I ty możesz zostać magistrem

25 września 2020

Spis treści

1 Baza Bernsteina

Baza Bernsteina ze względu na świetne właściwości numeryczne i geometryczne jest szeroko stosowana w systemach CAD/CAM pomimo faktu, że

nie jest trójkątna (traingular ???) i nie jest najszybsza w obliczeniach.

Ważne właściwości Bazy Bernsteina:

- Formują bazę w $n+1$ wymiarowej przestrzeni w^n wszystkich wielomianów stopnia nie większego niż n.
- Sumują się do 1 dla każdego $t \in R$
- Są nieujemne w przedziale $[0, 1]$ i dodatnie w $(0, 1)$.
- Sa symetryczne, tzn. $B^n_{i=0\ldots n}(t) = B^n_{n-1}(1-t)$

1.1 Algorytm de Casteljau

Algorytm de Casteljau służy do obliczania wartości wielomianów w Bazie Bernsteina. Jest stabilny numerycznie. Niewielkim kosztem możemy uzyskąć nie tylko wartość, ale i pochodną w punkcie. Należy odczytać obie wartości algorytmu dla $n-1$ odjąć je i pomnożyć przez n.

1.2 Twierdzenie Weierstrassa

Każdą funkcję ciągłą o wartościach rzeczywistych na przedziale domkniętym $[a, b]$ można przybliżyć jednostajnie z dowolną dokładnością wielomianami (np. wielomianami Bernsteina). Im więcej punktów kontrolnych, tym większa dokładność.

1.3 Baza Hermite'a

Jeżeli znamy wartości na krańcach przedziału i znamy wartości pochodnych w tych punktach, to podstawiamy do wzoru i mamy aproksymację funkcji na przedziale.

1.4 Baza Lagrange'a

Baza nie jest triangularna (???). Można w niej interpolować. Wielomiany Lagrange'a pozwalają nam na interpolacje punktów, bez potrzeby rozwiązywania układów współrzędnych.

1.5 Węzły Czebyszewa

Interpolacja w węzłach Czebyszewa jest prawie najlepsza (znika efekt Rungego).

2 Piecewise polynomials

2.1 Baza B-Spline

They are much more complex. There are two interesting properties that are not part of the Bézier basis functions, namely: (1) the domain is subdivided by knots, and (2) basis functions are not non-zero on the entire interval. In fact, each B-spline basis function is non-zero on a few adjacent subintervals and, as a result, B-spline basis functions are quite łocal".

3 Postacie normalne relacyjnych baz danych

1. Pierwsza NF – atomiczność danych.
2. Druga NF – kolumny zależą od całego klucza głównego (a nie np. od tylko jednego elementu).
3. Trzecia NF – żaden atrybut niekluczowy nie jest zależny od innego atrybutu niekluczowego (np. stanowisko-pensja).

23 Metody interpolacji obrotów

1. LERP
2. SLERP
3. Liniowa interpolacja kątów Eulera
4. SQUAD - interpolacja sekwencji orientacji za pomocą wzoru

24 Aproksymacja obszaru obrobionego

Obszar obrobiony możemy aproksymować za pomocą paraboloidy ściśle stycznej.

$$d(\Delta x, \Delta y) = \frac{1}{2}[\Delta x, \Delta y]\mathbf{D}[\Delta x, \Delta y]^T$$

gdzie D to dwuforma (przekształcenie dwuliniowe) określająca przybliżaną powierzchnię.
Typowe zadanie z PUSNu: jest dana powierzchnia w postaci implicite. Sprawdź jaki maksymalny promień freza nie spowoduje podcięć lub o ile musimy się przesuwać, żeby frezować z zadaną tolerancją (żeby rowki miały maksymalnie jakąś wysokość).
Robimy dwuformę powierzchni i freza i je odejmujemy. Sprawdzamy czy ta dwuforma jest dodatnio określona ($X^T DX > 0$, dla każdego X). Jeżeli tak, to nie ma podcięć. Jeżeli nie, to podcięcia mogą wystąpić, ale nie muszą (chyba trzeba sprawdzić kierunki, które chcemy frezować ???).

25 Lokalne i globalne algorytmy programowania 3C

Lokalne - tak jak frezowaliśmy nasze modele. APT to język do programowania urządzen sterowanych numerycznie (czyli do robienia ścieżek). Wyróżniamy w nim procedury TO/ON/PAST/TANTO (TANgential TO) oraz drive surface, part surface i check surface. PS - frezujemy stycznie do niej, DS - prowadzimy frez stycznie do niej, CS - na niej się zatrzymujemy.
Lokalne metody nie gwarantują ścieżek bez podcięć (collision free tool paths). Natomiast metody globalne wykluczają kolizje z ostatecznym kształtem obrabianej części. Globalne metody projektujemy poprzez wyznaczenie ITO (Inverse Tool Offset). Robimy sumę Minkowskiego przeszkód z odwórconym narzędziem w przestrzeni konfiguracji.

26 Metody rozwiązywania zadania odwrotnego prostych łańcuchów kinematycznych

Łańcuchy kinematyczne składają się ze zbioru sztywnych elementów połączonych za pomocą przegubów. Mamy dwa rodzaje przegubów: prismatic (prze-

suwające) i revolute (obracający). Możemy rozwiązywać metodami explicite (jak na zajęciach): algebraiczną i geometryczną. Metodę geometryczną możemy sobie wyobrazić, co często ułatwia implementację, lecz przy bardziej skomplikowanych układach łatwiej jest w niej o błąd niż w bardziej ogólnej metodzie algebraicznej. Oprócz tego w przemyśle są stosowane metody numeryczne, które liczą dowolny łańcuch kinematyczny. Są wolniejsze, ale łatwiejsze do zaimplementowania. Do rozwiązywania zadania odwrotnego łancuchów kinematycznych stosujemy notację Denavita-Hartenberga, która upraszcza opis łańcucha.

$$F_{i+1} = F_i F_{i(i+1)}$$

$$F_{i(i+1)} = R_{x_i}(a_i) T_{x_i}(u_i) T_{z_{i+1}}(h_{i+1}) R_{z_{i+1}}(b_{i+1})$$

27 Algorytmy szukania drogi

Flood-Fill - jak w robocie 2D na PUSNie. Mało optymalny, łatwy w implementacji.

PRM (Probabilistic Roadmap):

1. Losujemy punkty w przestrzeni konfiguracji, które nie są w przeszkodach.
2. Próbujemy połączyć każdy punkt z każdym (czyli łączymy, gdy nie ma między nimi kolizji).
3. Szukamy ścieżki.

Przydatną strukturą danych są kd-drzewa.

RRT (Rapidly-exploring random tree):

1. Sprawdzamy czy wierzchołek startowy jest dopuszczalny (nie jest w kolizji).
2. Wykonujemy k kroków polegających na wylosowaniu nowej konfiguracji q_i i łączymy ją do najbliższego wierzchołka (jeżeli nie ma kolizji).

35 Interpolacja i aproksymacja w bazach B-spline

http://www.cad.zju.edu.cn/home/zhx/GM/009/00-bsia.pdf

36 Powierzchnie obciętę i standard IGES

Standard IGES jest standardem międzynarodowym dotyczącym danych topologicznych, geometrycznych i niegeometrzycznych (np. materiały, cechy użytkowe). Na podstawie tego standardu powstał również format pliku o tej samej nazwie pozwalający na zapisanie ponad 150 różnych typów obiektów, np. powierzchni trymowanych.

Powierzchnie obcięte składają się z dwóch części: powierzchni bazowej oraz krzywych trymowania, które wyznaczają obszary trymowania.

37 Struktury danych reprezentacji B-rep

Aby uniknąć częstych obliczeń takich jak np, sprawdzenie czy punkt leży na krzywej warto zapamiętywać informacje przy tworzeniu tych struktur.

Boundary representation składa się z dwóch części: topologii oraz geometrii (powierzchnie, krzywe oraz punkty). Główne elementy topologii to: ściany (faces), krawędzie i wierzchołki. Inne elementy to powłoka (shell) - zbiór połączonych ścianek, pętla - cykl krawędzi ograniczającej ściankę, loop- edge links (znane także jako skrzydlaczki???) - służą do tworzenia cykli z krawędzi (edge circuits).

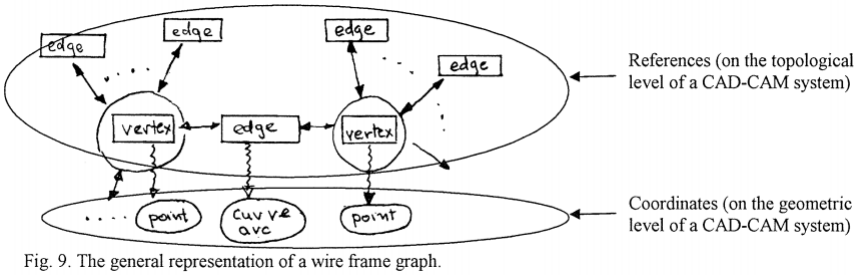


Fig. 9. The general representation of a wire frame graph.

Rysunek 1: B-rep od prof. Marciniaka

37.1 Porównanie CSG

CSG jest drzewem i zapamiętuje kolejne operacje wykonane na prymitywach. Jest budowane z prymitywów i podstawowych operacji boolowskich.

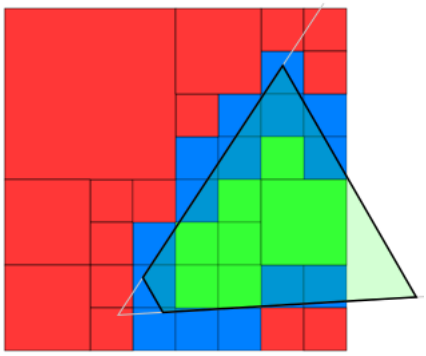
B-rep ma większą liczbę operacji, np. wyciągnięcie (extrusion) i wygładzenie krawędzi (chamfer), co pozwala na bardziej „ludzkie" tworzenie modeli.

38 Metody lokalizacji obliczeń geometrycznych

Przykładem problemu lokalizacji obliczeń geometrycznych jest wykrycie kolizji dwóch złożonych obiektów. Zamiast sprawdzać każdy ich element z każdym, chcemy uprościć obliczenia, odrzucając obszary, w których wiemy, że kolizja raczej nie zajdzie.

