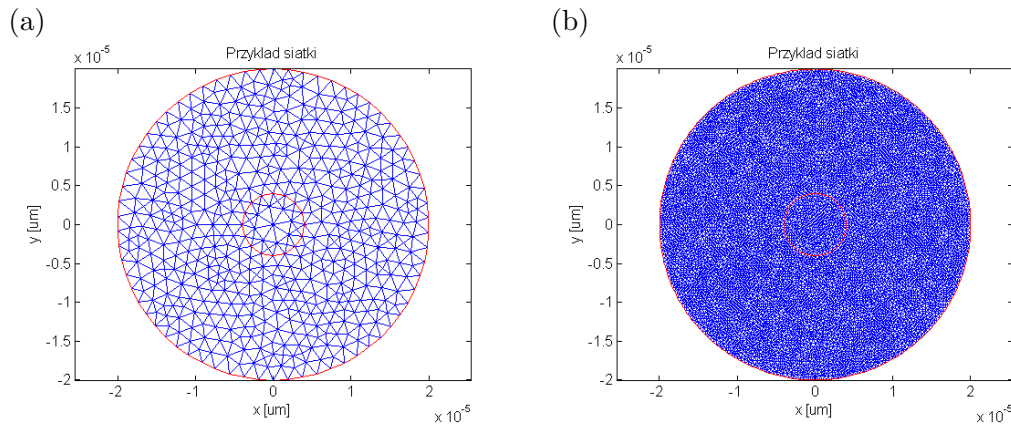


Rysunek 4.4. Przekrój poprzeczny światłowodu reprezentowany przez dwa współśrodkowe okręgi $C1$ i $C2$ (a). Efektu wywołania funkcji *decsg*: zamiana okręgów $C1$ i $C2$ na podobszary minimalne 1 i 2 (kolor czerwony) rozdzielone krawędziami 1 – 8 (kolor niebieski i zielony)

```
% Czyszczenie pamięci i ekranu konsoli
clear all; clc;
% Definiowanie zmiennych opisujących geometrie:
% pierwszy okrag (plaszcz) – C1
gd(:, 1) = [1; 0; 0; 20.0e-6];
% drugi okrag (rdzen) – C2
gd(:, 2) = [1; 0; 0; 4.0e-6];
% Ich nazwy ('C1' i 'C2') sa rozwijane w kolumny
% a nastepnie konwertowane na typ double.
ns = double(reshape(['C1', 'C2'], 2, []));
% Podanie formuly wiazacej elementy geometrii
% w analizowanym przypadku jest to suma dwóch obszarow
sf = 'C1+C2';
% Podzial geometrii na obszary minimalne
dl = decsg(gd, sf, ns);
```

Relacje, które wiążą elementy CSG (czyli suma, różnica bądź część wspólna zbiorów) kodowane są w zmiennej *sf* (od *set formula*). W analizowanym przypadku zbiorem wynikowym jest to suma okręgów $C1$ i $C2$ (czyli ' $C1 + C2$ '). Funkcja *decsg* (nazwa od *decompose constructive solid geometry*) pozwala w oparciu o opisane powyżej zmienne przechowujące geometrię problemu utworzyć zbiór minimalnych rozłącznych obszarów ograniczonych granicami zewnętrznymi i wewnętrznymi.

W analizowanym przypadku zbiór ten zawierać będzie w zmiennej *dl* dwa podobszary ograniczone ośmioma krawędziami, jak zilustrowano to na Rys. 4.4.. Krawędzie są łukami odpowiednich ćwiartek okręgów $C1$ i $C2$. Zmienna *dl* w każdej kolumnie zawiera informacje o poszczególnych krawędziach. W pierwszym wierszu znajduje się flaga charakteryzująca kształt krawędzi (łuk okręgu - 1, prosta - 2, łuk elipsy - 4), w kolejnych wierszach znajdują się współrzędne x początku i końca krawędzi oraz współrzędne y początku i końca. Pozostałe wiersze w zależności od kształtu krawędzi zawierają dodatkowe informacje (np. współrzędne środka i promień krzywizny dla łuku okręgu) – więcej szczegółów można znaleźć w obszernym temacie pomocy dla funkcji *decsg*.



