PL ISSN 0209-1593

P. 4201/82

2 1982

prace naukowo-badawcze

Instytutu Maszyn Matematycznych

rok XXIV

I n s t y t u t M a s z y n M a t e m a t y c z n y c h

P 4201/83

p r a c e n a u k o w o - b a d a w c z e

I n s t y t u t u

M a s z y n

M a t e m a t y c z n y c h

MOKRZYCKI W.:

Dyskretyzacja krzywych algebraicznych na jednorodnych kwadratowych siatkach k-spójnych s.3

MOKRZYCKI W.:

O pewnych problemach projektowania komputerowych systemów graficznych s.43

FAUDROWICZ A.:

Rozwiązywanie niesymetrycznych i rozrzedzonych układów równań liniowych s.46

Warszawa 1982

Copyright © 1982 - by Instytut Maszyn Matematycznych
Poland

Wszelkie prawa zastrzeżone

KOMITET REDAKCYJNY

dr inż. Stanisława BONKOWICZ-SITTAUER
doc. mgr Jan BOROWIEC
mgr Cezary DZIADOSZ /sekretarz redakcji/
doc. dr inż. Jan ŁYSKANOWSKI
doc. dr hab. inż. Stanisław MAJERSKI
doc. dr inż. Henryk ORŁOWSKI /redaktor naczelny/
dr inż. Piotr PERKOWSKI

Opracowanie redakcyjne: mgr Romana NITKOWSKA

Adres Redakcji: Instytut Maszyn Matematycznych
Branżowy Ośrodek INTE
02-078 Warszawa, ul.Krzywickiego 34
tel. 28-37-29 lub 21-84-41 w. 244

Druk IMM zam. 25/83 nakł. 400 egz.

Prace naukowo-badawcze Instytutu Maszyn Matematycznych

Dyskretyzacja krzywych algebraicznych na jednorodnych kwadratowych siatkach k-spójnych

Wojciech MOKRZYCKI

1. WSTĘP

1.1. Geneza zagadnienia

W wielu zastosowaniach wynika konieczność operowania przestrzenią dyskretną, w której są zdefiniowane jedynie punkty o współrzędnych całkowitoliczbowych. Dla przykładu można podać: komputerowe przetwarzanie rysunków liniowych, mapy topograficzne, monitory graficzne, kreślaki, drukarki znakowo - graficzne, obrabiarki sterowane numerycznie, w których to przypadkach płaszczyzną obrazowania /przestrzenią sterowania/ jest płaszczyzna /przestrzeń/ dyskretna, ze zdefiniowanymi węzłami całkowitoliczbowej siatki współrzędnych. Przedstawienie krzywych na takich płaszczyznach /lub określenie drogi ruchu narzędzia skrawającego w obrabiarkach sterowanych numerycznie/ odbywa się przez zadanie ciągu punktów, które /bądź to połączone kolejno, z reguły odcinkami linii prostej, bądź też bezpośrednio/ aproksymują zadaną krzywą. Punkty ciągu aproksymującego muszą mieć współrzędne całkowitoliczbowe, gdyż powinny pokrywać się z węzłami siatki współrzędnych.

W klasycznych metodach aproksymacji wielomianowej krzywych punkty aproksymujące wyznacza się dobrze znanymi metodami za pomocą funkcji trygonometrycznych lub rozwinięć na szeregi. Występujące tam operacje są długotrwałe i złożone, gdyż wymagają zmiennoprzecinkowej arytmetyki, a w wyniku dają współrzędne punktów będące liczbami rzeczywistymi. Dokładność, z jaką są współrzędne tych punktów wyznaczone, jest więc w pewnym sensie niewykorzystana. Punkty te są bowiem umiejscawiane w węzłach całkowitoliczbowej siatki współrzędnych, co wymaga zaokrąglenia rzeczywistych wartości współrzędnych punktów do liczb całkowitych. Z drugiej strony, gdyby dokładność wyznaczania współrzędnych była mniejsza, to w wyniku tego, że kolejne punkty są wyznaczane za pomocą wzorów rekursyjnych nie uwzględniających błędów powstałych we wcześniejszych krokach aproksymacji, kumulacja błędów doprowadziłaby /zwłaszcza przy długotrwałych obliczeniach/ do zupełnej "degeneracji" krzywych. Powstał więc problem opracowania takiego algorytmu, który wyznaczałby punkty aproksymujące z dokładnością tylko do liczb całkowitych, wykorzystywał jedynie proste operacje na krótkich liczbach całkowitych, był dostatecznie prosty i szybki oraz uwzględniał, podczas wyznaczania kolejnego punktu ciągu, błąd powstały na każdym kroku dyskretyzacji, w celu wyeliminowania kumulacji błędów. Przyjmuje się więc założenie, że punkty aproksymujące krzywą nie leżą na niej dokładnie. Powinna natomiast istnieć zależność pomiędzy warunkami

wyboru kolejnego punktu ciągu i odchyleniem poprzedniego punktu od żądanej krzywej.

1.2. Aktualny stan badań

Opisana wyżej całkowitoliczbowa dyskretyzacja krzywych jest stosunkowo mało znana i rzadko opisywana w literaturze. Z literatury znanych jest zaledwie kilkanaście tego typu algorytmów i dotyczą one z reguły dyskretyzacji prostych bądź okręgów. W większości z nich przyjęto założenie, że odległość między kolejnymi punktami aproksymującymi równa jest jednej lub $\sqrt{2}$ jednostek. Ograniczenie to, istotne w przypadku aproksymacji krzywych bezpośrednio punktami daje "mało gładki" kształt łamanej oraz dużą liczbę punktów aproksymujących. W przypadku, gdy nie ma ograniczenia na maksymalną odległość między kolejnymi punktami ciągu aproksymującego i łączy się je odcinkami prostej /np. w monitorach graficznych ze sprzętowymi generatorami wektorów/ można uzyskać "bardziej gładką" łamaną aproksymującą oraz znacznie mniejszą liczbę punktów aproksymujących.

W niniejszym opracowaniu zajmiemy się algorytmami tego drugiego typu, w których ograniczenie na odległość pomiędzy sąsiednimi punktami nie występuje. Co więcej, wprowadza się warunek otrzymywania maksymalnie możliwych odległości pomiędzy sąsiednimi punktami aproksymującymi, przy założonej dokładności aproksymacji.