38.1 Drzewo BSP

Drzewo BSP (binary space partitioning) - dzielimy przestrzeń na dwie mniejsze dowolną płaszczyzną.
Drzewo kd - dzielimy przestrzeń na dwie mniejsze płaszczyzną ortogonalną do osi układu.
Octree - dzielimy przestrzeń 3D na osiem sześcianów.



Rysunek 2: Przykład quadtree

39 Struktura systemu do projektowania przez podanie ograniczeń

Zamiast sekwencyjnie rysować scenę (np. najpierw wstawiamy punkt, potem okrąg, który ma w nim środek, następnie styczną itp.), podajemy układ równań, który definiuje nam całą scenę. Jeżeli zmiennych mamy więcej niż równań to część zmiennych musi wprowadzić użytkownik jako dane wejściowe.

40 Pojęcie naprężenia w materiale. Wektor i tensor naprężenia

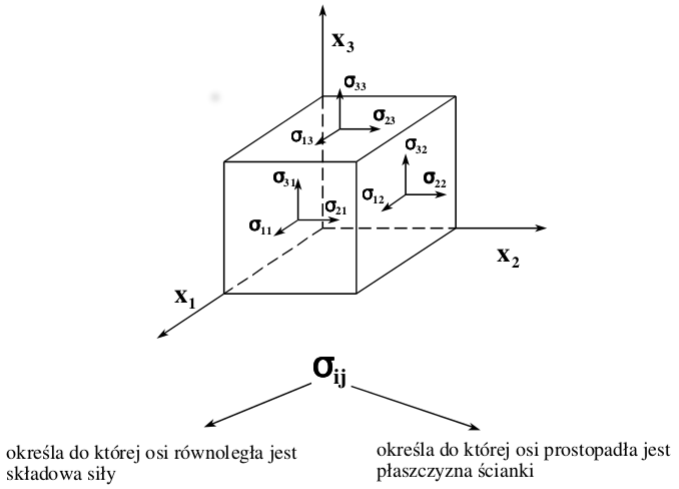
Naprężenie σ (ang. stress) to miara gęstości powierzchniowych sił wewnętrznych, występujących w pewnym punkcie przekroju ośrodka ciągłego (ciała wolumetrycznego). Jednostką naprężenia jest paskal. Reprezentowany jako trójwymiarowy symetryczny tensor drugiego rzędu.

Wektor naprężenia t to gęstość sił działających na element powierzchni w danym punkcie $P \in B$, gdzie B to ośrodek ciągły (ciało). W przypadku sił wewnętrznych wartość t w punkcie P zależy od przekroju. Wektor t jest skierowany wzdłuż wersora normalnego n do powierzchni.

Po prostu: t to wektor z kierunkiem n, a σ to jego wartość.

$$t_i = \sigma_{ij} n_i \tag{1}$$

Rozważmy najpierw tensor naprężenia, którego składowe definiują siły przypadające na jednostkową powierzchnię. W tym celu zdefiniujmy bardzo mały (jednostkowy) sześcian wewnątrz materiału.



Siły działające na ścianki jednostkowego sześcianu definiują składowe tensora naprężenia σ_{ij}. Pierwszy wskaźnik (i) definiuje kierunek, wzdłuż którego działa siła, zaś drugi (j) – oś do której jest prostopadła płaszczyzna ścianki, w której działa siła.

Rysunek 3:

41 Grafika Komputerowa II

41.1 Źródła światła

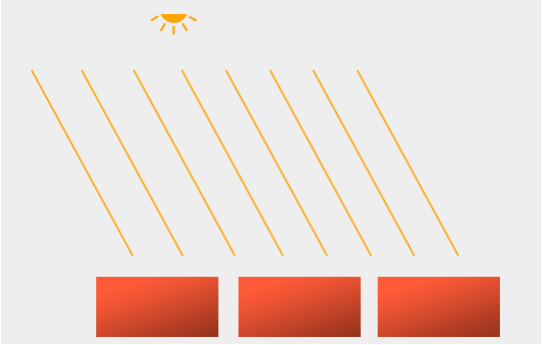
Każde źródla światła posiada parametry: ambient, diffuse i specular

41.1.1 Światło kierunkowe

Światło kierunkowe (Directional light)

1. Reprezentowany tylko poprzez kierunek (vec3 direction)
2. Znajduje się w nieskończoności (pozycja nie ma znaczenia)
3. Wszystkie promienie światła mają taki sam kąt padania

4. Przykład: słońce

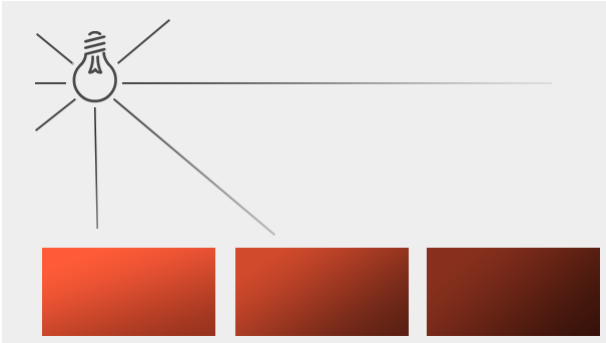


Rysunek 4: Światło kierunkowe

41.1.2 Światło punktowe

Światło punktowe (Point light)

1. Reprezentowany tylko poprzez pozycje pozycje (vec3 position)
2. Promienie światła padają we wszystkich kierunkach
3. Tłumienie (Attenuation) światła redukuje intensywność swiatła wraz z odległościa promienia światła
4. Przykład: żarówka

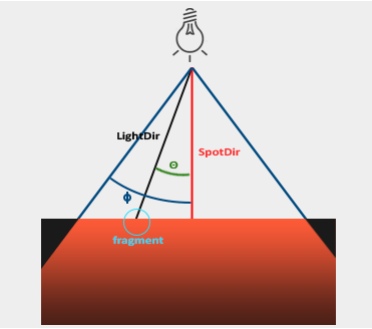


Rysunek 5: Światło punktowe

41.1.3 Reflektor

Reflektor (Spotlight)

1. Reprezentowany poprzez pozycje, kierunek i cutoff (vec3 position, vec3 direction, float cutOff)
2. Promienie padaja w danych kierunku.
3. cutOff definiuje nam promień reflektora
4. Przykład: latarka



Rysunek 6: Reflektor

41.2 Oświetlenie lokalne

Hasła związane z oświetleniem lokalnym:

1. Obliczanie intensywności (koloru) światła odbitego od punktu powierzchni
2. Źródła światła
   (a) kolor
   (b) geometria
3. Właściwości powierzchni
   (a) kolor
   (b) materiał (metal, dielektryk)

(c) geometra

4. Obserwator

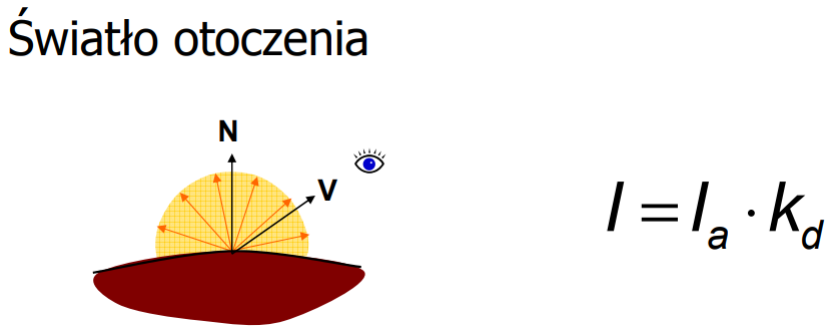
Lokalne modele oświetlenia możemy podzielić na 3 typy:

1. Empiryczne
2. O podłożu fizycznym
3. Hybrydowe

41.2.1 Empiryczne

Światło otoczenia (Ambient)

1. Oświetlenie tła
2. Światło oświetlające obiekt nie pochodzi z konkretnego źródła
3. Intensywność światła jednakowa we wszystkich kierunkach
4. Każdy element oświetlony w jednakowy sposób
5. Światło odbite rozchodzi się jednakowo we wszystkich kierunkach

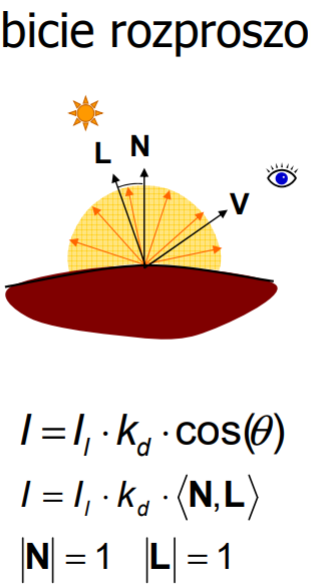


Rysunek 7: Światło otoczenia (Ambient)

Światło rozproszone (Diffuse).