Rysunek 4.5. Wizualizacja podziału obszaru rozwiązania na elementy trójkątne przy różnych wartościach parametru h_{max} określającego maksymalną długość krawędzi elementu; (a) $h_{max} = 2.0$ mm, siatka zawiera 1020 elementów, (b) $h_{max} = 0.5$ mm, siatka zawiera 17414 elementów.

```
% Generowanie siatki
[p, e, t] = initmesh(dl, 'hmax', 2e-6);
% Wyświetlenie obszaru rozwiązania i wygenerowanej siatki
set(figure(1), 'Color', 'w')
pdemesh(p,e,t); title('Przykład siatki');
axis equal; xlabel('x [um]'); ylabel('y [um]')
% Zapisanie zmiennych p i t na dysku
save('p.mat', 'p', '-mat'); save('t.mat', 't', '-mat');
```

Następnie wywoływana jest funkcja *initmesh*, której argumentem jest zbiór podobszarów *dl* a wynikiem opisane wcześniej macierze *p*, *e* i *t* zawierające informacje o punktach i elementach będących wynikiem trinarularyzacji. Funkcja *initmesh* posiada również kilka argumentów opcjonalnych a wśród nich *hmax*, który określa maksymalną długość krawędzi wygenerowanych elementów. Na rysunku 5 pokazano siatki uzyskane dla dwóch wartości tego parametru, $h_{max} = 2.0$ mm (Rys. 4.5. a, 1020 elementów) oraz $h_{max} = 0.5$ mm (Rys. 4.5. b, 17414 elementów). Do wizualizacji wygenerowanych siatek służy funkcja *pdemesh*. W dalszych obliczeniach wykorzystano gęstszą siatkę.

Po zdefiniowaniu geometrii i wygenerowaniu siatki można przystąpić do właściwej części rozwiązania problemu, tj. implementacji procedury MES. W pierwszej kolejności definiowane są wartości parametrów rozwiązywanego zadania.

```
% Czyszczenie pamięci i ekranu konsoli
clear all; clc;
% Długość fali
lambda = 1.00*1e-6;
% Przenikalności dielektryczne podobszarów
Epsilon = [1.45, 1.44].^2;
% Kwadrat modułu wektora falowego
kos2 = (2*pi/lambda)^2;
% Macierz (4.34) dla pojedynczego elementu
Tsel = [2, 1, 1; 1, 2, 1; 1, 1, 2]/12;
```

Następnie ładowane są do pamięci zmienne *p* i *t*, wygenerowane uprzednio opisanym sposobem i zapisane na dysku, zawierające punkty i elementy.



```
% Zmienna p zawierająca współrzędne x i y kolejnych węzłów  
load('p.mat');  
% Zmienna t zawierająca numery węzłów należące do kolejnych  
% elementów oraz numery podobszarów do których należą elementy  
load('t.mat');
```

Potem pobierane są liczby elementów i węzłów siatki oraz prealokowane macierze rzadkie \mathbf{C} i \mathbf{T} z równania (4.48) i odpowiadające im macierze pomocnicze \mathbf{C}_{el} i \mathbf{T}_{el} z równania (4.36) dla pojedynczych elementów.

```
% Liczba elementów  
nElems = size(t, 2);  
% Liczba węzłów  
nPoints = size(p, 2);  
% Prealokacja macierzy dla globalnego układu równań(4.48)  
C = sparse(nPoints, nPoints);  
T = sparse(nPoints, nPoints);  
% Prealokacja macierzy (4.33), (4.34) i (4.37) dla  
% pojedynczego elementu  
Tel = zeros(4,3);  
Sel = zeros(4,3);  
Cel = zeros(4,3);
```