W pierwszej ze znanych publikacji na temat całkowitoliczbowego algorytmu dyskretyzacji, nie ograniczającego odległości pomiędzy sąsiednimi punktami aproksymującymi krzywą [1] z 1972 r. Lappulainen i Ojala proponują stosunkowo prostą i szybką oraz dokładną metodę dyskretyzacji okręgu, opracowaną specjalnie do generowania testu-szablonu, pozwalającego rozpoznawać niedoskonałości obrazu telewizyjnego.

Zasada działania tego algorytmu jest następująca: jedna ze współrzędnych x bądź y ciągu punktów dyskretyzujących okrąg zmienia się na każdym kroku dyskretyzacji o jedną jednostkę. Przyrost wzdłuż drugiej ze współrzędnych jest zmienny, ale również jest całkowitoliczbowy. Kwadrat odległości kolejnych punktów ciągu od środka okręgu wzdłuż drugiej ze współrzędnych jest liniową funkcją promienia okręgu oraz numeru punktu ciągu i będzie liczbą całkowitą, gdy podwojony promień okręgu R będzie liczbą całkowitą. Ma on następującą postać:

$$L_{i+1} = L_i + 2R + 1 - 2i$$

$$L_1 = 0,\ i = 1, 2, \ldots, 2R$$

Powyższy wzór rekursyjny jest zupełnie prosty i błędów nie kumuluje. Pozostało jednak wyznaczenie, na podstawie L_i drugich współrzędnych punktu ciągu, a to wymaga wyznaczenia pierwiastka kwadratowego z liczby całkowitej L_i. Autorzy algorytmu zaproponowali bardzo naturalną i ekonomiczną metodę, polegającą na generacji sekwencji k^2+k $k=1,2,3,\ldots$ i porównywaniu każdej wartości $k_i^2+k_i$ z L_i. Jako wynik tego porównania jest brane to k_i, dla którego $k_i^2+k \geqslant L_i$. Kwadrat wielkości k otrzymywano metodą Brauna za pomocą następujących par rekursyjnych:

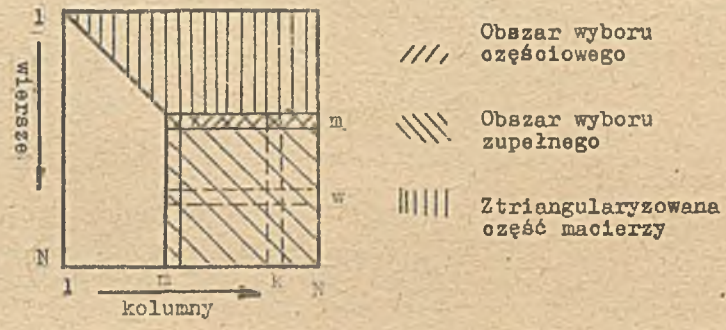
$$k_{i+1}^2 = S_{i+1} = S_i + a_{i+1},\ S_1 = 1$$

$$a_{i+1} = a_i + 2 \qquad ,\ a_1 = 1$$

Metoda powyższa jest bardzo prosta. W każdej pętli generacyjnej wymaganych jest zaledwie kilka operacji sumowania liczb /całkowitych/ oraz ich porównywania. Z tych też względów jest ona bardzo szybka. Jej

elementu w m-tym wierszu (lub w m-tej kolumnie), gdzie m jest numerem kroku eliminacji. Jest to tzw. częściowy wybór elementu głównego [11]. Wybór całkowity polega na przeszukiwaniu całej nieztriangularyzowanej części macierzy. Poszukiwanie w tym obszarze elementu największego co do modułu jest uzasadnione, gdy macierz współczynników nie jest rozrzedzona. Ponieważ jednak na dokładność wyników wpływa nie tylko wielkość elementu głównego, ale również liczba wykonywanych obliczeń (błędy zaokrągleń), przy rozwiązywaniu rozrzedzonych układów równań liniowych należy tak prowadzić obliczenia, aby w trakcie ich wykonywania powstała jak najmniejsza liczba nowych niezerowych elementów. Z drugiej strony, aby zmniejszyć błędy przy dzieleniu, element główny nie może być za mały (co do modułu). Dlatego w opisywanym programie zastosowano algorytm następujący [12]:

- w nieztriangularyzowanej części macierzy poszukuje się takiego wiersza, który zawiera tylko jeden niezerowy element (nie licząc kolumny wyrazów wolnych); jeżeli wiersz taki istnieje, niezerowy element wybiera się jako główny. Jeżeli wiersza takiego nie ma, wykonują się punkty B,C,D.



Rys.1. Wybór elementu głównego

- Z nieztriangularyzowanej części macierzy wybiera się element, którego wartość bezwzględna jest największa

$$A_{max} = MAX \left| A_{ik} \right| \; ; \quad i,k = m \div N. \tag{3}$$

m = numer kroku eliminacji

- W nieztriangularyzowanej części macierzy wyznacza się zbiór Q elementów A_{ik} takich, że

$$A_{ik} \in B \text{ gdy } \left| A_{ik} \right| > \frac{1}{4} A_{max} \tag{4}$$

- Ze zbioru Q wybiera się jako główny taki element, dla którego wartość

$$F = (n_k - 1) \cdot (n_w - 1) \tag{5}$$

jest najmniejsza;

n_k = liczba niezerowych elementów w tej części k-tej kolumny, która należy do nieztriangularyzowanej części macierzy

n_w = liczba niezerowych elementów w w-tym wierszu

k,w = m - N

Liczba 1/4 została wybrana empirycznie [12].

Zapamiętanie macierzy

Sposoby zapisu macierzy rozrzedzonej w pamięci komputera można podzielić na dwie grupy: listowe i tablicowe. Sposoby listowe charakteryzują się tym, że opis jednego elementu obejmuje jego wartość, identyfikator położenia elementu w macierzy oraz pointery do elementów poprzedniego i następnego (może być tylko jeden pointer, wtedy lista nazywa się jednokierunkową).

Sposób listowy zapewnia bardzo szybką aktualizację macierzy, jednak zajmuje więcej miejsca w pamięci komputera niż zapis tablicowy, w którym do opisu elementu nie wchodzą pointery. Aktualizacja macierzy w zapisie tablicowym zajmuje dużo czasu, gdyż elementy trzeba fizycznie przesuwać w pamięci komputera. Ponieważ opisywany program opracowano z myślą o użyciu w minikomputerze - zastosowano zapis tablicowy.

