🔢

2. Metody numeryczne w technice

Status	Zrobione
Assign	

▼ a) metody rozwiązywania zagadnień nieliniowych

Zagadnienia nieliniowe rozwiązuje się przy użyciu metod numerycznych nazywanych ***metodami kolejnych przybliżeń*** lub ***metodami iteracyjnymi***. Polegają one na tym, że na określonym przedziale poszukiwane jest **przybliżone położenie pierwiastka**. W kolejnych krokach **powtarza się obliczenia** uzyskując coraz dokładniejsze położenie tego pierwiastka, **aż do utrzymania zadowalająco niskiego błędu** (który również powinien być szacowany przez stosowaną metodę).

Wśród metod rozwiązywania zagadnień nieliniowych można wymienić metodę bisekcji, metodę regula-falsi, metodę siecznych, metodę Newtona oraz wszelkie modyfikacje metody Newtona.

Metoda bisekcji

W metodzie bisekcji zakłada się, że funkcja jest ciągła na analizowanym przedziale <a,b>, natomiast **granice tego przedziału - a i b - mają przeciwne znaki**. Przeciwne znaki oznaczają, że na tym przedziale funkcja musi przecinać zero. **Metoda ta znajduje tylko jeden pierwiastek funkcji**, nawet jeżeli w zadanym przedziale jest ich kilka.

Na czym polega metoda bisekcji?

1. Oblicza się wartość funkcji w środku przedziału <a,b>, czyli $f(x_1 = \frac{a+b}{2})$.
   - jeżeli wartość w tym punkcie jest równa 0 lub wystarczająco bliska 0, mamy pierwiastek (miejsce zerowe)
   - jeżeli nie, przechodzimy do kolejnego kroku
2. Mamy dwa przedziały, czyli $< a, x_1 >; < x_1, b >$, czyli przedziały od punktu a do centrum całego badanego przedziału, i od centrum do końca badanego przedziału
   - Wybieramy przedział, w którym znaki końców granic są sobie przeciwne.
   - Na wybranym przedziale ponownie wyznaczamy wartość funkcji w jego środku.

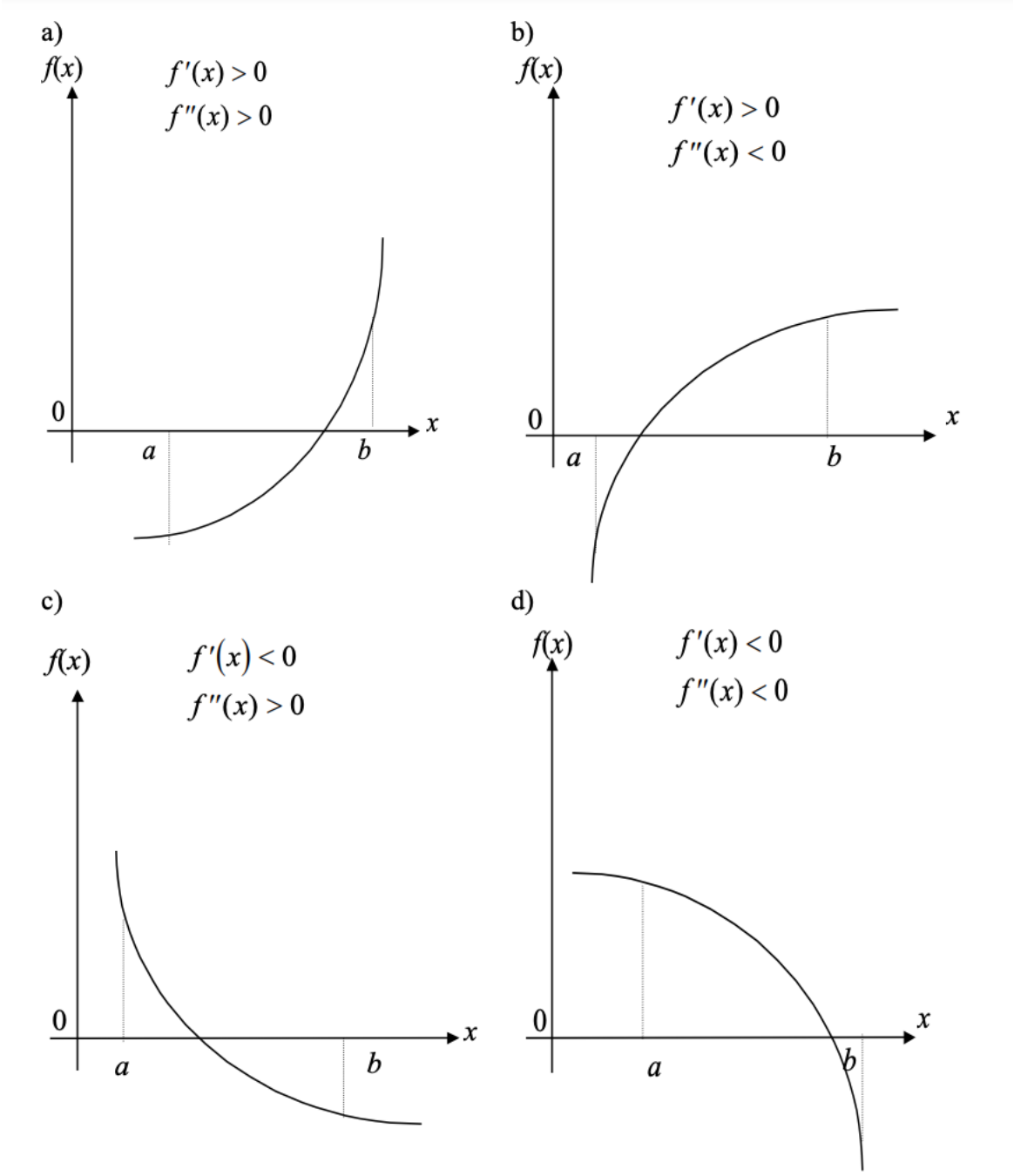
- Jeżeli otrzymana wartość jest równa 0 lub wystarczająco bliska 0, mamy pierwiastek (miejsce zerowe)
- jeżeli nie, powtarzamy znowu ten krok.