1. Model Lamberta (odbicie lambertowskie)

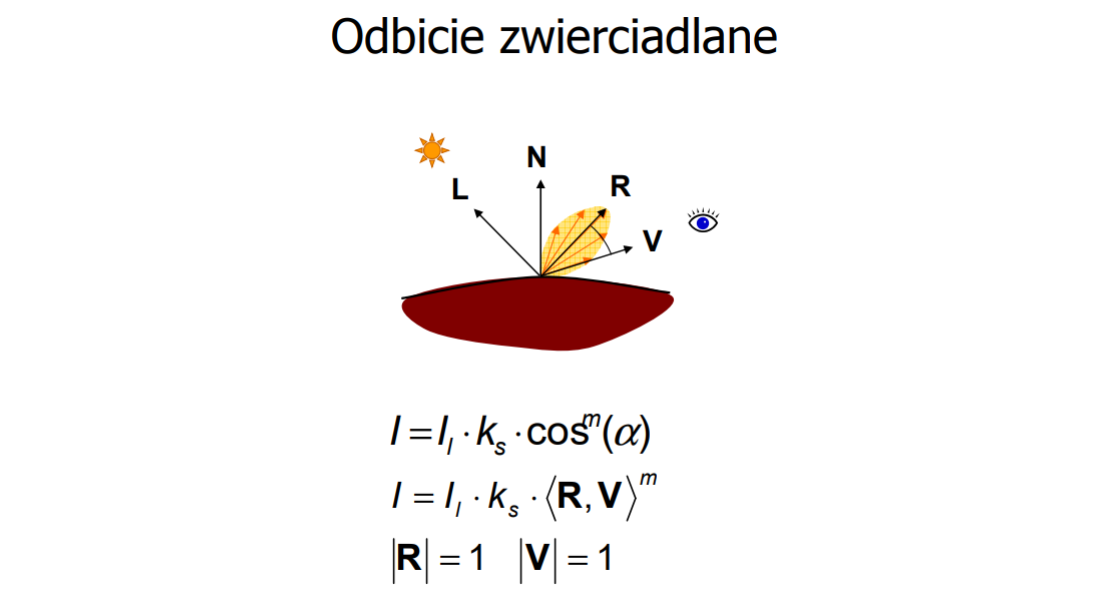
2. Obiekt posiada matową powierzchnię
3. Światło odbite rozchodzi się jednakowo we wszystkich kierunkach
4. Intensywność światła odbitego zależy od kąta padania światła



Rysunek 8: Światło rozproszone (Diffuse)

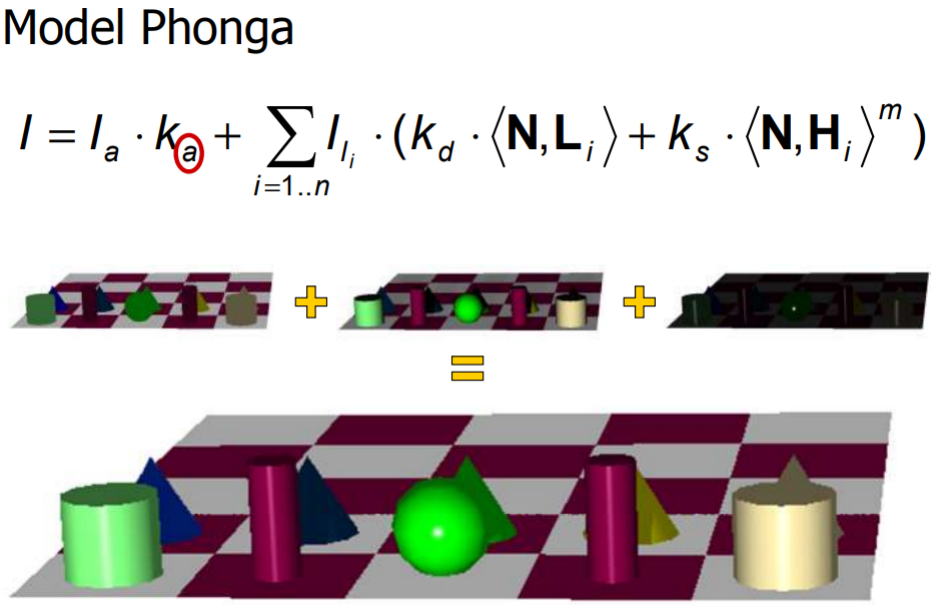
Odbicie Zwierciadlane (Specular).

1. Obiekt posiada gładką powierzchnię
2. Prawo Snella: kąt odbicia jest równy kątowi padania
3. Część energii rozprasza się w innych kierunkach
4. Intensywność światła odbitego zależy od kąta padania światła i od kąta padania oka (obserwatora)



Rysunek 9: Odbicie Zwierciadlane (Specular)

Model Phonga polega na połączeniu trzech komponentów: ambient, diffuse i specular.



Rysunek 10: Phong

Oświetlenie anizotropowe – oświetlenie obiektu jest zależne od właściwości fizycznych jego materiału. Czyli w zależności jak na niego patrzymy, oświetlenie się różni. Przykład: chropowaty metal (jak czajnik), włosy.

41.2.2 O podłożu fizycznym

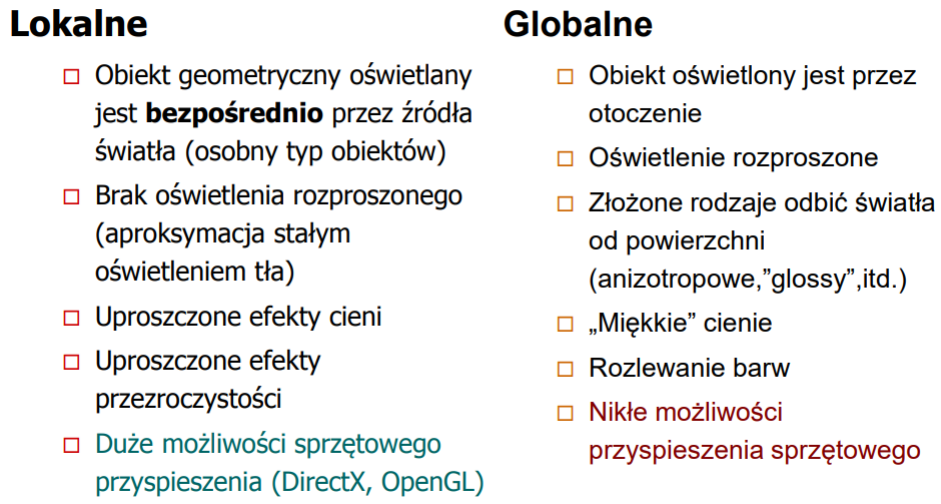
W modelach o podłożu fizycznym podstawową modyfikacją w stosunku do modeli empirycznych jest to, że powierzchnie traktujemy jako mikrolustra. Ponadto wprowadzamy funkcję BRDF, która zapewnia nam, że energia światła nie wzrośnie po odbiciu. Określa ona prawdopodobieństwo, że nadchodzący foton zostanie odbity w danym kierunku. Rozkład mikroluster może być uzyskany różnymi metodami jak np. rozkład Gaussa. Współczynnik Fresnela – ilość światła odbitego do padającego.
Przykładowe modele fizyczne (różnią się rozkładem mikroluster):

- Torrence'a-Sparrowa
- Cooka-Torrence'a

41.2.3 Hybrydowe

41.3 Oświetlenie globalne

41.4 Porównanie



Rysunek 11:

41.4.1 Radiosity

Dyskretyzujemy elementy sceny na płaty (kd-drzewem??). Radiosity wyznacza globalny rozkład natężenia światła uwzględniając pochłonięcia i odbicia światła jakie mają miejsce na wszystkich powierzchniach znajdujących się na scenie. Czyli modeluje prawie dokładnie to samo, co obserwuje się w rzeczywistym świecie, gdzie każda powierzchnia pochłania światło, ale także część odbija. Radiosity uwzględnia wyłącznie odbicia rozproszone, tj. intensywność światła odbitego jest niezależna od kierunku - dzięki temu uzyskane wyniki są niezależne od położenia obserwatora, co pozwala na wielokrotną, dowolną wizualizację sceny bez ponawiania obliczeń.

Metoda nie uwzględnia jednak efektów świetlnych zależnych od położenia obserwatora takich jak rozbłyski na powierzchniach metalicznych, odbicia zwierciadlane, załamanie światła itp.

41.4.2 Path tracing

Algorytm przebiega następująco: z punktu w którym znajduje się obserwator wypuszczane są promienie (wiele promieni przez każdy pixel). Jeśli promień trafi w jakiś obiekt, z punktu przecięcia (rekursywnie) wypuszczane są kolejne promienie (co najmniej jeden), przy czym kierunek nowych promieni jest losowy; od jakości funkcji losującej zależy jakość obrazu. Każdy obiekt może pochłaniać lub emitować światło.

Tworzenie pojedynczej ścieżki kończy się, gdy głębokość rekursji przekroczy pewien limit. Wówczas wyznacza się ostateczne natężenie światła, jakie dociera do obserwatora: składa się na nie natężenie pochodzące od obiektów emitujących, które następnie na ścieżce jest tłumione; tłumienie zależy od współczynnika pochłaniania światła dla każdego trafionego obiektu (funkcja BRDF, uwzględniające m.in. kąt pomiędzy promieniem padającym i odbitym).
Powstaje **zaszumiony** obraz! (filtr macierzowy medianowy chyba naprawi)

41.4.3 Photon mapping

Algorytm dzieli się na dwie fazy: sianie i zbieranie fotonów.
Sianie:

- Wystrzeliwujemy dużo fotonów z każdego źródła światła i pozwalamy im poodbijać się po scenie.
- Przy każdym odbiciu zostawiamy informację o niesionej energii i kierunku przybycia.

Zbieranie:

- Wystrzeliwujemy promienie od oka w stronę sceny.
- Na podstawie pierwszego przebiegu (siania) wyznaczamy informacje o oświetleniu miejsca kolizji.
- Odbijamy promień w odpowiednim kierunku.

41.5 Teksturowanie

41.5.1 Normal map

Zapisujemy wektory normalnych do tekstury. W czasie renderowania zamiast wyliczać normalne, pobieramy je z tekstury. W ten sposób możemy uzyskać efekt nierówności obiektu. Bump map to stara wersja mapy normalnych, która zamiast wektora trzyma skalar. Gdy chcemy uzyskać efekt głębi to przyciemniamy pixel (na podstawie grayscale z bump map), a gdy wypukłość to rozjaśniamy.

41.5.2 Displacement map

Zapisujemy wektory przesunieć do tekstury. W czasie renderowania zmieniamy pozycje wierzchołków o wartość z mapy. Height map to uproszczona wersja mapy przesunięć, która zamiast wektora trzyma skalar (więc zmienia tylko wysokość). Height mapy stosujemy np. do generowania górzystego terenu. Jednak nie da się zrobić podcięć (jaskiń). Displacement map stosujemy do dodania szczegółowości geometrii.

41.6 Widoczność

Metody w screen space:

1. Algorytm malarski – sortujemy obiekty po z-cie. Rysujemy od najdalszego do najbliższego. Nie działa dla cyklicznie nakładających się obiektów i przy penetracji.

2. Scan linia – idziemy scanlinią wierszami po ekranie. W punktach przecięć obiektów sprawdzamy, który jest na górze i kolorujemy w jego kolorze aż do następnego przecięcia. Algorytm scanlinii: sortujemy krawędzie, mamy listę krawędzi aktywnych.