	A_{12}	A_{13}	B_1
A_{21}			B_2
A_{31}	A_{32}	A_{33}	

Tabl.										
kol	2	3	4	1	4	1	2	3	
A	A_{12}	A_{13}	B_1	A_{21}	B_2	A_{31}	A_{32}	A_{33}	
I	1	2	3	4	5	6	7	8	9
IPT	1		6	9	Tablica 2.				

Rys.2. Zapis macierzy rozrzedzonej

kol - numer kolumny, A - wartość elementu
I - wskaźnik, IPT - pointer początku wiersza

Składa się on z dwóch tablic. W pierwszej zapisano wierszami kolejne wartości niezerowych elementów oraz odpowiadające im numery kolumn, w drugiej - pointery wskazujące położenie pierwszych elementów w wierszach. Szczegóły wyjaśnia rys.2, na którym pokazano przykład zapisu małej macierzy. Aktualizacja tablicy jest dość pracochłonna:

- usunięcie elementu wymaga znalezienia jego położenia i przesunięcia w lewo o jedną pozycję wszystkich elementów znajdujących się na prawo od usuniętego
- dostawienie elementu wymaga znalezienia pozycji I_d, na której powinien on się znaleźć, przesunięcia o jedną pozycję w prawo wszystkich elementów o wskaźnikach $I \geqslant I_d$ oraz dostawienia nowego elementu na pozycję I_d

Jest oczywiste, że stosowanie takiego postępowania za każdym razem gdy jakiś element znika lub gdy powstaje nowy, wydłużyłoby bardzo znacznie czas rozwiązywania układu równań. Dlatego dane aktualizacyjne gromadzi się w jeszcze jednej pomocniczej tablicy i aktualizacji dokonuje się tak rzadko, jak to tylko możliwe, tzn. raz na krok eliminacji (czyli N razy).

Pozycję elementu lub miejsca aktualizacji znajdywano metodą bisekcji wiersza. Średni czas dostępu do elementu wynosi:

$$T_{bis} \sim 2+E\left(\lg_2 \frac{w}{2}\right)$$

gdzie:

T_{bis} - średni czas dostępu metodą bisekcji
w - - liczba elementów w wierszu
E - funkcja Entier

Przy bardzo małej liczbie elementów w wierszu, czas ten jest dłuższy od średniego czasu dostępu metodą przeszukiwania ($T_{prz} \sim \frac{w}{2}$), ale już przy ośmiu elementach w wierszu czasy te zrównają się a przy 32 elementach w wierszu

$$T_{prz} \approx 2,7 \cdot T_{bis}$$

okręgu i elipsy, która daje mniej więcej jednakowo odległe punkty na obwodzie /lub bardzo blisko konturu/ i stosuje jedynie stałoprzecinkową arytmetykę. Idea tej metody oparta jest na użyciu równań różniczkowych do opisu okręgu

$$\frac{dx}{d\theta} = -y, \quad \frac{dy}{d\theta} = x$$

i odpowiadających im równań różnicowych

$$x_{n+1} = x_n - y_n/k, \quad y_{n+1} = y_n + x_n/k$$

na podstawie których generowana jest elipsa, a nie okrąg, zbliżające się kształtem do okręgu w miarę wzrostu k.

Horn twierdzi, że zmiana tego układu równań do postaci

$$x_{n+1} = x_n - y_n/k, \quad y_{n+1} = x_{x+1}/k + y_n$$

i zastosowanie arytmetyki stałoprzecinkowej nie tylko nie pogarsza dokładności aproksymacji lecz odwrotnie, dla k z ustalonego zakresu punkty aproksymujące leżą w pewnym ograniczonym obszarze, jeszcze bliżej spodziewanej elipsy. Operacje algorytmu są rewersyjne, gdyż zachodzi:

$$y_n = y_{n+1} - x_{n+1}/k$$

$$x_n = x_{n+1} + y_n/k$$

s więc pierwszy powtórzony punkt pokrywa się z punktem początkowym.

Niezwykle istotny jest wybór k. Dla małych k aproksymacja elipsy jest całkiem ekscentryczna. Dla dużej liczby kroków zniekształcenie z powodu zaokrągleń jest tak duże, że elipsa staje się ośmiobokiem /przy k = max $(|x_o|, |y_o|)$, x_o, y_o - współrzędne punktu początkowego/. Dla ściśle określonych wartości k punkty przylegają całkowicie do rozważanego okręgu.

Dobrą aproksymacją na x' i y' są wyrażenia:

$$x' \simeq (x+y)\left(1 - \frac{1}{4k}\right)$$

$$y' \simeq (x-y)\left(1 + \frac{1}{4k}\right)$$

Jeśli k jest potęgą 2, wymagane są jedynie operacje sumowania i przesuwania w rejestrze.

Autor twierdzi również, że wzory powyższe zapewniają generację (jednakowo odległych) punktów na okręgu.

Tych pozytywnych opinii na temat generacji za pomocą algorytmu opisanego przez Horna opartej na równaniach różnicowych nie potwierdzają Maxwell i Baker - autorzy [14] z 1979 r. Ich zdaniem uzyskanie dokładności jednej j.r. generacji krzywych 2-stopnia w obszarze 1024x1024 j.r. wymaga zastosowania dokładniejszego wzoru kwadraturowego /z członem 2-rzędu/ oraz dostatecznie małego przyrostu parametru Δt, nie przekraczającego wartości 2^{-5}. Metoda aproksymacji wykorzystująca równania różnicowe kumuluje bowiem błędy już z definicji.

D.C.Sutcliffe w [9] z 1976 r. opisuje algorytm podobny do algorytmów z [5] i [16], który do aproksymacji dowolnej krzywej stosuje prostoliniowe odcinki o długości $\sqrt{2}$ i 2 jednostki, oraz wykorzystuje własność funkcji f (x,y), która przyjmuje różne znaki po obu stronach konturu. Działanie tego algorytmu jest następujące.

Podlegający analizie rejon określoności funkcji $f(x,y) = 0$, tj. obszar na którym ma być wykreślona funkcja $f(x,y)$ dzielony jest na prostokąty /komórki/, których boki są równe podwojonej długości kroku plotera. Proces wyznaczania krzywej dyskretnej polega na tym, że znajduje się bok komórki, który jest przecinany przez krzywą $f(x,y) = 0$, a do środka tego boku przemieszcza się głowicę plotera. Przecięty bok komórki jest nazywany linią bazową. Jest ona podstawą do tworzenia następnej komórki, w której będzie odbywać się śledzenie położenia krzywej.