Kolejną część programu stanowi pętla *for* przebiegająca po numerach elementów przy czym zmienna *elem_nr* reprezentuje numer bieżącego elementu. Ze zmiennej *t* (która jest macierzą o wymiarach $5 \times nElems$) w każdym obiegu pętli pobierany jest numer podobszaru, który jest następnie wykorzystywany do wyznaczenia kwadratu współczynnika załamania *eps* w obrębie bieżącego elementu. Ze zmiennej *t* pobierane są również indeksy węzłów w notacji globalnej *wezel_1*, *wezel_2* i *wezel_3*, odpowiadające pierwszemu, drugiemu i trzeciemu węzłowi w notacji lokalnej (czyli w obrębie bieżącego elementu).



```
for elem_nr = 1 : nElems
% Ustalenie numeru podobszaru i odpowiadajej mu wartosci
% przenikalnosci dielektrycznej
eps = Epsilon(t(4, elem_nr));
% Pobranie globalnych numerow wezlow odpowiadajacych
% numerom 1, 2 i 3 w obrebie biezacego elementu
wezel_1 = t(1, elem_nr);
wezel_2 = t(2, elem_nr);
wezel_3 = t(3, elem_nr);
% Wspolczynniki b i c dla biezacego elementu, wzor (4.22)
b1 = p(2, wezel_2) - p(2, wezel_3);
b2 = p(2, wezel_3) - p(2, wezel_1);
b3 = p(2, wezel_1) - p(2, wezel_2);
c1 = p(1, wezel_3) - p(1, wezel_2);
c2 = p(1, wezel_1) - p(1, wezel_3);
c3 = p(1, wezel_2) - p(1, wezel_1);
% Pole powierzchni elementu, wzor (4.21)
A = 0.5*(p(1, wezel_1)*b1 + ...
p(1, wezel_2)*b2 + p(1, wezel_3)*b3);
% Wyznaczenie macierzy Sel, wzor (4.33)
Sel(1, 1) = b1*b1 + c1*c1;
Sel(1, 2) = b1*b2 + c1*c2;
Sel(1, 3) = b1*b3 + c1*c3;
Sel(2, 1) = b2*b1 + c2*c1;
Sel(2, 2) = b2*b2 + c2*c2;
Sel(2, 3) = b2*b3 + c2*c3;
Sel(3, 1) = b3*b1 + c3*c1;
Sel(3, 2) = b3*b2 + c3*c2;
Sel(3, 3) = b3*b3 + c3*c3;
% Wyznaczenie macierzy Tel, wzor (4.34)
Tel = A*Tsel;
% Wyznaczenie macierzy Cel, wzor (4.37)
Cel = eps*kos*Tel - Sel/(4*A);
% Zlozenie macierzy T globalnego ukladu rownan, wzor (4.48)
T(wezel_1, wezel_1) = T(wezel_1, wezel_1) + Tel(1, 1);
T(wezel_2, wezel_1) = T(wezel_2, wezel_1) + Tel(2, 1);
T(wezel_3, wezel_1) = T(wezel_3, wezel_1) + Tel(3, 1);
T(wezel_1, wezel_2) = T(wezel_1, wezel_2) + Tel(1, 2);
T(wezel_2, wezel_2) = T(wezel_2, wezel_2) + Tel(2, 2);
T(wezel_3, wezel_2) = T(wezel_3, wezel_2) + Tel(3, 2);
T(wezel_1, wezel_3) = T(wezel_1, wezel_3) + Tel(1, 3);
T(wezel_2, wezel_3) = T(wezel_2, wezel_3) + Tel(2, 3);
T(wezel_3, wezel_3) = T(wezel_3, wezel_3) + Tel(3, 3);
```

```
% Złożenie macierzy C globalnego układu równań, wzor (4.48)
```