3. Z-buffer – przechodzimy po obiektach po kolei. Dla każdego pixela sprawdzamy jego z i porównujemy z aktualną wartością z-buffera. Jeżeli obecny pixel ma mniejsze z, to pixel przyjmuje jego kolor. Algorytm powszechnie stosowany.

Drzewa:

1. BSP (binary space partitioning) – dzielimy przestrzeń na dwie części. Nie muszą być równe ani prostopadłe do osi.
2. KD-drzewo – jak BSP, ale dzielimy ortogonalnie.
3. Octree – dzielimy na 8 takich samych sześcianów.

Zastosowania kd-drzew: określanie widoczności, wypełnianie wielokątów, optymalizacje systemów cząsteczkowych (w niektórych przypadkach możemy łączyć grupy cząste w jedną, np. gdzy mamy dwie zwarte grupy położone od siebie daleko, to możemy je potraktować jak dwie większe cząstki).

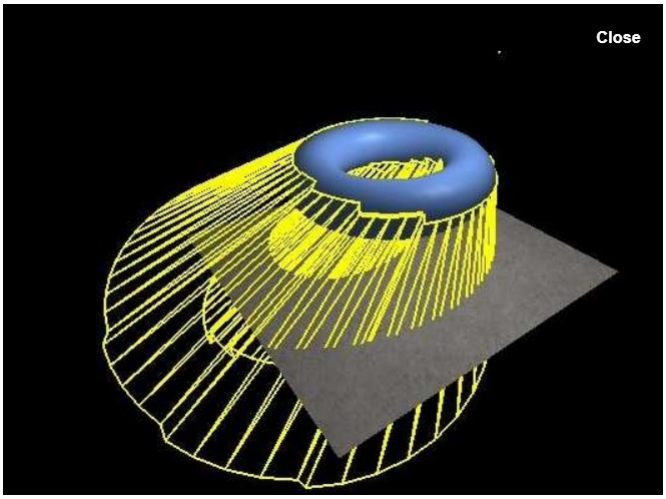
41.7 Cienie

41.7.1 Shadow mapping

Jest to metoda dwuetapowa (dwa passy) i jest domyślnie stworzona dla direction light, ale są możliwe modyfikacje dla point light (cubemap). Renderujemy scenę z perspektywy źródła światła i zapisujemy z-buffer do tekstury (shadow mapy). Przechodzimy do drugiego etapu. Patrzymy z kamery. Każdy pixel, który widzimy, transformujemy do układu współrzędnych światła. Następnie sprawdzamy czy jego z jest większy od z zapamiętanego w shadow mapie. Jeżeli tak to ten pixel jest w cieniu (czyli czarny + ambient). Jakość cienia jest zależna od rozdzielczości mapy, każde źródło światła wymaga swojej shadow mapy.

41.7.2 Shadow volumes

Bierzemy wektor kierunku od światła do bryły. Następnie w tą stronę wydłużamy kontur bryły w nieskończoność (w praktyce do far-plane). Patrzymy z kamery czy pixel jest w bryle cienia. Aby to zrobić liczymy liczbę przecięć z bryłą cienia. Ilość przecięć jest zapisywana w stencil bufforze. Przy każdym przecięciu ściany bryły cienia inkrementujemy/dekrementujemy zmienną w stencil bufforze. Jeżeli ostatecznie wychodzi nam 0, to pixel nie jest zacieniony. Mamy dwie metody: z-fail i z-pass. Z-fail jest lepsza, bo działa gdy kamera jest w cieniu. Czas obliczeń jest zależny od geometrii (jeśli mamy dużo trójkątów, to wyznaczenie brył cienia jest kosztowne).

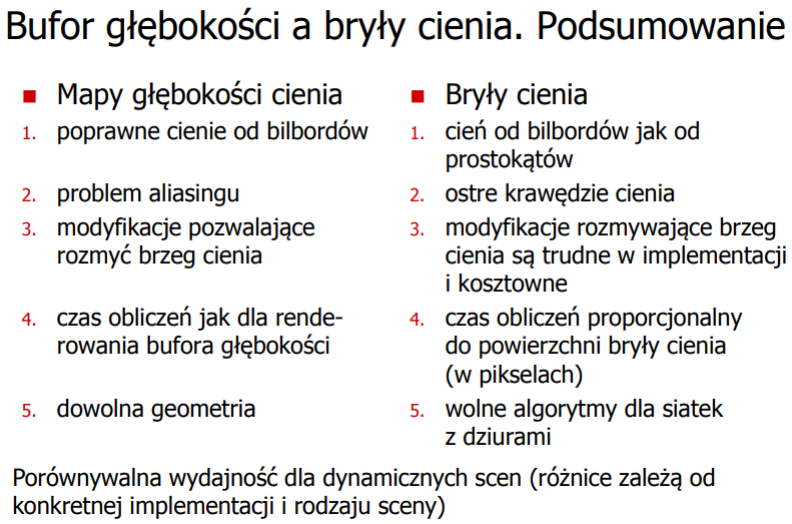


Rysunek 12: Shadow volumes

41.7.3 Cienie miękkie

1. PCF – Wyznaczamy mapę cienia i na jej podstawie sprawdzamy czy dany piksel jest lub nie jest w cieniu (czyli ma wartość 0 lub 1). Następnie sprawdzamy jego sąsiadów (np. macierz 3x3) i uśredniamy wynik. Przykład: wyszło nam, że nasz pixel jest w cieniu (czyli ma wartość 0). Ale po uśrednieniu wyników jego sąsiadów wyszło nam 0,55. Więc zamiast brać 0 jako jego intensywność, bierzemy 0,55 (i mamy szarość).

2. VSM (variance shadow maps) – zapamiętujemy z oraz z^2. Następnie na podstawie dwóch pierwszych momentów rozkładu prawdopodobieństwa określamy czy pixel jest w cieniu.



Rysunek 13:

41.7.4 Antyaliasing

Tekstur:

1. Mip-mapy – wraz z odległością zmniejszamy poziom szczegółowości tekstury (musimy mieć kilka tekstur).

2. Rip-mapy – jak mip mapy, ale mamy różne poziomy szczegółowości w zależności od kierunku (np. pas startowy).

Sceny:

1. Super-sampling (SSAA) – renderujemy na x-razy większą rozdzielczość i uśredniamy.

2. Multi-sampling (MSAA) – dzielimy poksel na x kawałków i w zależności ile nam przetnie linia to taki kolor bierzemy (jak przetnie 1 pixel z 4 to 25% koloru).

3. FXAA – wyszukujemy patternów i odpowiednio przetwarzamy.

4. TXAA – mieszamy kilka klatek.

42 Algebra Zewnętrzna

42.1 Forma różniczkowa (k-forma)

Forma różniczkowa albo k-forma to tensor rzędu k, który jest antysymetryczny względem wymiany par indeksów. 1-form ω_1 o wymiarze n:

$$\omega_1 = a_1 dx_1 + a_2 dx_2 + ... + a_n dx_n = a_i dx_i \tag{2}$$

gdzie a_i są funkcjami $a_i(x_1, x_2, ..., x_n)$

Przykład 2-formy:

$$\omega_2 = a_1 dx_1 \wedge dx_2 + a_2 dx_1 \wedge dx_3 + a_3 dx_2 \wedge dx_3 \tag{3}$$

gdzie operator $(\cdot) \wedge (\cdot)$ to iloczyn zewnętrzny zdefiniowany:

1. $t \wedge s = -s \wedge t$
2. $t \wedge t = 0$
3. $t \wedge (r + s) = t \wedge r + t \wedge s$

42.2 Pochodna zewnętrzna

Pochodna zewnętrzna k-formy ω to $k+1$-forma $d\omega$. Jeśli 1-forma ω to:

$$\omega = adx + bdy \tag{4}$$

wtedy $d\omega$ jest zdefiniowane jako:

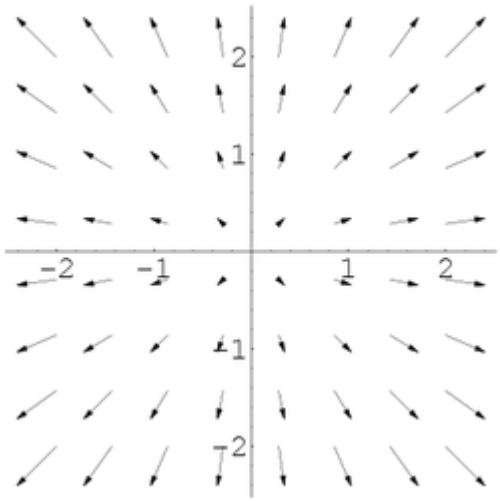
$$d\omega = d(adx + bdy) = [\frac{\partial a}{\partial x}(dx \wedge dx) + \frac{\partial a}{\partial y}(dy \wedge dx)] + [\frac{\partial b}{\partial x}(dx \wedge dy) + \frac{\partial b}{\partial y}(dy \wedge dy)] = \frac{\partial a}{\partial y}(dy \wedge dx) + \frac{\partial b}{\partial x}(dx \wedge dy)$$

42.3 Dywergencja

Dywergencja pola wektorowego to operator różniczkowy przyporządkowujący trójwymiarowemu (dwuwymiarowy też) polu wektorowemu pole skalarne będące formalnym iloczynem skalarnym operatora nabla ∇ z polem.

Dywergencja to miara ilości strumienia wchodzącego lub wychodzącego z punktu. Dywergencja to tempo ekspansji (ang. expansion, positive divergence) lub skurczania (ang. contraction, negative divergence) się strumienia.

Jeśli dany punkt zobaczy strumień, który do niego wchodzi to zacznie krzyczeć, że wszystko się zbliża (ang. negative divergence). Jeśli dany punkt zobaczy strumień, który z niego wychodzi to zacznie krzyczeć, że wszystko sie od niego oddala (ang. positive divergence).