Śledzenie to polega na wyznaczaniu wartości funkcji w obu rogach komórki przeciwległych do linii bazowej i wybranie na następny węzeł aproksymujący środka tego boku komórki, który przecinany jest przez krzywą. Przecięty bok komórki staje się linią bazową /podstawą/ dla następnej komórki poddawanej badaniu i proces powtarza się, aż do przekroczenia granicy rejonu, w którym funkcja jest aproksymowana.

Tak więc w aproksymacji krzywej metoda korzysta z trzech podstawowych przemieszczeń:
- o 2 kroki w kierunku jak poprzednio, do środka przeciwległego boku komórki,
- o $\sqrt{2}$ kroków w kierunku zmienionym o 45°, do środka przyległego boku komórki,
- o $\sqrt{2}$ kroków w kierunku zmienionym o -45°, do środka przyległego boku komórki.

Autor publikacji poświęca również sporo miejsca problemowi tzw. zdegenerowanych komórek, tj. takich, których wszystkie cztery boki są przecinane przez kontur krzywej $f(x,y)$. Zarzuca on badaczom, że unikają rozważania tego problemu lub, w najlepszym razie, rozważają tylko niektóre jego przypadki, lekceważąc inne. Odnosi się to zwłaszcza do przypadku przecinania się wewnątrz komórki różnych gałęzi krzywej. Sutcliffe sugeruje, żeby powyższy przypadek rozstrzygnąć opierając się na zasadzie minimalnej zmiany kierunku przemieszczenia, że jeśli należy wybrać któryś z boków komórki przecinanych przez kontur $f(x,y)$ to wybieramy ten, którego osiągnięcie wymaga jak najmniejszej zmiany kierunku ruchu /w stosunku do ruchu poprzedniego/.

Odmienny od poprzednich algorytm całkowitoliczbowej aproksymacji krzywych 2-stopnia zaproponowałem w pracy [11] z 1977 r. oraz [12] z 1978 r./algorytm aproksymacji na siatce n-spójnej/. Algorytm ten wyznacza ciąg całkowitoliczbowych punktów aproksymujących /dyskretyzujących/ dowolną z krzywych 2-stopnia i działa wg zasady, że węzeł aproksymujący powinien być możliwie najbliżej krzywej oraz że odcinek łamanej wyznaczony przez dwa kolejne węzły ciągu aproksymującego powinien najściślej przylegać do krzywej i być możliwie najdłuższym. Za miarę oddalenia punktu od krzywej przyjąłem moduł wartości funkcji w tym punkcie.

Odległość pomiędzy punktami aproksymującymi nie jest z góry ograniczona. Jedna ze współrzędnych x bądź y kolejnego punktu ciągu różni się od tejże współrzędnej punktu poprzedniego najwyżej o jedną jednostkę. Druga ze współrzędnych - o całkowitą liczbę jednostek w zależności od parametrów geometrycznych krzywej /lokalnej krzywizny/ oraz położenia krzywej względem układu współrzędnych.

Wyznaczanie kolejnego punktu aproksymującego krzywą odbywa się w następujący sposób. Poczynając z bieżącego punktu ciągu wykonuje się dwa typy elementarnych ruchów /ruchów do punktów sąsiednich odległych o jedną lub $\sqrt{2}$ jednostek/: jeden tzw. "ruch od krzywej" oraz szereg tzw. "ruchów wzdłuż krzywej". W każdym z punktów osiąganych w wyniku wykonania elementarnego ruchu wyznacza się moduł wartości funkcji /i jej wszystkich pochodnych cząstkowych/ i porównuje się go z modułem wartości funkcji w punkcie poprzednim. Kierunek następnego elementarnego ruchu nie ulega zmianie, gdy moduł wartości funkcji w osiąganym węźle jest mniejszy niż w punkcie poprzednim. W przeciwnym przypadku następuje

modyfikacje kierunku ruchu /o kąt $\pm \frac{\pi}{2}$ zgodnie z przyjętą zasadą/, aby uzyskać zmniejszanie się modułu wartości funkcji lub przecięcie krzywej.

Ruch "od krzywej" jest zawsze ruchem do następnego węzła wzdłuż przekątnej /odległego o $\sqrt{2}$ jednostek/. Ruchy "wzdłuż krzywej" są ruchami wzdłuż tej ze współrzędnych x bądź y, która tworzy z wektorem kierunkowym stycznej, wystawionym w punkcie bieżącym mniejszy kąt. "Ruchy wzdłuż krzywej" wykonywane są aż do przecięcia krzywej. Na kolejny węzeł aproksymujący wybierany jest ten z dwóch sąsiednich punktów /odległych od siebie o jedną jednostkę i leżących po przeciwnych stronach krzywej/, w którym moduł wartości funkcji jest mniejszy.

Ten ostatni algorytm ma przewagę nad algorytmami poprzednio opisanymi przede wszystkim ze względu na uniwersalność /aproksymuje dowolne krzywe 2-stopnia/ i dokładność /błędy nie większe niż $\sqrt{2}/2$ jednostki/. Zapewnia również jednoznaczność aproksymacji krzywych niezależnie od kierunku obiegu krzywej. Ma on jednak istotne wady wynikające z założenia, że niezależnie od położenia krzywej względem układu współrzędnych "ruch od krzywej" odbywa się zawsze wzdłuż przekątnej, natomiast "ruchy wzdłuż krzywej" - zawsze wzdłuż kierunku jednej z osi x bądź y. Powoduje to, że łamana aproksymująca odcinek krzywej przebiegający równolegle /w przybliżeniu/ do przekątnej układu współrzędnych jest mało "gładka" i złożona jest z dużej liczby krótkich odcinków.

W publikacji [13] z 1978 r. Ward opisuje metodę zgrubnej aproksymacji konturów, niepodobną do żadnej z metod poprzednio opisanych. Jest to metoda obrazowania funkcji $f(x,y) = C$ na urządzeniu znakowym /np. drukarce/.