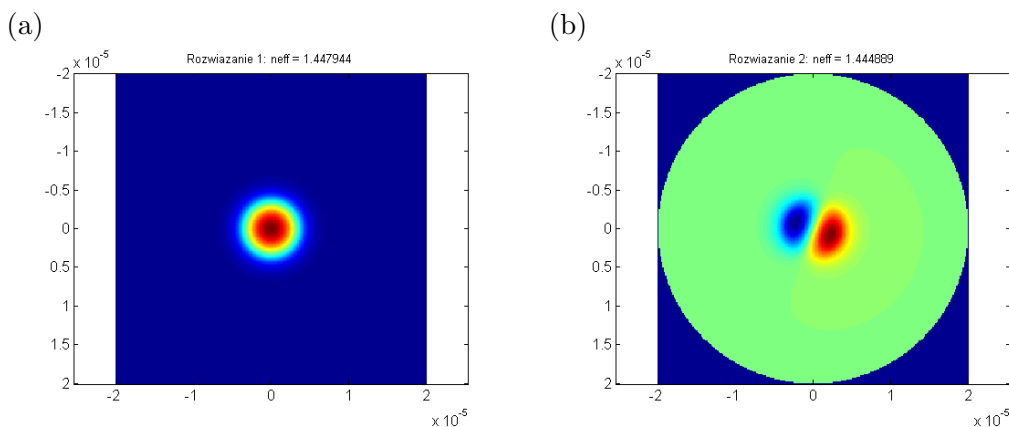
```
C(wezel_1, wezel_1) = C(wezel_1, wezel_1) + Cel(1, 1);  
C(wezel_2, wezel_1) = C(wezel_2, wezel_1) + Cel(2, 1);  
C(wezel_3, wezel_1) = C(wezel_3, wezel_1) + Cel(3, 1);  
C(wezel_1, wezel_2) = C(wezel_1, wezel_2) + Cel(1, 2);  
C(wezel_2, wezel_2) = C(wezel_2, wezel_2) + Cel(2, 2);  
C(wezel_3, wezel_2) = C(wezel_3, wezel_2) + Cel(3, 2);  
C(wezel_1, wezel_3) = C(wezel_1, wezel_3) + Cel(1, 3);  
C(wezel_2, wezel_3) = C(wezel_2, wezel_3) + Cel(2, 3);  
C(wezel_3, wezel_3) = C(wezel_3, wezel_3) + Cel(3, 3);  
end;
```

Dzięki znajomości numerów węzłów możliwe jest pobranie ich współrzędnych ze zmiennej p , w której indeks kolumny jest numerem węzła. Współrzędne umożliwiającą z kolei obliczenie współczynników interpolacji b_1, b_2, b_3 i c_1, c_2, c_3 z równania (4.22). Współczynniki te są służyć do wyznaczenia pola powierzchni elementu A ze wzoru (4.21) oraz macierzy S_{el} ze wzoru (4.33). Następnie oblicza się wartości elementów macierzy T_{el} (4.34) i C_{el} (4.37). Macierze te zawierające współczynniki sprzężenia między elementami i są z kolei wykorzystywane do wyznaczenia globalnych macierzy sprzężeń metodą opisaną w punkcie 3.3.

Uzyskane w ten sposób algebraiczne zagadnienie własne może być rozwiązane za pomocą wbudowanej procedury *eigs*. Procedura ta jako argumenty przyjmuje dwie macierze uogólnionego zagadnienia własnego, liczbę wartości własnych do wyznaczenia (podano liczbę 20) oraz tzw. przesunięcie czyli wartość, w pobliżu której poszukiwane będą wartości własne.

```
[V, eigVals] = eigs(C, T, 20, kos2*Epsilon(1));  
neff = diag(sqrt(eigVals/kos));  
save('sols.mat', 'V', 'neff', '-mat');
```

Po rozwiązaniu na głównej przekątnej macierzy *eigVals* zapisywane są znalezione wartości własne a w kolejnych kolumnach macierzy V odpowiadające im wektory własne. Otrzymane wartości własne są zamieniane na efektywne współczynniki załamania – wartości bardziej użyteczne z punktu widzenia rozpatrywanego problemu. Uzyskane wyniki są zapisywane na dysku w pliku *sols.mat*.



Rysunek 4.6. Przykładowe wyniki symulacji uzyskane dla przytoczonych w przykładowym kodzie źródłowym wartości parametrów (tj. długość fali 1 mm, współczynnik załamania rdzenia 1.45, współczynnik załamania płaszczka 1.44, promień rdzenia 4 mm, promień płaszczka 20 mm)



```
% Czyszczenie pamieci i ekranu konsoli
clear all; clc;
% Zmienna p zawierajaca wspolrzedne x i y kolejnych wezlow
load('p.mat');
% Zmienna t zawierajaca numery wezlow nalezace do kolejnych
% elementow oraz numery podobszarow do których naleza elementy
load('t.mat');
% Zmienne V i neff zawierajace rozwiazania
load('sols.mat');
% Wyznaczenie granic obszaru obliczeń
zakres_x = [min(p(1, :)), max(p(1, :))];
zakres_y = [min(p(2, :)), max(p(2, :))];
% Liczba punktow
liczba_punktow = 250;
% Siatka punktow w układzie kartezjanskim
x = linspace(zakres_x(1), zakres_x(4.2), liczba_punktow);
y = linspace(zakres_y(1), zakres_y(4.2), liczba_punktow);
% Numer rozwiazania do wizualizacji
nr_rozw = 1;
% Wektor własny odpowiadajacy numerowi rozwiazania
u = V(:, nr_rozw);
% Wyznaczenie wartosci rozwiazan w wezlach siatki xy
uxy = tri2grid(p, t, u, x, y);
% Wyszwietlenie rozwiazania
set(figure(nr_rozw), 'Color', 'w');
imagesc(x, y, uxy);
axis equal;
```