Obliczenia kończy się, gdy uzyskany przedział jest wystarczająco mały, żeby z wystarczającą pewnością założyć położenie pierwiastka funkcji.

Z zasady działania, metoda ta nazywana jest czasem metodą połowienia lub metodą równego podziału.

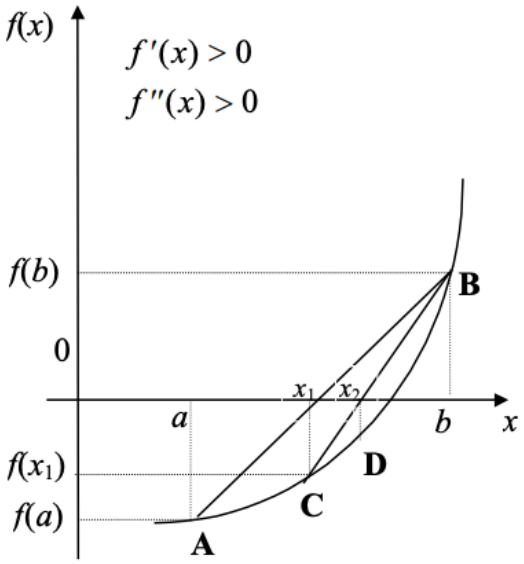
Metoda regula-falsi

Metoda regula-falsi jest nazywana też *metodą fałszywego założenia liniowości funkcji*. Przyjmuje się w niej założenia, że **funkcja ma dwie ciągłe pochodne niezerowe**, przy czym **ich znaki na przedziale <a,b> są stałe**. **Znaki granic przedziału a i b muszą być sobie przeciwne** - co oznacza, że funkcja na tym przedziale przecina oś x.



Rys. 5.1. Wykres funkcji $f(x)$ w przedziale $< a, b >$ w zależności od znaku pierwszej i drugiej pochodnej funkcji

Metoda polega na tym, że prowadzi się *cięciwę* od punktu A(a, f(a)) do punktu B(b, f(b)). Cięciwa przecina w jakimś miejscu oś x - miejsce to można określić jako x1 i jest ono przybliżeniem pierwiastka. Jeżeli ma wystarczającą dokładność - to jest to koniec rozumowania.



Rys. 5.2. Interpretacja geometryczna metody regula falsi, jeśli $f'(x) > 0$ i $f''(x) > 0$

Jeżeli jednak dokładność nie jest zadowalająca, to prowadzi się kolejną cięciwę, tym razem od punktu C(x1, f(x1)) do punktu A albo B, zależnie od tego, w którym funkcja ma znak przeciwny do f(x1). Ponownie, punkt w którym cięciwa przecina oś x jest kolejnym przybliżeniem pierwiastka i można go użyć jako miejsca zerowego, jeżeli dokładność jest zadowalająca.