Koncepcja metody jest następująca: obszar, w którym $f(x,y)$ ma być wykreślona dzieli się na mniejsze podobszary

$$x_i < x < x_{i+1}, \quad y_j < y < y_{j+1}, \quad i=0-n, \quad j=0-m,$$

mające położenie (i,j) oraz wymiary $S_x \times S_y$ identyczne jak wymiary pola znaku. Następnie bada się położenie funkcji f w kolejnych podobszarach i w tych podobszarach przybliża się ją odpowiednimi znakami graficznymi. Problemem jest wybranie z małego, stałego alfabetu symbolu graficznego, który będąc wykreślonym w analizowanym podobszarze, będzie w nim najściślej przylegał do konturu funkcji $f(x,y) = C$.

Najprostsze /1-rzędu/ rozwiązanie zagadnienia dopasowania wymaga jedynie sześciu znaków graficznych i próbkowania wartości funkcji f w czterech rogach badanego podobszaru. Gdy badanie funkcji w całym obszarze obrazowania odbywa się wg metody rozwinięcia rastrowego, wtedy liczba próbek funkcji f w całym obszarze obrazowania wynosi $(R_x+1) \times (R_y+1)$, gdzie R_x i R_y są rozdzielnościami powierzchni obrazowania mierzonymi liczbą znaków.

Znak graficzny przybliżający krzywą w danym obszarze jest wybierany na podstawie kombinacji położenia dodatnich i ujemnych znaków wartości funkcji w rogach badanego podobszaru.

Dokładność odwzorowania funkcji $f(x,y)$ opisanym sposobem jest ograniczona. Poza tym pojawiają się lokalne dwuznaczności np. w przypadku przecinania się linii konturu.

Lepsze dopasowanie znaku do położenia konturu można uzyskać wykonując próbkowanie funkcji f w większej liczbie punktów obszaru znaku i rozszerzając liczebność zbioru znaków graficznych. Dopasowanie 2-rzędu wymaga próbkowania funkcji f również w środkowych punktach obrzeża pola znaku. Otrzymuje się to za pomocą 24 odpowiednio dobranych znaków graficznych. Wynik porównania między C i każdą próbką f określa, które z brzegów pola znaku są przecinane przez kontur; określa się zatem punkty

ntrajne znaku dopasowującego.

W ogólności, dopasowanie n-rzędu wymaga operowania zbiorem $6n^2$ znaków graficznych i próbkowania funkcji f n razy na każdej z czterech granic pola znaku.

Interesujący algorytm dyskretyzacji parametrycznych krzywych giętkich /spline'ów/ 3-stopnia zaproponowali R.Moss i A. Lindgurd w [15]/1979 r./.Algorytm ten, opracowany do wykreślania znaków alfanumerycznych za pomocą plotera posługuje się jedynie arytmetyką całkowitoliczbową.

Idea tego algorytmu jest następująca: kształty znaku alfanumerycznego zadaje się za pomocą niewielkiej liczby n węzłów siatki, tzw. punktów charakterystycznych:

$$S_i = \{S_x(t), S_y(t)\} = \{x_i, y_i\}, \quad t_i = (i-1)\cdot h, \qquad h = \text{const}.$$

Przez punkty te prowadzi się parametryczną krzywą giętką /spline/ 3-stopnia, nadającą znakom płynną zmieniające się, przyjemne dla oka kształty. Postać tej funkcji jest następująca:

$$S(t) = S_{i-1} + (S_i - S_{i-1})\,\frac{t-t_{i-1}}{h} + \frac{h^2}{6} M_{i-1}\left[\left(\frac{t_i-t}{h}\right)^3 - \frac{t_i-t}{h}\right] +$$

$$+ \frac{h^2}{6} M_i\left[\left(\frac{t-t_{i-1}}{h}\right)^3 - \frac{t-t_{i-1}}{h}\right], \qquad t_{i-1} \leqslant t \leqslant t_i .$$

M_i jest pochodną funkcji giętkiej i wyznacza się ją na podstawie pewnych wzorów /rekursyjnych/ wynikających z teorii funkcji giętkich; S_i jest wartością funkcji w punkcie charakterystycznym, a h jest przyrostem parametru t przy wyznaczaniu punktów na krzywej giętkiej.

Ta forma zapisu krzywej uwidacznia, że krzywa zawiera /między punktami charakterystycznymi/ segmenty prostoliniowe z nałożoną na nie korekcją 3-stopnia nadającą krzywej gładkość.

Na krzywej giętkiej wyznacza się z kolei punkty:

$$T_p = \{2h^3 S_x(p),\ 2h^3 S_y(p)\}, \quad p \equiv t_p = 1,2,\ldots,h,h+1,\ldots,nh,$$

które sprowadzone do najbliższych węzłów siatki dają dyskretną formę tej krzywej. Dla wygody obliczeniowej w realizacji algorytmu dokonuje się pewnych przekształceń. Zaokrąglając wielkość $K_i = \frac{1}{3} h^2 M_i$ $i=1,\ldots,$ h do najbliższej liczby całkowitej i wykonując odpowiednie przekształcenia można dojść do wygodnej postaci/dla celów dyskretyzacji/ zapisu krzywej

$$2h^3 S(t) = 2h^3 S_{i-1} + 2h^2 (S_i - S_{i-1})p + K_{i-1}q(q^2-h^2) + K_i p(p^2-h^2)$$

dla $t_{i-1} \leqslant t \leqslant t_i$, $p = h-q = 0,1, \ldots, h$, $p = t-t_i$, $q = t_i - t$.

Dla wyznaczonych wartości p, $2h^3 S(t)$ przyjmuje wartości całkowite, równe S_{i-1}, S_i dla p=0, h.

Wyrażenie to musi być wyznaczone dla parametru t przyrastającego o h z wykorzystaniem standardowego schematu różnicowego, ze stałą różnicą 3-rzędu $\Delta S_i^{(3)} = 6(K_i - K_{i-1})$, i=2, ..., n.