Ostatnim etapem jest wizualizacja wyników, której przykład zaproponowana powyżej. Ponieważ uzyskane wektory własne zawierają wartości rozwiązania w węzłach elementów trójkątnych, wizualizacji wyników nie da się przeprowadzić bezpośrednio. Wykorzystuje się w tym celu wbudowaną funkcję MATLABA *tri2grid* [5], która w oparciu o interpolację wyznacza wartości rozwiązania w węzłach siatki zdefiniowanej przez wektory x i y . Przykładowe wyniki (dwa pierwsze rozwiązania) dla parametrów obliczeniowych zaproponowanych w kodzie źródłowym przedstawiono na Rys. 4.6..

4.5. Podsumowanie

W rozdziale tym zaprezentowane zostały podstawy wykorzystania metody elementów skończonych do rozwiązywania problemów elektromagnetycznych na przykładzie skalarne równania falowego. Rozwiązanie przeprowadzono metodą Galerkina. Przedstawiono przykładową implementację rozwiązania w środowisku MATLAB aby ułatwić czytelnikowi opracowanie własnego programu. Po zaimplementowaniu powyższych procedur dobrze jest przeprowadzić obliczenia dla jednego z przykładów zawartych w referencjach [1–4] (zwłaszcza pozycja [3] będzie wyjątkowo użyteczna) aby upewnić się, że opracowany program jest wolny od błędów. W celu uzupełnienia i poszerzenia wiedzy przedstawionej w skrótovej formie w tym rozdziale, warto sięgnąć na początek do pozycji [1] a następnie do podręcznika [2], który najobszerniej traktuje temat zastosowania metody elementów skończonych w elektromagnetyzmie.



4.6. Zadania

1. Skopiuj powyższy kod do trzech plików (pierwszy – generacja siatki, drugi – procedura MES, trzeci – wizualizacja wyników). Uruchom procedury i porównaj uzyskane wyniki z wynikami zawartymi w tabeli 1 oraz na rysunkach 5 i 6.
2. Napisz procedurę wyświetlającą rozkład współczynnika załamania dla omówionego przykładu.
3. Zapoznaj się z dokładnym opisem i dodatkowymi parametrami funkcji *initmesh* i *refinemesh*. Przebadaj wpływ liczby elementów na rozwiązanie (np. wartość efektywnego współczynnika załamania modu podstawowego).
4. Zaproponuj procedurę pozwalającą wyznaczyć pochodne (wzory 49 – 50) uzyskanych w rozwiązaniu rozkładów pól.
5. Opracuj procedurę wyznaczającą mody TM dla omówionego przykładu światłowodu cylindrycznego.
6. Zmodyfikuj program tak, aby można było przy jego pomocy wyznaczyć rozwiązania dla równania Laplace'a (UWAGA: zadanie to wymaga dodatkowo opracowania umiejętności jawnego wprowadzania warunków brzegowych).



Bibliografia

- [1] M. N. O. Sadiku *Numerical Techniques in Electromagnetics*, Rozdział 6, CRC Press; 2 edition (July 12, 2000).
- [2] Jian-Ming Jin *The Finite Element Method in Electromagnetics*, Wiley-IEEE Press; 2 edition (May 27, 2002).
- [3] M. K. Haldar *Introducing the Finite Element Method in electromagnetics to undergraduates using MATLAB*, International Journal of Electrical Engineering Education 45, str. 232-244 (2006).
- [4] C. J. Reddy, M. D. Deshpande, C. R. Cockrell, and F.B. Beck *Finite Element Method for Eigenvalue Problems in Electromagnetics*, NASA Technical Paper 3485, str. 1-38 (1994).
- [5] MATLAB, <http://www.mathworks.com>.