Metoda siecznych

Metoda siecznych to modyfikacja metody regula falsi, która jest od niej szybsza. W przypadku metody siecznych, **nie wymaga się, żeby w punktach wyznaczających kolejną cięciwę funkcja miała przeciwne znaki**.

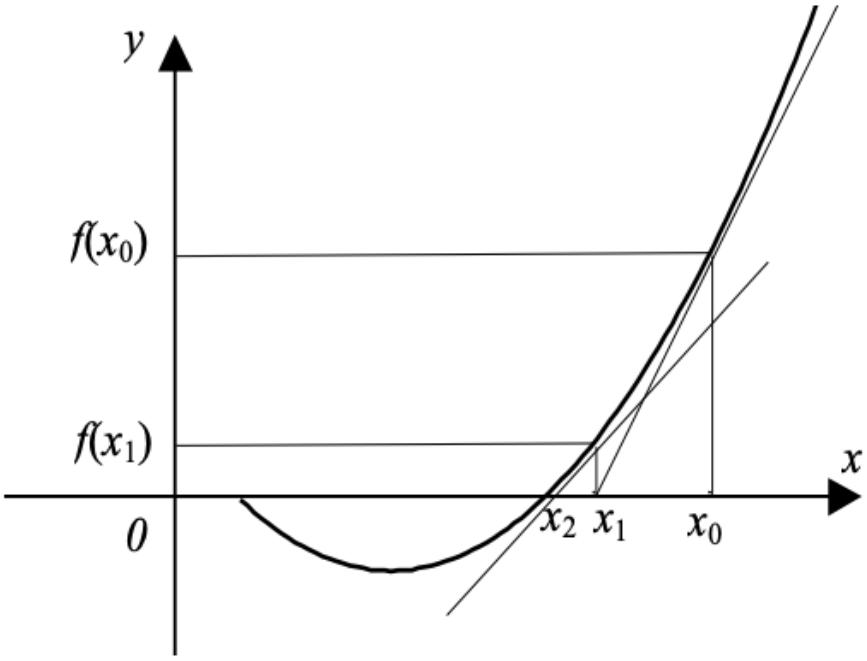
Do kolejnych przybliżeń wykorzystuje się punkty x1, x2 itd. Obliczenia w tej metodzie mogą po uzyskaniu określonego przybliżenia zacząć generować coraz większe niepewności. Moment, w którym odchyłki są najmniejsze od zera, uznaje się za końcowy moment obliczeń, a gdy potrzebna jest większa dokładność, proces iteracji zaczyna się od początku, ale na zmniejszonym przedziale.

Metoda Newtona

Metoda Newtona nazywana jest też metodą Newtona-Raphsona lub metodą stycznych. W tej metodzie wybieramy tylko jeden punkt początkowy rozwiązywania równania. W punkcie tym prowadzona jest styczna do wykresu funkcji. Przecięcie tej stycznej z osią x jest pierwszym przybliżeniem pierwiastka.

Jeżeli przybliżenie nie jest zadowalające, to w punkcie pierwszego przybliżenia oblicza się wartość funkcji i ponownie prowadzi styczną do tego punktu wykresu funkcji.

Ponownie, punkt przecięcia jest kolejnym przybliżeniem. Równania te wykonuje się ponownie aż do uzyskania zadowalającego przybliżenia.



Rys. 5.3. Interpretacja geometryczna metody Newtona-Raphsona

Metodę Newtona można rozszerzyć na układy równań nieliniowych.

Źródła

[1] Pańczyk, Łukasik, Sikora, Guziak - *Metody numeryczne w przykładach* - http://www.math.uni.wroc.pl/~ikrol/metody_num.pdf

▼ b) analityczne metody optymalizacyjne

Metody optymalizacyjne to w skrócie metody numeryczne mające na celu **znalezienie ekstremum** zadanej funkcji (nazywanej też **funkcją celu**). Metody analityczne optymalizacji opierają się na znajomości postaci analitycznej badanej funkcji (lub zestawu badanych funkcji).