Wartość ta i różnice niższego rzędu muszą być obliczone dla każdego segmentu krzywej, wyznaczonej przez kolejne punkty charakterystyczne. Dla różnic 1- i 2-rzędu wyprowadza się z równania ogólnego wzory:

$$\Delta S_i^{(2)}(p) = 6K_i(p+1) + 6K_{i-1}(q-1)$$

$$\Delta S_i^{(1)}(p) = 2h^2 (S_i - S_{i-1}) + (h^2-1)(K_{i-1} - K_i) + 3K_i p(p+1) - 3K_{i-1} q(q-1).$$

Dyskretyzacja krzywej gładkiej wykonywana jest następująco: wyliczana jest różnica między punktem

bieżącym /aktualnie osiągniętym węzłem siatki/ i wyznaczonym punktem na krzywej gładkiej obliczonym przy przyroście parametru t o stałą wartość h. Ruch do nowego punktu następuje, gdy ta różnica przekracza moduł siatki. W trakcie tych operacji jest wyznaczona wielkość kontrolna $2h|S(t) - S(pióra)|$ za pomocą schematu różnicowego. Jest ona dokładnie równa zero jedynie w punktach charakterystycznych. Gdy h jest duże, wtedy wyznaczyć trzeba więcej niż jeden punkt na krzywej giętkiej zanim nastąpi ruch do nowego punktu. Gdy h jest małe - kolejne punkty na krzywej gładkiej odległe są od siebie o kilka modułów siatki. Górna granica liczby kroków, które punkt bieżący wykonuje zanim osiągnie następny punkt na krzywej gładkiej, określa się wyrażeniem $4\Delta S/h$ /ΔS - maksymalna różnica między sąsiednimi punktami charakterystycznymi/.

Maksymalne odchylenie między łamaną aproksymującą i krzywą giętką nie przekroczy modułu siatki, gdy zachowana zostanie zależność $h>\sqrt{\Delta S}$. Maksymalna liczba kroków pomiędzy kolejnymi węzłami na krzywej gładkiej wyniesie wtedy $4\sqrt{\Delta S}$.

Dyskretyzację krzywej giętkiej można więc przeprowadzić za pomocą dwóch różnych schematów:
- schemat pojedynczego kroku, gdy $h > 4(\Delta S)$ oraz
- schemat linii prostej, gdy $h > \sqrt{\Delta S}$.

Maksymalna różnica pomiędzy łamaną aproksymującą i krzywą giętką wyznaczona dla parametru p=0, 1,...,h określana jest poniższym wzorem:

$$|\varepsilon| \leqslant \frac{3}{2}h\Delta S = \frac{3}{4h}\,\frac{\Delta S}{h}$$ jednostek siatkowych.

Metoda powyższa jest złożona, a poza tym nie zapewnia symetrii krzywym dyskretnym, które są postaciami symetrycznych krzywych gładkich. Zasługuje jednak na szczególną uwagę, gdyż jest jedyną znaną z literatury całkowitoliczbową metodą dyskretyzacji parametrycznych krzywych giętkich na siatce, z oszacowanym błędem aproksymacji.

Ogólne podejście do problemu dyskretyzacji krzywych sformułowali H.Freeman i J.A.Saghvi w [17] /1980 r./. Opisują oni metodę kwantyzacji wg tzw. schematu uogólnionego łańcucha kodowego. Łańcuch kodowy jest ciągiem kodów przemieszczeń do kolejnych punktów - węzłów aproksymujących krzywą. Punkty aproksymujące wyznacza się w wyniku nałożenia na krzywą siatki i wybrania węzłów tej siatki najściślej przylegających do krzywej. Odległość między węzłami aproksymującymi nie może być większa niż założona.

Dyskretyzację wg schematu uogólnionego łańcucha kodowego realizuje się w dwóch etapach:

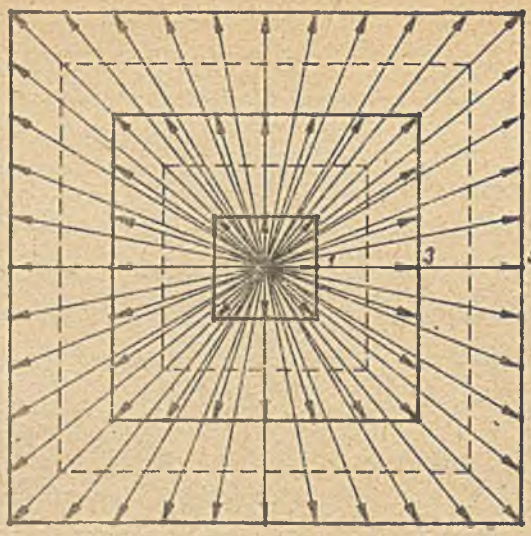
1/ określa się /dla pewnego węzła x_p, y_p, będącego punktem początkowym segmentu aproksymowanego i pewnej dopuszczalnej długości odcinka aproksymującego krzywą/ zbiór szablonów;

2/ wybiera się taki szablon, który w całości zawiera pewien, możliwie najdłuższy odcinek aproksymowanej krzywej.

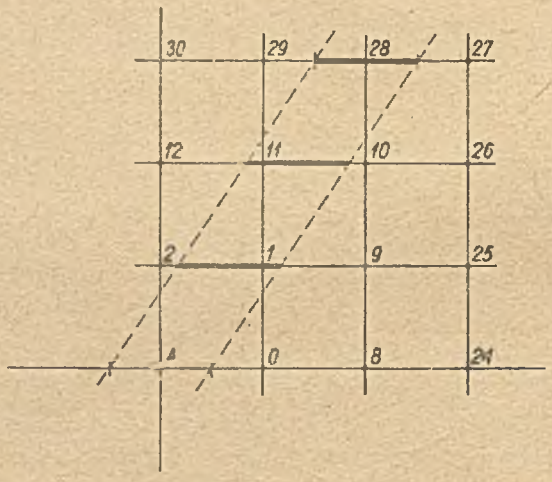
Każdy z dwóch etapów zawiera wiele czynności.

Określenie zbioru szablonów wykonywane jest w następujący sposób:

1/ łączy się ze sobą węzły siatki tak, aby powstawały koncentryczne względem węzła $x_p y_p$ kwadratowe pierścienie; liczba k=i·h /h - moduł siatki, i - naturalna/ określa numer pierścienia, a tym samym i dozwoloną maksymalną długość odcinka aproksymującego; położenie tych pierścieni oraz ich liczba określają sposób kodowania przemieszczeń do kolejnych węzłów aproksymujących; na rys. 1.1. pokazano położenie pierścieni dla kodu przemieszczeń 1-3-5;



Rys.1.1. Położenie pierścieni kodowych i wektorów przemieszczeń dla kodu przemieszczeń 1 - 3 - 5



Rys.1.2. Szablon dla pierścienia 3, węzła 28

2/ dla każdego węzła na każdym z pierścieni prowadzi się dwie równoległe linie w stronę punktu x_p, y_p, tak aby przechodziły przez punkty środkowe ogniw pierścienia przylegających do danego węzła, a węzły na pierścieniu oraz węzeł x_p, y_p leżały między tymi liniami w równej od nich odległości; linie te wycinają na pierścieniach odcinki /bramki/, które tworzą szablon; dla pierścienia k istnieje 8k szablonów /na rys. 1.2. pokazano szablon dla pierścienia 3 i węzła 28/.