Rysunek 14: Przykład pola wektorowego pokazujący prędkość strumienia cieczy. Strumień oddala się od początku ukladu współrzędnych (ekspansja). Ta ekspansja (ang. expansion) jest udowodniona przez dodatnią wartość dywergencji div F

42.4 Twierdzenie Stokesa

Całka formy różniczkowej (k-formy) ω na brzegu $\partial\Omega$ jakieś rozmaitości (manifold) jest równa całce pochodnej zewnętrznej $d\omega$ na całości Ω

$$\int_{\partial\Omega} \omega = \int_{\Omega} d\omega \tag{5}$$

Twierdzenie Gaussa-Ostrogradskiego, podstawowe twierdzenie rachunku całkowego (Fundamental theorem of calculus) i twierdzenie Greena są specjalnymi przypadkami twierdzenia Stokesa.

Tw. Stokesa umożliwia nam zamianę całki powierzchniowej na objętościową (i na odwrót)

42.5 Twierdzenie Gaussa-Ostrogradskiego

Znane także jako twierdzenie o dywergencji (Divergence theorem). Specjalny przypadek tw. Stokesa, w którym funkcja podcałkowa po objętości to dywergencja pola wektorowego F

$$\int_V (\nabla \cdot F) dV = \int_{\partial V} (F \cdot n) dA \tag{6}$$

F to pole wektorowe zdefiniowane w sąsiedztwie V. n to pole normalnych (skierowanych na zewnątrz, ang. outward) do brzegu ∂V.

Załóżmy, że chcemy napompować koło samochodowe, które jest idealną bryła sztywną (koło nie rozszerzy się po dodaniu powietrza). Co się stanie z powietrzem w środku koła? Powietrzne w środku koła skurczy się.

Jeśli F to pole wektorowe reprezentujące strumień cieczy to dywergencja div F reprezentuje ekspansję lub skurczanie sie tej cieczy. Tw. o dywergencji mówi, że całkowita ekspansja strumienia cieczy w jakims trójwymiarowym regionie V jest równa całkowitemu strumieniowi cieczy na zewnątrz brzegu V.

W naszym przykładzie koło stanowi region V. Pompowanie powietrza do środka koła w kierunki przeciwnym do normalnej n daje nam ujemna wartość $\int_{\partial V}(F \cdot n)dA$. Co oznacza, że wartość $\int_V (\nabla \cdot F)dV$ też jest ujemna, co oznacza, że powietrze skurczyło się. Dokładnie to co założyliśmy.

43 Projektowanie środowiska wirtuanego

43.1 Porównanie mechanik

Wszystkie mechaniki są sobie równoważne. Różnią się łatwością zapisu równań dla różnych przypadków.

Mechanika Newtona $F = ma$. Jest dobra do systemów cząsteczkowych. Równania zależą od układu współrzędnych, w szczególności czy jest inercyjny (Wojtek patrzy na karuzelę) czy nieinercyjny (Wojtek siedzi na karuzeli). Ciężko w nim uwzględnić ograniczenia. W przypadku bryły sztywnej oprócz ruchu postępowego mamy też ruch obrotowy, który jest wyrażony wzorem

$M = I\omega$.

Mechanika d'Alamberta ułatwia wprowadzenie ograniczeń, ale przez to wzrasta nam liczba zmiennych i równań w układzie. Gdy mamy M parametrów i U ograniczeń, to mamy $M + U$ równań. W tej mechanice mamy równania DAE (różniczkowo-algebraiczne): U algebraicznych i M różniczkowych.

$$\begin{bmatrix} m_1 & & & \\ & m_1 & & \\ & & m_2 & \\ & & & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x_1} \\ \ddot{y_1} \\ \ddot{x_2} \\ \ddot{y_2} \end{bmatrix} = F + \lambda_i G_i \tag{7}$$

gdzie:
$i = 1 : U$
G_i - gradient i-tego ograniczenia

Zasada prac wirtualnych redukuje liczbę równań w mechanice d'Alamberta. Aby ją zastosować, należy wziąć wektor v z przestrzeni stycznej do wszystkich ograniczeń. Siły reakcji w d'Alambercie są prostopadłe do ograniczeń, więc taki magiczny wektor v nam je wszystkie wyzeruje (robimy dot product stronami). Dzięki temu upraszczamy układ, a liczba równań spada do $M-U$.

Mechanika Lagrange'a jest niezależna od układu współrzędnych. Dzięki temu możemy łatwo rozwiązywać skomplikowane układy ruchu - musimy tylko opisać parametryzację. Mechanika Lagrange'a bazuje na energii układu.

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{Q_i}}) - \frac{\partial L}{\partial Q_i} = 0$$

gdzie:
$L = T - U$ – funkcja Lagrange'a
T – energia kinetyczna
U – energia potencjalna
Q_i – współrzędne parametryzacji

Mechanika Hamiltona podobnie jak w mechanice Lagrange'a potrzebujemy energii i parametryzacji. Jednak zamiast jednego równania drugiego rzedu mamy dwa równania pierwszego. Mechanika Hamiltona jest połącze-

niem mechaniki klasycznej z mechaniką kwantową.

$$\dot{p_i} = -\frac{\partial H}{\partial q_i} + D$$

$$\dot{q_i} = -\frac{\partial H}{\partial p_i}$$

gdzie:
$H = T + U$ – funkcja Hamiltona T – energia kinetyczna
U – energia potencjalna
p_i – współrzędne pędu q_i – współrzędne parametryzacji

43.2 Układy dynamiczne

Układ dynamiczny gładkie pole wektorowe na N-wymiarowej rozmaitości różniczkowej zwanej przestrzenią stanów układu. Każdy stan należący do tej przestrzenii może zmieniać się w czasie zgodnie z funkcją zmiany stanu (równaniem różniczkowym).

$$\dot{(x)} = v(x, t)$$

System dynamiczny nie zależy od czasu, to wtedy jest autonomiczny.
Jeżeli funkcja zmiany stanu jest liniowa i system jest autonomiczny to jest to system LTI (Linear Time Invariant).

43.2.1 Stabilność

Na przykładzie wahadła: mamy dwa punkty równowagi - na górze i na dole. Górny nie jest stabilny, bo po wytrąceniu wahadła z tego położenia nigdy do niego nie wrócimy. Punkt na dole jest stabilny, bo wahadło do niego wraca. Stabilność z definicji oznacza, że po wytrąceniu wahadło wróci w otoczenie punktu równowagi. Jednak nasz punkt na dole wahadła jest „bardziej" stabilny, ponieważ jest stabilny asymptotycznie. Znaczy to, że wraz z upływem czasu zawsze zbliża się on do punktu równowagi (granica dąży do 0). W praktyce wiemy, że takie wahadło wróci dokładnie do punktu równowagi, a nie zatrzyma się gdzieś obok. Do sprawdzania czy punkt jest stabilny może nam posłużyć kryterium Lyapunova.

43.3 Sterowanie

System sterowania jest to system dynamiczny z dodatkowym zbiorem parametrów u, za pomocą których możemy zmieniać pole wektorowe (trajektorię) systemu.

Bang-Bang - głupie sterowanie, włączamy i wyłączamy na zmianę.

Open loop control - system kontroli, w którym u jest obliczane tylko na podstawie modelu systemu. Oznacza to, że kontroler nie obserwuje wyjścia, co w praktyce prowadzi do tego, że system nie poprawia błędów na wyjściu. Nie poprawia też zaburzeń. Jest przydatny gdy dokładnie znamy pole wektorowe i pożądaną trajektorię. Przykładowo się go stosuje w silniku krokowym sterującym ramieniem robota.

Closed loop control - jest to open loop control z dodanym regulatorem, czyli do wyznaczenia u bierzemy także wyjście z poprzedniej iteracji i wyjście idealne, do którego dążymy (wcześniej zadane). W ten sposób staramy się kompensować błędy na wyjściu. Przykład: autopilot w samolocie. Mamy zadaną trajektorię, po której chcemy lecieć. Co krok porównujemy wyjście z tą trajektorią i jeżeli są odstępstwa to poprawiamy.

Open loop control służy do minimalizacji jakiegoś kryterium (np. odległości trafienia rakiety od samolotu) i do przejścia ze stanu początkowego w dany stan końcowy. Closed loop control pilnuje nam, żeby system nie przestał być stabilny (dzięki poprawianiu błędów).

Stany równowagi - przekształcamy układ tak, żeby mieć tylko równania pierwszego rzędu (możemy wstawiać nowe zmienne, np. $v = \dot{x}$). I dla wszystkich $\dot{x} = 0$ mamy stany równowagi.

Bifurkacja - zmiana właściwości układu dla pewnych wartości. Przykład dla paraboli: ma dwa miejsca zerowe. Jednak jeśli współczynnik przy x^2 wynosi 0, to funkcja jest liniowa i ma tylko jeden pierwiatek.

43.4 Linearyzacja

Linearyzacja umożliwia nam analizę systemu nieliniowego w otoczeniu jakiegoś punktu. Linearyzacja funkcji pierwszego rzędu to szereg Taylora wokół punktu. Gdy linearyzujemy wokół punktu równowagi, macierze są stałe. W innych punktach mogą one zależeć od parametrów.

43.5 Transformata Laplace'a

Transformata Laplace'a pozwala nam przejść z równań różniczkowych określonyw w dziedzinie rzeczywistej, do równań algebraicznych w zespolonych. W ten sposób możemy obliczyć pierwiastki, a następnie za pomocą odwróconej tranformaty wracamy do dziedziny rzeczywistej. Odwrotna transformata jest słaba numerycznie, więc oblicza się ją symbolicznie lub sprawdza wartości w tabelach. Transformatę Laplace'a stosujemy tylko do liniowych równań różniczkowych ze stałymi współczynnikami (czyli LTI w rozumieniu z ćwiczeń).

43.6 Stabilność układu LTI

Możemy wyznaczyć pierwiastki wielomianu charakterystycznego. Jeżeli ich części rzeczywiste są mniejsze od zera to układ jest asymptotycznie stabilny. Możemy także to sprawdzić za pomocą Kryterium Routha lub Kryterium Hurvitza. Kryterium Routha - liczymy wielomian charakterystyczny. Jeżeli wszystkie współczynniki wielomianu są większe od zera i wszystkie elementy z pierwszej kolumny macierzy trójkątnej Routha też są większe od zera to system jest asymptotycznie stabilny. Kryterium Hurvitza: bierzemy wielomian charakterystyczny i jego współczynniki wkładamy do macierzy. Układ jest stabilny, gdy $a_n > 0$ i główne minory macierzy są dodatnie.