Algorytmy optymalizacji funkcji jednej zmiennej

Grupa algorytmów do optymalizacji funkcji jednej zmiennej ma na celu znalezienie minimum tej funkcji, czyli min(f(x)). Algorytmy te można wykorzystać również w celu znajdowania maksimów - wtedy po prostu wyszukuje się minimum funkcji o znaku przeciwnym, czyli min(-f(x)).

Wyróżnia się metody bezpośrednich poszukiwań oraz metody oparte na gradientach. Metody bezpośrednich poszukiwań bazują po prostu na obliczanych wartościach funkcji

celu. Metody gradientowe uwzględniają natomiast również pochodne pierwszego i/lub drugiego rzędu funkcji celu.

Metody optymalizacji funkcji jednej zmiennej zakładają, że funkcja jest unimodalna - czyli ma tylko jedno minimum; w praktyce jednak funkcję dzieli się na fragmenty i dla każdego z nich szuka minimum.

Znajdowanie minimum funkcji odbywa się w dwóch krokach. W pierwszym, ustala się przedział, w którym znajduje się minimum. Dopiero w drugim znajduje się minimum z odpowiednią dokładnością.

Wśród metod ustalania przedziału, w którym znajduje się minimum funkcji można wymienić:

- **metodę wyczerpującego poszukiwania** - porównuje ona wartości funkcji celu w kilku punktach przy równych odstępach i wybiera przedział, w którym funkcja zdaje się mieć minimum
- **metodę przyspieszonego poszukiwania** - opiera się ona o jeden punkt startowy i kierunek poszukiwań, a krok sprawdzania wartości może się tu zmieniać wykładniczo (lub wolniej lub szybciej).

Natomiast wśród metod znajdowania minimum z określoną dokładnością można wymienić:

- **metody eliminowania obszarów** - np. metoda dzielenia obszaru na pół, metoda złotego podziału, metoda liczb Fibonaciego - w tych metodach w każdym kolejnym kroku iteracji pomniejsza się obszar, w którym poszukiwane jest minimum - aż do uzyskania zadowalająco małego przedziału;
- **metody estymacji punktowej** - np. metoda interpolacji kwadratowej Powella - oprócz informacji o wartości funkcji w różnych punktach, metody te wykorzystują informację o wielkości różnic wartości funkcji;
- **metody oparte na gradientach** - np. metoda Newtona-Raphsona, metoda siecznych - metody te oprócz informacji o wartości funkcji, do znajdowania minimów wykorzystują również informacje o pochodnych analizowanych funkcji celu.

Algorytmy optymalizacji funkcji wielu zmiennych

Metody te analogicznie polegają na poszukiwaniu minimów funkcji, przy czym mówimy tutaj o funkcjach wielu zmiennych - niejako wielowymiarowych. Stąd, złożoność obliczeniowa jest odpowiednio większa.

Wśród metod optymalizacji funkcji wielu zmiennych można wymienić metody bezpośrednich poszukiwań oraz metody gradientowe.

Metody bezpośrednich poszukiwań bazują na określaniu wartości funkcji w różnych punktach. Można tu wyróżnić:

- **metodę hipersześcianu**: w przestrzeni wielowymiarowej wybiera się $2^N + 1$ punktów, z czego jeden jest środkiem, pozostałe wierzchołkami hipersześcianu. Wybiera się punkt z najmniejszą wartością funkcji. Jeśli był on wcześniej wierzchołkiem, to teraz staje się centrum hipersześcianu. Jeżeli był centrum, to zmniejsza się hipersześcian i tak aż do uzyskania odpowiedniej niepewności.
- **metodę sympleksu Neldera-Meada**: zamiast hipersześcianu, na przestrzeni funkcji celu rysuje się sympleks - figurę, która ma liczbę wierzchołków o 1 większą niż liczba zmiennych funkcji celu. Czyli w przypadku funkcji dwóch zmiennych, w przestrzeni trójwymiarowej pojawia się trójkąt. Punkty takiego trójkąta nie mogą być na jednej linii (albo punkty czworokąta nie mogłyby być na jednej płaszczyźnie na funkcji 3 zmiennych). Sympleks z każdą iteracją jest zmniejszany.
- **metodę kierunków sprzężonych Powella**: najpopularniejsza z tych metod; tworzy się N liniowo niezależnych kierunków poszukiwań minimum i sekwencyjnie wykonuje się serię poszukiwań w tych kierunkach, startując za każdym razem z poprzednio znalezionego punktu.