Wybranie szablonu zawierającego określony odcinek aproksymowany krzywą rozpoczyna się od szablonów zewnętrznego pierścienia i kontynuuje się je dla pierścieni coraz to mniejszych, w sposób następujący:

1/ wybiera się dowolny szablon pierścienia największego i sprawdza się przechodzenie aproksymowanej krzywej przez wszystkie odcinki danego szablonu /przecinanie się krzywej ze wszystkimi odcinkami

określonego szablonu/; jeżeli krzywa przechodzi, wówczas odpowiedni węzeł pierścienia największego jest kolejnym punktem aproksymującym, jeżeli nie - wybiera się kolejny /przylegający/ szablon pierścienia /największego/ i operację powtarza się aż do wyczerpania się wszystkich szablonów pierścienia dużego;

2/ jeżeli krzywa przechodzi przez pewien odcinek pierścienia zewnętrznego lecz nie przechodzi przez którykolwiek odcinek pierścieni mniejszych, to wówczas wybiera się szablony pierścienia wewnętrznego i operację powtarza.

Algorytm ten zapewnia błąd aproksymacji nie większy niż $\frac{h}{2}$ i produkuje możliwie najdłuższe z dowolnych odcinki aproksymujące.

Zaproponowany schemat aproksymacji jest uniwersalny i dokładny. Ponadto obejmuje on aproksymację dowolnych krzywych analitycznych. Ma jednak podstawową wadę: nie nadaje się do realizacji wprost, a może jedynie służyć jako podstawa do opracowania szczegółowych algorytmów dyskretyzacji obejmujących dokładniej sprecyzowane klasy krzywych, uwzględniających sposoby ich przedstawiania.

W niniejszym opracowaniu proponuję dwa inne algorytmy całkowitoliczbowej dyskretyzacji krzywych, które są wolne od opisanych wcześniej wad, a także są bardziej uniwersalne, gdyż dotyczą aproksymacji dowolnych płaskich krzywych algebraicznych i opisane są w dalszej części tego opracowania.

2. DYSKRETYZACJA KRZYWYCH NA SIATKACH KWADRATOWYCH

2.1. Sformułowanie zagadnienia dyskretyzacji

Niech w prostokątnym układzie współrzędnych ϕXY będzie zadana krzywa algebraiczna n-stopnia względem zmiennych x i y w postaci uwikłanej.

$$
\begin{aligned}
F(x,y) = {} & a_{n,\phi}x^n + a_{n-1,1}x^{n-1}y + \ldots\ldots + a_{1,n-1}xy^{n-1} + a_{\phi,n}y^n + \\
& + a_{n-1,\phi}x^{n-1} + \ldots\ldots + a_{1,n-2}xy^{n-2} + a_{\phi,n-1}y^{n-1} + \\
& \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
& + \ldots\ldots + a_{1,n-1}xy^{n-2} + a_{\phi,n}y^r + \\
& \quad \vdots \\
& + a_{1,\phi}x + a_{\phi,1}y + \\
& + a_{\phi,\phi} = \\
= {} & \sum_{k=\phi}^{n} \sum_{i=\phi}^{n-k} a_{n-i-k,i}\, x^{n-i-k} y^{i} = \phi
\end{aligned}
\qquad /2.1/
$$

i niech w tym układzie będzie zadana również jednorodna, kwadratowa siatka z modułem h, za pomocą linii [12] :

$$x = h\,l$$

oraz

$$y = \left(\frac{p}{q}\right)x + h\cdot i \qquad /2.2/$$

gdzie: h - moduł siatki /liczba rzeczywista/, l = 0, ±1, ± 2 ± 3 ...,

Wprowadźmy pomocnicze definicje:

Definicja 2.1

Rząd spójności siatki kwadratowej określamy jako liczbę półprostych siatkowych wychodzących z węzła siatki.

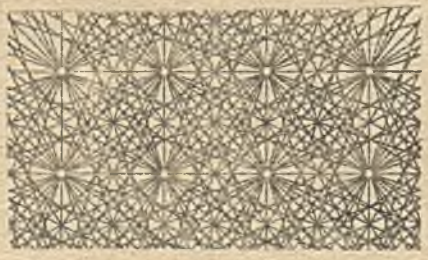
Definicja 2.2

Bezpośrednio spójnymi węzłami siatki nazwiemy takie dwa węzły, które połączone są linią siatkową bezpośrednio /bez pośrednictwa innych węzłów/.

Definicja 2.3

Głównymi liniami siatki nazwiemy te linie, wzdłuż których odległość pomiędzy bezpośrednio spójnymi węzłami siatki równa jest h, liniami dwu-przekątnymi - gdy ta odległość jest równa $\sqrt{2}h$, p+r - przekątnymi - gdy jest ona równa $\sqrt{p^2+r^2}\,h$.

Przyjmując za p i q dowolne liczby całkowite /ale $q \neq 0$ / otrzymujemy siatkę k-spójną /$k=2^l$, l - liczba naturalna/. Na rys. 2.1 przedstawiona jest siatka 32-spójna.



Rys. 2.1. Siatka kwadratowa 32-spójna

Poszukujemy algorytm dyskretyzacji krzywej algebraicznej płaskiej n-stopnia określonej równaniem /2.1/, najlepszy w sensie minimum odległości pomiędzy tą krzywą a łamaną, otrzymaną z połączenia prostoliniowymi aproksymantami ciągu bezpośrednio spójnych węzłów siatki /k-spójnej/ przybliżających krzywą.

Metryka dyskretyzacji

Kryterium najmniejszej odległości pomiędzy krzywą zadaną i łamaną aproksymującą postawiona

w zagadnieniu aproksymacji, wymaga obliczenia odległości pomiędzy tymi liniami, co jest operacją dość złożoną. W tej pracy /podobnie jak w pracy [12]/, jako miarę oddalenia węzła siatki od krzywej przyjmiemy moduł wartości funkcji $F(x,y)$ w rozważanym węźle. Spełnia on bowiem lokalnie warunki metryki i daje się stosunkowo łatwo wyznaczyć.