43.7 Kontrolowalność

System musi być kontrolowalny, żebyśmy mogli zrobić z systemem co chcemy za pomocą wejścia.
System jest kontrolowalny, gdy dowolny stan końcowy jest osiągalny z dowolnego stanu początkowego w skończonym czasie. Kontrolowalność systemu LTI sprawdzamy za pomocą własności:

$$Rank[A|B] = n$$

gdzie

$$[A|B] = [B, AB, A^2B, ..., A^{n-1}B]$$

43.8 Obserwowalność

System jest obserwowalny, gdy stan początkowy $x(0)$ może być jednoznacznie wyznaczony na podstawie wiedzy o wejściu $u(t)$ i wyjściu $y(t)$ dla wszystkich t pomiędzy 0 i dowolnym skończonym T. Obserwowalność systemu LTI sprawdzamy za pomocą własności:

$$Rank[A^T|C^T] = n$$

gdzie

$$[A^T|C^T] = [C^T, A^TC^T, (A^T)^2C^T, ..., (A^T)^{n-1}C^T]$$

Jeżeli system jest kontrolowalny i obserwowalny to **istnieje kontroler**, który stabilizuje system wzdłuż pożądanej trajektorii.

43.9 Serwomechanizm

Serwomechanizm jest jak closed loop control tylko, że nie tylko stabilizuje system i kompensuje odchylenia, ale także liczy pożądaną kontrolę. Jest szeroko wykorzystywany w praktyce, np. w sterowaniu łodzią.

43.10 PID

Kontroler PID (proportional-integral-derivative) - kontroler zawierający trzy człony poprawigające.

1. Proporcjonalny jest po to, żeby dawać wejście proporcjonalne do błędu (w Bang-Bang zawsze dawaliśmy max lub min, co było bezsensu; P nam to poprawia). Czyli kompensuje błąd bieżący.

2. Itegral (całkowy) Kompensuje akumulację błędów z przeszłości.

3. Derivative (różniczkowy) kompensuje przewidywane błędy w przyszłości.

PID jest najczęściej stosowanym kontrolerem.

43.11 LQR

Linear-Quadratic regulator jest regulatorem optymalnym, czyli lepszym niż PID. Może być stosowany, gdy system dynamiczny jest opisany liniowym równaniem różniczkowym i koszt (który minimalizujemy) jest dany funkcją kwadratową.

44 Programowanie matematyczne

44.1 Programowanie liniowe z ograniczeniami

Rozwiązań ZPL należy szukać w wartościach ekstremalnych (wierzchołkach) zbioru ograniczeń.

Sympleks:

1. Wybieramy BRD (bazowe rozwiązanie dopuszczalne): dopełniamy ograniczenia do postaci standardowej, żeby były równości. Zmienne dopełniające tworzą BRD. Jeżeli rozwiązanie mamy dane z równościami to możemy albo wybrać macierz jednostkową, albo musimy dodać zmienne sztuczne. Jeżeli dodaliśmy zmienne sztuczne to musimy rozwiązać zadanie metodą dwufazową lub metodą kar ("dużego M"). W metodzie"dużego M" robimy sympleks raz, ale mamy inne drzewo interpretacji.

2. Obliczamy wskaźniki optymalności $(z_j - c_j)$.

3. Jeżeli $z_j - c_j \geq 0 \forall j = 1, 2, ..., n$ to aktualne BRD jest optymalne — STOP.

4. Badamy współczynniki a_{ij}. Jeżeli $\exists j z_j - c_j < 0$, przy którym $a_{ij} \leq 0$ dla $i \in B$, to zadanie nie posiada skończonego rozwiązania optymalniego — STOP

5. Szukamy niebazowej zmiennej, która wejdzie do bazy nowego BRD za pomocą kryterium wejścia $z_k - c_k = min\{z_j - c_j\}$.

6. Stosując kryterium wyjścia $\frac{b_r}{a_{rk}} = min\frac{b_i}{a_{ik}}$, znaleźć indeks zmiennej usuwanej z aktualnego BRD.

7. Uaktualniamy tabelkę (liczymy nowe BRD dla sąsiedniej bazy i obliczamy nowe a_{ij}).

8. Wracamy do 2.

Reguła Blanda chroni przed zapętlaniem się sympleksu (może się zapętlić gdy mamy zdegenerowane BRD – co najmniej jedna ze składowych części bazowej jest równa zeru). Od zwykłego sympleksu różni się innym kryterium wejścia i wyjścia (po prostu bierzemy najmniejszy indeks).

Zalety i wady sympleksu:

1. Ogólnie znany i często stosowany do rozwiązywania ZPL w praktyce.

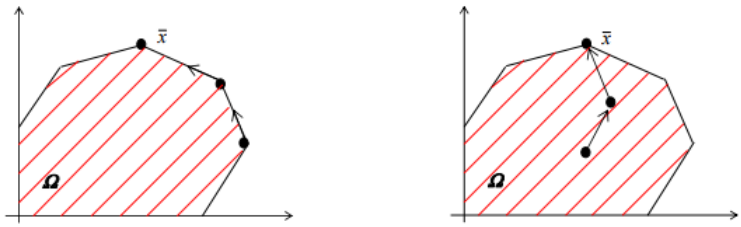
2. Złożoność pesymistyczna: wykładnicza (metoda punktu wewnętrznego wielomianowa).

3. Złożoność średnia: liniowa O((m+n)/2).

Zadanie dualne
Jeśli ZPL posiada dużą liczbę ograniczeń, a niewielką liczbę zmiennych ($m > n$), to warto zamiast niego rozważać zadanie dualne o niewielkiej liczbie ograniczeń i zwiększonej liczbie zmiennych decyzyjnych. Aby przejść do zadania dualnego zamieniamy: max na min, zmienne na ograniczenia i ograniczenia na zmienne.

Warunek komplementarności
Niech $\bar{x} = (x, x^d)$ oraz $\bar{y} = (y, y^d)$ – BRD odpowiednio dla ZP i ZD. Rozwiązania te są optymalne, gdy $x^T y^d + y^T x^d = 0$. Za pomocą warunku komplementarności możemy sprawdzić czy rozwiązanie jest optymalne (układ równań i rozkminy).

algorytm sympleks	***metoda punktu wewnętrznego***
Startując z punktu ekstremalnego zbioru Ω, metoda sympleks odwiedza kolejne punkty ekstremalne, kończąc w jednym z nich – będącym RO o optymalnej wartości funkcji celu.	Startując z pewnego punktu wewnętrznego zbioru Ω, w kolejnych krokach algorytmu przesuwamy się po punktach wewnętrznych w kierunku wierzchołka, w którym znajduje się RO.

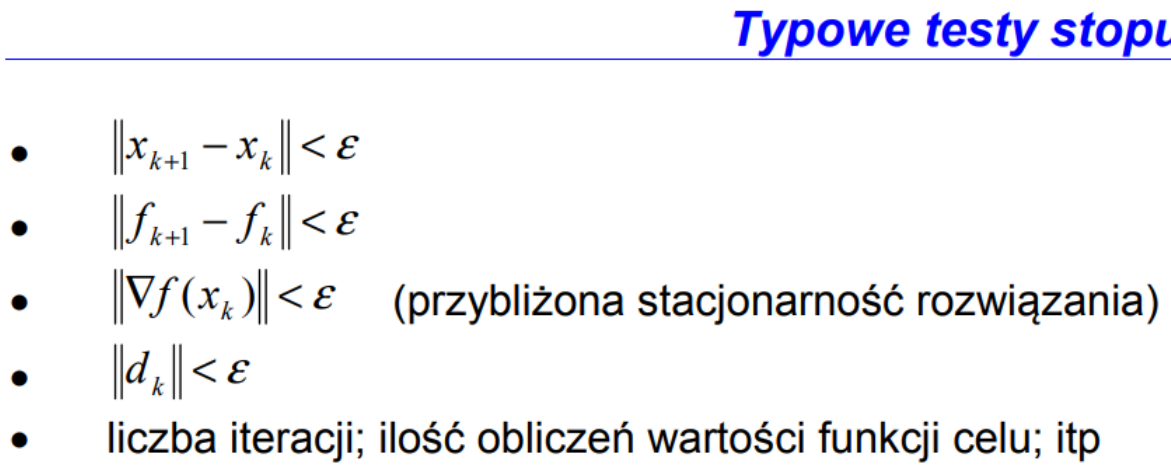


Rysunek 15: Porównanie algorytmu sympleksu z metodą punktu wewnętrznego

Mnożniki Lagrange'a
Za pomocą Mnożników Lagrange'a sprawdzamy czy dane rozwiązanie jest RO (rozwiązaniem optymalnym). Funkcja musi być wypukła. Dla danego rozwiązania x i dla zadanych ograniczeń sprawdzamy, czy są one aktywne. Są aktywne, gdy po wstawieniu do nich x zachodzi równość. Korzystając ze wzoru obliczamy λ i jeżeli dla każdego ograniczenia aktywnego $\lambda \geq 0$ to x jest RO.

44.2 Optymalizacja nieliniowa bez ograniczeń

Ekstremum funkcji mamy, gdy gradient jest równy 0. Jeżeli funkcja jest C^2 to możemy obliczyć jej Hessian (macierz drugich pochodnych) i sprawdzić jego określoność. W ten sposób sprawdzamy czy ekstremum jest minimum czy maksimum.



Rysunek 16: Typowe warunki stopu

Algorytmy optymalizacji nieliniowej bez ograniczeń
Każdy algorytm składa się z dwóch etapów. W pierwszym znajdujemy kierunek poszukiwań d_k, a w drugim długość kierunku α_k.