W przypadku metod gradientowych, oblicza się gradient funkcji celu - ponieważ gradient opisuje kierunek najszybszego wzrostu wartości funkcji. Minus gradient natomiast oznacza najszybszy spadek. Wśród metod gradientowych, można wymienić:

- **metodę Cauchy’ego**: oblicza się (ujemny) gradient funkcji a następnie poszukiwania minimum wykonuje się w kierunku tego gradientu; określa się tu liczbę iteracji, bo im bliżej minimum tym gradient mniejszy i tym więcej iteracji trzeba żeby zwiększać dokładność o coraz mniejszy krok; im dalej znajduje się punkt początkowy od minimum funkcji, tym algorytm jest sprawniejszy
- **metodę Newtona**: metoda ta operuje na szeregach Taylora, ale pomijając niuanse matematyczne, jej istotną cechą jest to, że najlepiej sprawdza się gdy punkt początkowy działania algorytmu znajduje się blisko minimum
- **metodę Marquardta**: metoda ta na początku stosuje algorytm Cauchy’ego, a następnie - bliżej minimum - zaczyna stosować algorytm metody Newtona. Dlatego jest najbardziej efektywna z nich obu.

W przypadku, **gdy poszukujemy minimów funkcji o ograniczonym obszarze**, stosuje się jeszcze inne metody. Są to **metoda funkcji kar i barier** oraz **metoda Rosena rzutowanego gradientu**.

Źródła

[1] https://pl.wikipedia.org/wiki/Optymalizacja_(matematyka)

[2] https://en.wikipedia.org/wiki/Mathematical_optimization#Iterative_methods

[3] Michał Lewandowski - *Metody optymalizacji - teoria i wybrane algorytmy* https://web.sgh.waw.pl/~mlewan1/Site/MO_files/mo_skrypt_21_12.pdf

[4] Dahlquist - *Metody Numeryczne*

▼ c) metaheurystyczne algorytmy optymalizacyjne

Heurystyka to z definicji algorytm znajdowania rozwiązań, pozwalający na znalezienie “zadowalającego” rozwiązania w krótkim czasie. Heurystyki stosuje się tam, gdzie algorytm szukający dokładnego rozwiązania jest zbyt skomplikowany lub czasochłonny.

Metaheurystyka to ogólny algorytm (heurystyka) do rozwiązywania problemów. Metaheurystyka nie rozwiązuje problemu, metaheurystyka podaje sposób na utworzenie odpowiedniego algorytmu do rozwiązania problemu.

Metody metaheurystyczne są często inspirowane mechanizmami biologicznymi czy fizycznymi i można wśród nich wymienić algorytmy ewolucyjne, sztuczne sieci neuronowe, systemy rozmyte, algorytmy immunologiczne czy systemy mrówkowe.

Metody metaheurystyczne dzieli się na:

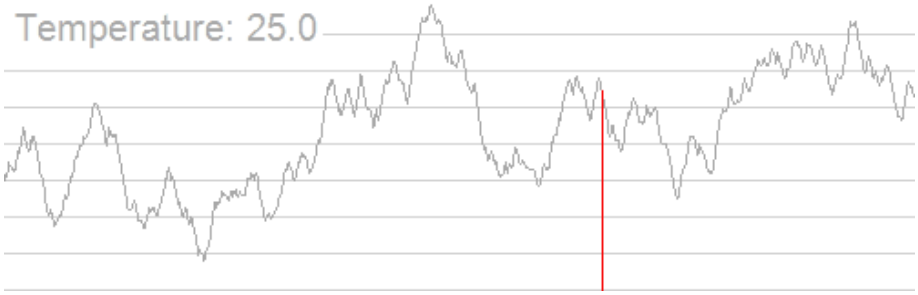
- **lokalne i globalne**: zależnie od tego, czy szukają minimów lokalnych czy minimów globalnych
- **pojedyncze rozwiązania lub rozwiązania oparte na populacji**: w tym pierwszym przypadku jest jedna ścieżka, która jest zmieniana w trakcie algorytmu, w drugim - istnieje kilka równoległych ścieżek na podstawie których wybierane są optymalne
- **Hybrydowe i memetyczne**: algorytmy hybrydowe wiążą metaheurystykę i tradycyjne metody optymalizacyjne; algorytmy memetyczne opierają się jedynie na środowisku do którego zostały stworzone
- **Równoległe metaheurystyki**: opierają się na uruchomieniu kilku równolegle działających algorytmów
- **Inspirowane naturą metaheurystyki**: wspomniane już wyżej metody bazujące na analogiach do natury.