Metody minimalizacji jednokierunkowej (szukania α):

1. Analityczna – obliczenia symboliczne, dobre dla funkcji „prostej" postaci.
2. Eliminacji przedziałów (funkcja musi być unimodalna):
   (a) Metoda dychotomii (dwudzielności) – jak wyszukiwanie binarne tylko lekko zmodyfikowane. W każdej iteracji obliczamy dwie wartości funkcji i usuwamy połowę przedziału.
   (b) Metoda złotego podziału – wybieramy dwa punkty próbne w przedziale. W każdej iteracji obliczamy tylko jedną wartość funkcji i zmniejszamy przedział o współczynnik c.
   (c) Metoda Fibonacciego
   (d) Metoda punktu środkowego

3. Interpolacja funkcji celu:

   (a) Metoda Newtona

   (b) Interpolacja paraboliczna

4. Metody przybliżone:

   (a) Testy Goldsteina

   (b) Algorytm Armijo

Metody optymalizacji nieliniowej bez ograniczeń:

1. Bezgradientowa: Gaussa-Seidela: $d_i = e_i$. W każdej iteracji idziemy w n kierunkach bazowych i sprawdzamy czy jesteśmy w optimum. Uwagi: prosta obliczeniowo, wolno zbieżna, może zygzakować. Jest to gorsza wersja gradientów sprzężonych.

2. Gradientowe:

   (a) Metoda gradientu prostego: $d_k = -\nabla f(x_k)$, $\alpha_k = const.$

   (b) Metoda najszybszego spadku: $d_k = -\nabla f(x_k)$, α_k – na podstawie dokładnej lub przybliżonej minimalizacji kierunkowej. Uwagi: zbieżność liniowa, relatywnie najszybsze zbliżanie się do RO (podczas początkowych kroków), wolna zbieżność (zygzakowanie, czyli każdy kolejny kierunek jest do siebie prostopadły – wynika to z tego, że α jest minimalna), wrażliwa na błędy zaokrągleń.

   (c) Metoda gradientu sprzężonego: aby nie wybierać tego samego kierunku dwa razy (czyli usunąć zygzakowanie), szukamy zbioru kierunków A-sprzężonych, czyli zmieniamy układ współrzędnych. Dzięki temu dojdziemy do rozwiązania w maksymalnie n krokach (każdy z kierunków jest wykorzystywany tylko raz) dla funkcji kwadratowej. Dzięki tej metodzie możemy też rozwiązać układ równań liniowych $Ax = b$, gdzie A jest symetryczna i dodatnio określona.

3. Gradientowe drugiego rzędu:

   (a) Metoda Newtona: funkcja C^2. Przybliżamy funkcję F parabolą w punkcie (gdzie $F(\delta) = f(x_x + \delta)$, δ - krok i liczymy jej ekstremum.

Jest zbieżny kwadratowo (czyli szybki), ale większy nakład obliczeniowy w każdym kroku. W każdej iteracji obliczenie kierunku uwzględnia odwrócenie Hesjana.

Metoda Newtona

(met. drugiego rzędu)

Założenia $f \in C^2$, ściśle wypukła

$$f(x_k+\delta) \approx f(x_k) + \nabla f(x_k)^T \cdot \delta + \frac{1}{2}\delta^T \nabla^2 f(x_k)\delta + o(\delta^2) \overset{ozn}{\equiv} F_k(\delta)$$

metoda podstawowa: $x_{k+1} = x_k + d_k$

kierunek d_k:

$$d_k = \min_{\delta} F_k(\delta)$$

$$\nabla F_k(\delta) = 0$$

$$\nabla f(x_k) + \nabla^2 f(x_k) \cdot \delta = 0 \;\Rightarrow\; \delta = -\left[\nabla^2 f(x_k)\right]^{-1} \cdot \nabla f(x_k)$$

krok algorytmu

x_0 - p. startowy

$x_{k+1} = x_k + d_k$

$d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \cdot \nabla f(x_k)$

Rysunek 17: Metoda Newtona

(b) Metoda Newtona-Raphsona – jak Metoda Newtona, tylko minimalizujemy jeszcze α. Dzięki temu mamy większe obszary zbieżności.

(c) Metoda quasi-Newtonowskie – liczymy odwrotność Hesjana tylko raz, potem co pętle go poprawiamy. Metody poprawiania: DFP, BGFS.

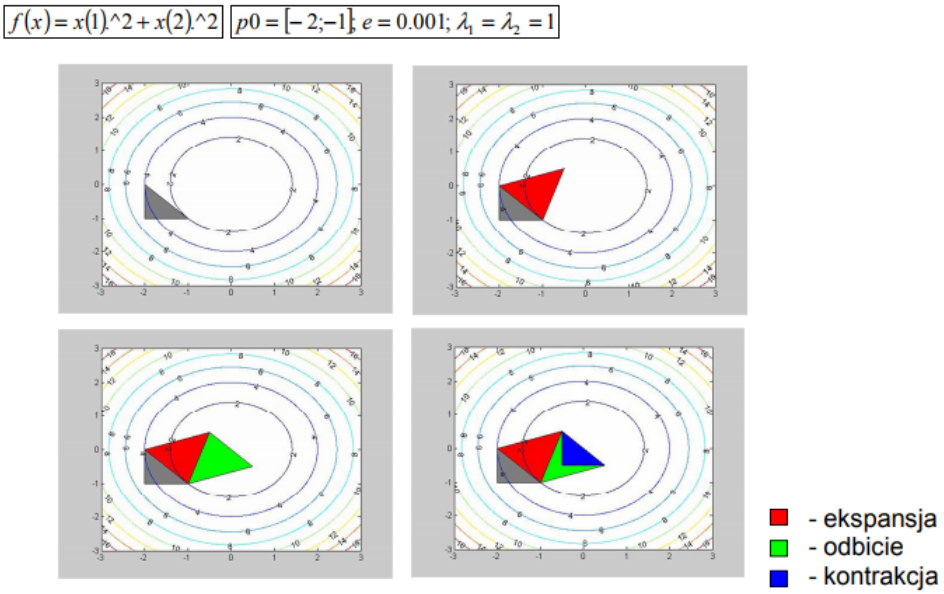
Metoda pełzającego sympleksu to metoda bezpośrednia heurystyczna.

Metoda Neldera-Meada

(„pełzającego sympleksu")

- poszukiwanie RO funkcji ***n*** zmiennych rozpoczyna się od skonstruowania ***n+1*** punktów stanowiących wierzchołki startowego ***sympleksu***
- startowy ***sympleks*** jest przekształcany w najbardziej obiecującym kierunku za pomocą elementarnych operacji geometrycznych:
  odbicie, ***ekspansja***, ***kontrakcja***, ***redukcja***
- po wykonaniu jednej z powyższych operacji, ***nowy sympleks*** można skonstruować przez odrzucenie „***najgorszego***" wierzchołka i zastąpienie go innym „***lepszym***"
  kolejne generowane sympleksy ***zbliżają się*** do lokalnego RO funkcji celu;
- w kolejnej iteracji wymagana jest ewaluacja ***jednej lub dwóch wartości*** funkcji celu, (periodycznie zaleca się wykonanie operacji ***redukcji***)

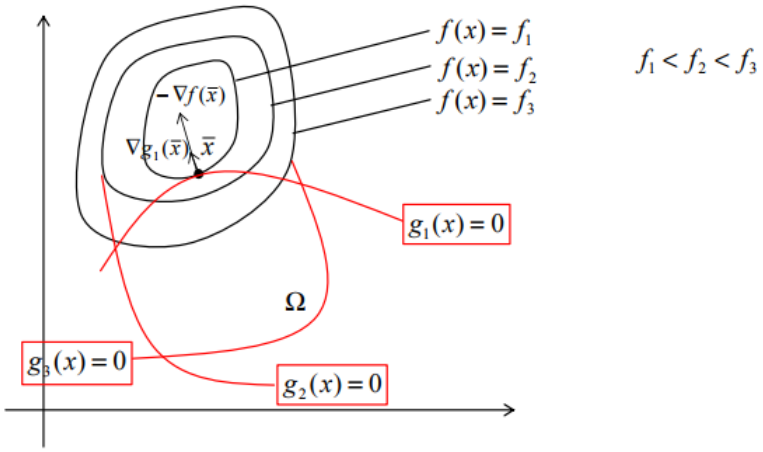
Rysunek 18: Pełzający sympleks



Rysunek 19: Pełzający sympleks - przykład

44.3 Optymalizacja nieliniowa z ograniczeniami

Obszar poszukiwań minimum jest teraz ograniczony ograniczeniami nieliniowymi. Kierunek dopuszczalny musi być wewnątrz lub na ściance zbioru ograniczającego.



Rysunek 20: Optymalizacja z ograniczeniami

Warunki Kuhna-Tuckera

Jeżeli ZPN jest regularne to WKT są konieczne, żeby punkt był ekstremum (był rozwiązaniem). Jeżeli zadanie jest wypukłe (funkcja celu jest wypukła, zadanie nie posiada nieliniowych ograniczeń równościowych, a ograniczenia nierównościowe są wypukłe) to WKT jest warunkiem koniecznym i wystarczającym.

Funkcja Lagarange'a

$$L(x, \lambda) = f(x) + \lambda^T (Ax - b)$$

Jest wykorzystywana do sprawdzenia czy dany punkt jest RO zadania otymalizacji nieliniowej z ograniczeniami. Sprawdzamy to za pomocą WKT.

Algorytmy programowania kwadratowego:

1. metoda eliminacji uogólnionej,
2. metoda zbioru ograniczeń aktywnych,
3. metoda rzutowania gradientu.

44.4 Metody funkcji kary

Gdy mamy zadanie optymalizacji nieliniowej z ograniczeniami, to możemy je przekształci na zadanie optymalizacji bez ograniczeń stosując funkcję kary.

$$minF_k(x, r_k)$$

gdzie

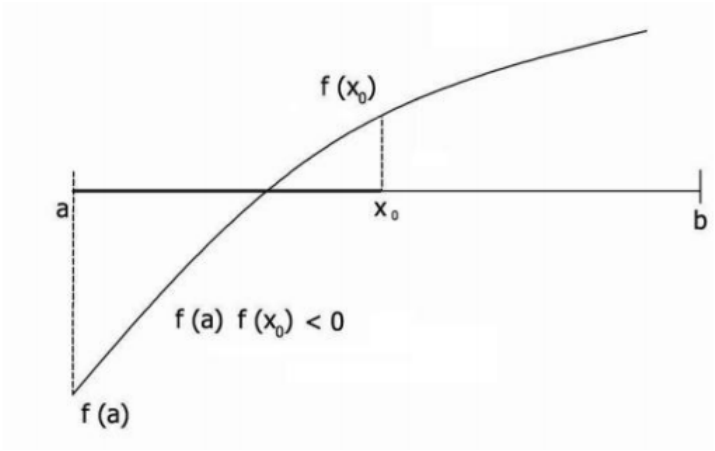
$$F_k(x, r_k) = f(x) + P_k(x, r_k)$$

Mamy różne metody wyznaczania funkcji kary: wewnętrzną, zewnętrzną, odmiany. Funkcja kary karze za używanie rozwiązań niedopuszczalnych. Podstawowym problemem związanym z funkcją kary jest kwestia jak bardzo karać – zbyt niska kara może doprowadzić do dominacji rozwiązań niedopuszczalnych, zbyt wysoka jej wartość odrzuci większość (jeśli nie wszystkie) rozwiązań niedopuszczalnych a wśród nich także te które mogły być bardzo cenne.