Poniżej opisano przykładowe algorytmy metaheurystyczne:

Symulowane wyżarzanie (simulated annealing)

Proces symulowanego wyżarzania opiera się na analogii tego, w jaki sposób metal schładza się. Przyjmuje się, że na początku algorytm ma “temperaturę wirtualną”. Algorytm ten losowo wybiera punkt startowy i sprawdza wartość funkcji w tym miejscu. Następnie “temperatura” jest obniżana, co ogranicza możliwość algorytmu o przeskakiwanie z miejsca losowego do innego losowego - ogranicza zakres funkcji, w której algorytm szuka optimum.

Ta sztuczna temperatura ogranicza w zasadzie możliwość przeskakiwania algorytmu z miejsca w miejsce. Na początku, algorytm może przeskakiwać z miejsca w miejsce po całej przestrzeni funkcji. Z każdym obniżeniem temperatury, możliwość skoku się zawęża i im niższa temperatura, tym mniejszy skok może wykonać algorytm. Poniżej GIF ilustrujący ten algorytm, na przykładzie poszukiwana najwyższej wartości wartości na wykresie.



Poszukiwanie Tabu

Algorytm wybiera sobie punkt na przestrzeni funkcji i w jego sąsiedztwie poszukuje punktów bardziej optymalnych. Jeżeli znajdzie punkt najbardziej optymalny w danym sąsiedztwie, to wybiera go na lokalne optimum. Jeżeli nie ma już bardziej “optymalnych” punktów, to to potencjalne optimum zostaje oznaczone jako tabu i algorytm przestaje poszukiwać w jego miejscu nowych rozwiązań - przeskakuje do nowego sąsiedztwa i zaczyna ponowne poszukiwanie minimum.

Algorytm mrówkowy

Algorytm ten jest analogią do mrówek poszukujących pożywienia dla kolonii. Mrówki poruszają się w sposób losowy, ale gdy jedna z nich zaznaczy ścieżkę, reszta zaczyna podążać tą ścieżką. Oznaczanie odbywa się przez feromony. Feromony na dłuższych ścieżkach od jedzenia do mrowiska mają więcej czasu na odparowanie, dlatego mrówki z czasem skracają ścieżkę na taką, która możliwie najbardziej skróci drogę. Ostatecznie wszystkie mrówki idą tą samą drogą.

Tę obserwację można zastosować w algorytmach metaheurystycznych. W nich na początku wybiera się dwie “ścieżki feromonowe”, a następnie aktualizuje je. Takie sztucznie tworzone ścieżki po pewnym czasie działania algorytmu prowadzą do znajdowania optimów funkcji.

Algorytmy immunologiczne

Tego typu algorytmy naśladują systemy odpornościowe jako bardzo odporne, adaptujące się do warunków, samoorganizujące się i działające w sposób równoległy. W skład odporności wchodzą różnego rodzaju komórki i przeciwciała, pełniące różne funkcje. Ponadto wyróżnia się odporność wrodzoną i nabytą.

Inne

Ogólnie rzecz ujmując, inne algorytmy mogą opierać się na podobnego typu analogiach. Algorytmy oparte o naturę często nazywa się **algorytmami ewolucyjnymi**. Mogą naśladować - oprócz mrówek i systemów odpornościowych - również pszczoły, kukułki czy embriogenezę.

Źródła

[1] https://en.wikipedia.org/wiki/Heuristic_(computer_science)