45 Metody numeryczne

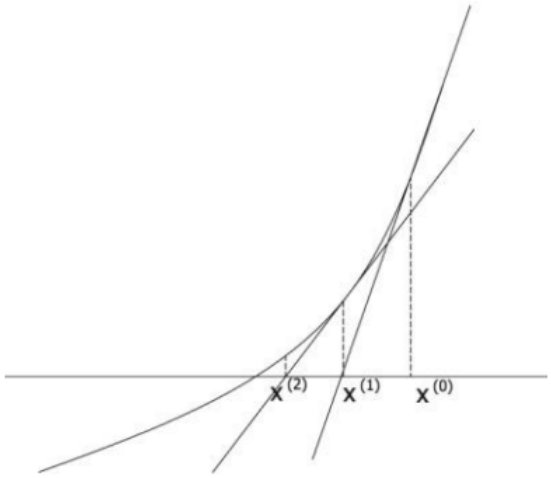
45.1 Poszukiwanie zer funkcji jednej zmiennej

Metoda bisekcji
Założenia: funkcja jest ciągła na przedziale $[a, b]$ i ma różne znaki na końcach przedziału.
Przedział $[a, b]$ dzielimy na połowę w punkcie x_0. Jeżeli x_0 nie jest pierwiastkiem to wybieramy jeden z przedziałów $[a, x_0]$ lub $[x_0, b]$. Wybieramy ten, na którym funkcja ma różne znaki na końcach. Zbieżność 1.



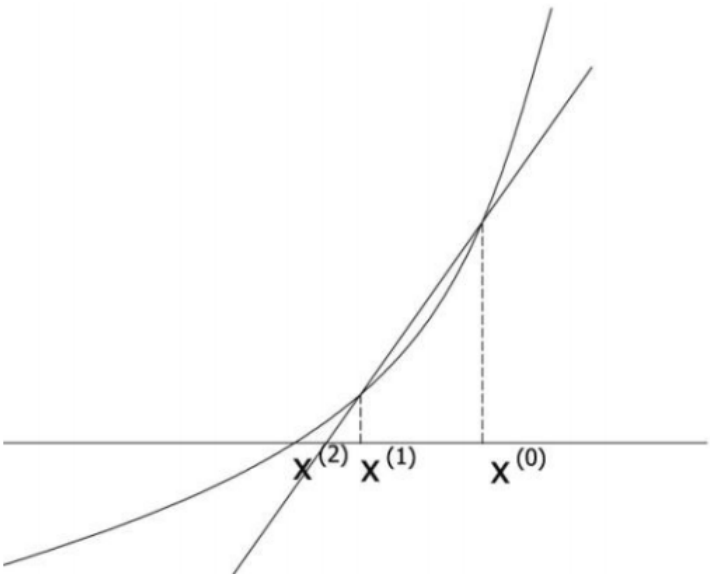
Rysunek 21: Metoda bisekcji

Metoda stycznych Newtona
Zbieżność kwadratowa.



Rysunek 22: Metoda Newtona

Metoda siecznych
Zbieżność 1.61.



Rysunek 23: Metoda siecznych

Często w **profesjonalnych solverach** używa się najpierw metody bisekcji, a potem metodę Newtona.

45.2 Metody całkowania

Całkujemy za pomocą różnych metod Newtona-Cotesa: metody prostokątów, metody trapezów, metody Simpsona (praboli). Metoda Simpsona interpoluje funkcję parabolą na trzech punktach za pomocą wielomianu Lagrange'a. Istnieje też metoda Monte-Carlo – funkcję zamykamy w kwadrat i strzelamy. Obliczamy stosunek liczny trafionych do liczby strzałów i na tej podstawie szacujemy wartość.

45.3 Metody rozwiązywania układów równań liniowych skończonych

Wzory Cramera

$$x_i = \frac{W_i}{W}$$

gdzie:
W – wyznacznik główny
W_i - wyznaczniki, w których współczynniki przy x_i zastępujemy wyrazami wolnymi.

Metoda eliminacji Gaussa
Metoda eliminacji Gaussa (złożoność $O(n^3)$) dzieli się na dwa etapy:

1. Sprowadzenie macierzy do postaci trójkątnej.
2. Rozwiązanie takiego układu.

Aby sprowadzić macierz do postaci trójkątnej możemy wykorzystać jedną z dwóch metod wyboru elementu głównego:

1. Wybór częściowy: w k-tym kroku największy element z k-tej kolumny z wierszy $k - n$ i zamieniamy wiersze (ewntualnie działamy na wierszach i zamieniamy kolumny).
2. Wybór pełny: maksymalnego elementu szukamy w podmacierzy i zamieniamy odpowiedni wiersz i kolumnę.

Te metody trzeba stosować, żeby uniknąć sytuacji, gdzie w czasie działania algorytmu mamy zero na przekątnej.

Rozkład LU

$$PA = LU$$

Rozkład LU możemy uzyskać za pomocą eliminacji Gaussa, ale bez "doklejania"do A wektora b. Wtedy z eliminacji otrzymujemy macierz schodkową, czylu U, a L to złożenie operacji elementarnych na A, które doprowadziłY do U. Znając rozkład L i U mamy złożoność $O(n^2)$.
Inną metodą rozkładu LU jest metoda Doolittle'a.

Rozkład Cholesky'ego-Banachawiecza
Jeżeli A jest macierzą symetryczną i dodatnio określoną to istnieje macierz trójkątna dolna L z dodatnimi elementami na głównej przekątnej:

$$A = LL^T$$

Jeżeli rozkład nie istnieje, to A nie może być macierzą symetryczną dodatnio określoną.

Zastosowania rozkładu LU: obliczanie wyznacznika, rozwiązywanie układu równań $Ax = b$, obliczanie A^{-1}.

45.4 Metody rozwiązywania układów równań liniowych iteracyjne

Niech x_0 będzie wektorem nazywanym przybliżeniem początkowym. We wszystkich metodach iteracyjnych tworzony jest ciąg kolejnych przybliżen, który dąży do rozwiązania x.

Metoda Jacobiego
Niech $a_{ii} \neq 0$ dla $i = 1, ..., n$. Przedstawiamy macierz A jako $A = U + L + D$. Liczymy x_k. Jeżeli A jest silnie diagonalnie dominująca to metoda jest zbieżna.

Metoda Gaussa-Seidela
Tak jak Jacobi, tylko mamy inny wzór na x_k. Metoda jest zbieżna w dwóch przypadkach:

1. jeżeli A jest silnie diagonalnie dominująca
2. $A = A^T$ i A jest dodatnio określona.

Metoda SOR
Metoda SOR jest uogólnieniem metody Gaussa-Seidela, w której wprowadzamy dodatkowy parametr relaksacji ω, z pomocą którego mieszamy wartość poprzedniego kroku z wartością nową $((1 - \omega)x_i + \omega x_{i+1})$.

Metoda gradientu sprzężonego

45.5 Metody rozwiązywania układów równań nieliniowych iteracyjne

Rozwiązywanie układów równań sprowadza się do znalezienia takiego rozwiązania $\alpha \in R^n$, w którym odwzorowanie $F : R^n \to R^n$ przyjmuje wartość zero, tzn. $F(\alpha) = 0$.

Metoda Newtona
Dla układów równań nieliniowych metoda Newtona działa jako uogólnienie dobrze nam znanej metody stycznych Newtona.

45.6 Wektory i własności własne

Metoda potęgowa znajduje największą co do modułu wartość własną i odpowiadający jej wektor. Nazywa się potęgowa, ponieważ jeżeli chcemy uzyskać k-te przybliżenie musimy obliczyć A^k. Jest stosowana w Google Page Rank.

$$Ax_0 = y_0$$
$$x_1 = \frac{y_0}{||y_0||}$$

Przekształcamy wektor dotąd aż nie będzie się zmieniać, więc będzie wektorem własnym. Zbieżność jest liniowa.

Metoda QR
Ideą metody QR jest doprowadzenie macierzy A do postaci $A = Q^T AQ$ za pomocą rozkładu QR. W każdym kroku macierz A rozkładamy na ortogonalną macierz Q i trójkątną macierz R. A więc ostatecznie

$$A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^T A_k Q_k$$

Macierz diagonalna A_k ma wartości własne na diagonali. Natomiast złożenie wszystkich macierzy Q_i ma wartości własne w kolumnach.

45.7 Metody rozwiązywania układ różniczkowych

Mamy dwa główne typy solverów: explicit i implicite. Żeby obliczyć kolejny stan metody explicite biorą pod uwagę jedynie stan obecny, a metody implicite stan obecny i stan przyszły. Mamy dwa główne solvery: Eulera i Rungego-Kutty. Metody Eulera (explicit i implicite) liczą rowzwiązanie w dyskretnych punktach. RK oblicza kilka kroków pomocniczych pomiędzy dwoma dyskretnymi wartościami. Ostatecznie wylicza wartość średniej ważonej kroków pomocniczych opartych na aproksymacji szeregiem Taylora zadanej funkcji. Możemy także powiedzieć, że metoda Eulera nie bierze pod uwagę krzywizny funkcji, więc dla różnych kroków całkowania zwraca różne wartości. RK bierze krzywiznę pod uwagę (jak bardzo to zależy od rzędu).