[2] https://en.wikipedia.org/wiki/Metaheuristic

[3] https://en.wikipedia.org/wiki/Mathematical_optimization#Iterative_methods

[4] https://www.dbc.wroc.pl/Content/23599/PDF/Janiak_Advanced_algorithms.pdf

[5] http://www.imio.polsl.pl/mh-mibm.aspx

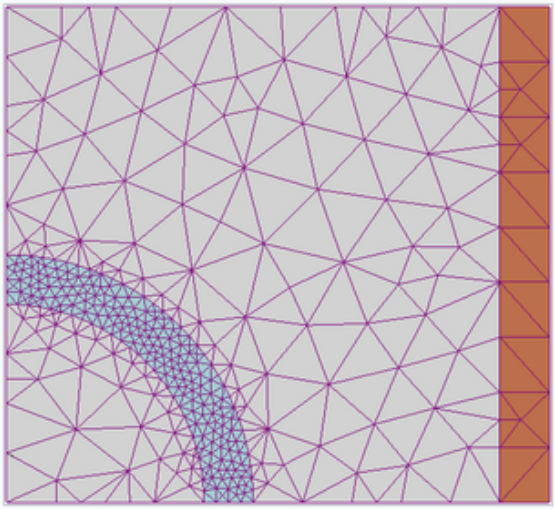
[6] https://en.wikipedia.org/wiki/List_of_metaphor-based_metaheuristics

[7] https://en.wikipedia.org/wiki/Evolutionary_algorithm#Comparison_to_biological_processes

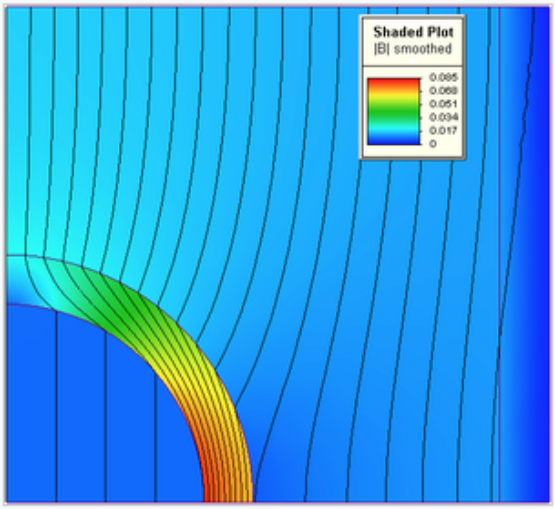
▼ d) metoda elementów skończonych

Równanie różniczkowe cząstkowe to równanie, w którym występuje niewiadoma funkcja dwóch lub więcej zmiennych oraz niektóre z jej pochodnych cząstkowych. **Metoda elementów skończonych** to metoda numeryczna służąca właśnie do rozwiązywania równań różniczkowych cząstkowych w dwóch lub trzech zmiennych przestrzennych.

W celu rozwiązania takiego równania różniczkowego, MES (FEM) dzieli duże systemy na mniejsze, prostsze części nazywane **elementami skończonymi**. Odbywa się to przez **dyskretyzację przestrzeni** w jej wymiarach - w uproszczeniu, przestrzeń funkcji dzieli się na siatkę mniejszych elementów o skończonej liczbie punktów.

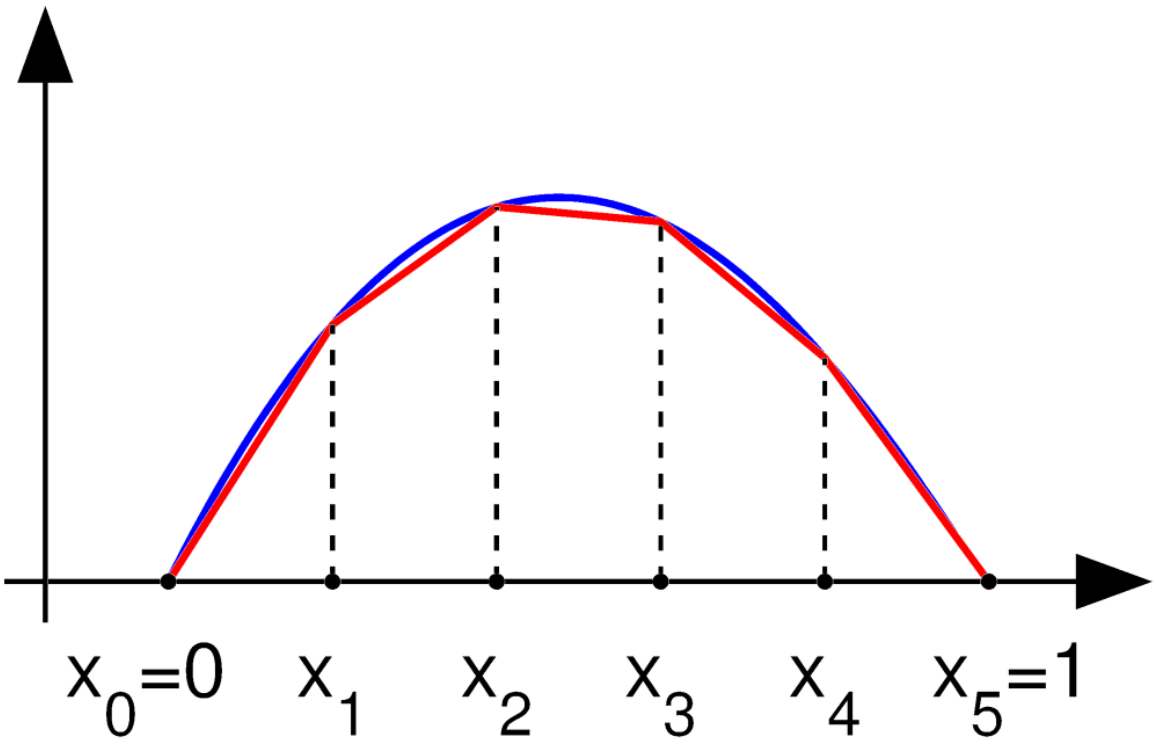


FEM mesh created by an analyst prior to finding a solution to a magnetic problem using FEM software. Colors indicate that the analyst has set material properties for each zone, in this case, a conducting wire coil in orange; a ferromagnetic component (perhaps iron) in light blue; and air in grey. Although the geometry may seem simple, it would be very challenging to calculate the magnetic field for this setup without FEM software, using equations alone.



FEM solution to the problem at left, involving a cylindrically shaped magnetic shield. The ferromagnetic cylindrical part is shielding the area inside the cylinder by diverting the magnetic field created by the coil (rectangular area on the right). The color represents the amplitude of the magnetic flux density, as indicated by the scale in the inset legend, red being high amplitude. The area inside the cylinder is the low amplitude (dark blue, with widely spaced lines of magnetic flux), which suggests that the shield is performing as it was designed to.

Taki podział przestrzeni jest bardzo istotny: zamiast dopasowywać jakiś wielomian czy inną funkcję do zadanego modelu, **model dzieli się na mniejsze elementy i rozpatruje przez pryzmat prostszych funkcji** - na przykład funkcję jednej zmiennej, zamiast opisywać wielomianem, można opisać od jednego do drugiego kawałka odcinkami prostymi i wyznaczać po prostu ich wartości. **Im mniejsze elementy, tym większa dokładność odwzorowania rzeczywistości, jednak wymaga to coraz większej mocy obliczeniowej.**



Po takim podziale i obliczeniu, pomniejsze elementy są ponownie składane w duży model początkowy. Po powrocie do tego modelu, MES aproksymuje rozwiązanie matematyczne poprzez analityczne minimalizowanie związanej z nią funkcji błędu.

Podstawowa zaleta MES to możliwość uzyskania rozwiązań dla obszarów o skomplikowanych kształtach, gdzie ścisłe obliczenia analityczne są trudne lub niewykonalne. MES jest dlatego stosowana często zwłaszcza w obliczeniach mechanicznych, termodynamicznych czy elektromagnetycznych.

Wadą MES jest to, że obliczenia wymagają dużej mocy, a na dodatek są wykonywane iteracyjnie, więc **małe błędy z czasem mogą rosnąć w duże** - co wymaga dodatkowego uodparniania algorytmu.

MES może zawierać różne błędy w swoich rozwiązaniach. Między innymi są to:

- **błąd modelowania**: model matematyczny nieoddający idealnie rzeczywistych warunków

- **błąd wartości współczynników**: założenia co do materiałów, etc. są obarczone błędem, co przekłada się na błąd współczynników równań różniczkowych w symulacji
- **błąd odwzorowania obszaru**: np. model trójwymiarowy ma wymiary nie do końca oddające wymiary obiektu rzeczywistego
- **błąd numeryczny**: np. błąd dyskretyzacji, czyli za mały krok do obliczeń lub błąd maszynowy
- **błąd zaokrągleń**: reprezentacja liczb w komputerze ma skończoną liczbę znaków, więc są one po prostu ucinane.

Inną wadą MES jest to, że symulacje nie mogą być wykonywane w czasie rzeczywistym, bo policzenie układu może zajmować bardzo dużo czasu.

Źródła

[1] https://pl.wikipedia.org/wiki/Metoda_elementów_skończonych

[2] https://en.wikipedia.org/wiki/Finite_element_method