Algorytm dyskretyzacji

Algorytm dyskretyzacji algebraicznej krzywej płaskiej $F(x,y)=\phi$ i jej aproksymowanie prostoliniowymi aproksymantami pokrywającymi się z odcinkami linii siatkowych, powinien zawierać ciąg kolejnych czynności przemieszczania się po liniach siatkowych wzdłuż krzywej, od punktu początkowego x_p,y_p do punktu końcowego x_k,y_k, w obranym na krzywej kierunku i wyznaczania ciągu bezpośrednio spójnych węzłów siatki takich, aby łamana uzyskana z połączenia prostoliniowymi aproksymantami kolejnych węzłów tego ciągu była jak najmniej odległa /w sensie metryki generacyjnej $|F(x,y)|$/ od krzywej zadanej, przy możliwie najmniejszej liczbie węzłów aproksymujących.

2.2. Analiza zagadnienia dyskretyzacji

Niech krzywa $F(x,y)=\phi$ będzie w rozważanym obszarze analityczna za wyjątkiem co najwyżej skończonej liczby punktów /osobliwych/ i niech ma w obszarze swojej określoności ciągłe, ograniczone i jednocześnie nierówne zeru pochodne cząstkowe 1-rzędu. Określimy /nieformalnie/ za pomocą tych pochodnych kierunek ruchu wzdłuż stycznej do krzywej w sposób następujący [12] :

$$\vec{V}^{+}=\vec{V}^{+}(V^x,V^y)=\left(-\frac{\partial F}{\partial y},\frac{\partial F}{\partial x}\right)$$

$$\vec{V}^{-}=\vec{V}^{-}(V^x,V^y)=\left(\frac{\partial F}{\partial x},\frac{\partial F}{\partial y}\right) \qquad /2.3/$$

Przyjmijmy następujący uproszczony zapis pochodnych cząstkowych funkcji $F(x,y)$:

$$F_{rx}\overset{def}{=}\frac{\partial^r F}{\partial x^r},\quad F_{sy}\overset{def}{=}\frac{\partial^s F}{\partial y^s},\quad F_{rxsy}\overset{def}{=}\frac{\partial^{r+s} F}{\partial x^r \partial y^s} \qquad /2.4/$$

r,s - rząd pochodnych funkcji $F(x,y)$ względem zmiennych, odpowiednio x i y.

Wprowadźmy również /dla wygody zapisu/ zmienną logiczną D jak poniżej:

$$D=\begin{cases}1 \text{ dla kierunku } \vec{V}^{+} & \text{/dodatniego/}\\ \phi \text{ dla kierunku } \vec{V}^{-} & \text{/ujemnego/}\end{cases} \qquad /2.4a/$$

Łatwo zauważyć, że przy takim określeniu kierunku ruchu wzdłuż krzywej ma miejsce następująca zależność: w trakcie ruchu wzdłuż krzywej, w dodatnim jej kierunku, dodatnie wartości funkcji $F(x,y)$ występują po prawej stronie krzywej, ujemne - po lewej stronie krzywej.

Niech punkty $x_p y_p$ oraz $x_k y_k$ leżą w węzłach siatki: na krzywej lub najbliższym jej sąsiedztwie. Węzły te oraz kierunek ruchu wzdłuż krzywej D jednoznacznie wyznaczają dyskretyzowany segment

krzywej. Należy określić ciąg punktów:

$x_{p+1}y_{p+1}, \ldots x_{p+r}y_{p+r}, \ldots x_{k-1}y_{k-1}$ spełniający warunki nałożone na aproksymację krzywej.

Zdefiniujmy kierunki przemieszczeń wzdłuż głównych linii siatki, odpowiadające przyjętemu w /2.3/ kierunkowi ruchów wzdłuż stycznej do krzywej.

$$x = \begin{cases} +h & \text{gdy zachodzi } (V^x \geq \phi)D \vee (V^x < \phi)\bar{D} = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases} \qquad /2.5/$$

$$x = \begin{cases} +h & \text{gdy zachodzi } (V^y \geq \phi)D \vee (V^y < \phi)\bar{D} = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases}$$

Relacja /2.5/ lokalizuje wektor kierunkowy stycznej z dokładnością do $\frac{\pi}{2}$. Dokładniejszą lokalizację wektora stycznej możemy uzyskać badając nierówności zachodzące pomiędzy pochodnymi cząstkowymi I-rzędu. Lokalizację wektora stycznej z dokładnością $\frac{\pi}{4}$ uzyskamy badając zależności $|F_x| \lesseqgtr |F_y|$, natomiast lokalizację z dokładnością do $\frac{\pi}{6}$ oraz $\frac{\pi}{8}$ - badając zależności

$$2|F_x| \gtrless |F_y| \quad \text{i} \quad |F_x| \lesseqgtr 2|F_y|$$

Chcąc wyznaczyć punkt $x_{p+1}y_{p+1}$ będziemy wykonywać szereg elementarnych ruchów śledzących przebieg krzywej. Będziemy przy tym postępować w następujący sposób. Rozpoczynając od punktu x_py_p będziemy wykonywać dwa typy elementarnych ruchów śledzących położenie krzywej do jednego z ośmiu sąsiednich węzłów: tzw. "ruch od krzywej" oraz "ruchy wzdłuż krzywej".

Kierunki "ruchu od krzywej" oraz "ruchów wzdłuż krzywej" otrzymamy w wyniku lokalizacji wektora stycznej z dokładnością do $\frac{\pi}{8}$. Przeprowadzając lokalizację wektora kierunkowego stycznej otrzymamy 2 różniące się o $\frac{\pi}{4}$ kierunki, z których jeden, bliższy kierunkowi wektora stycznej jest kierunkiem "ruchu wzdłuż krzywej", a drugi - kierunkiem "ruchu od krzywej".

Cykl wyznaczania drugiego punktu aproksymującego $x_{p+1}y_{p+1}$ jest następujący:

1/ z punktu początkowego x_py_p wykonujemy "ruch od krzywej" a w osiągniętym węźle wyznaczamy wartość funkcji $F(x,y)$;

2/ z bieżącego węzła wykonujemy szereg elementarnych "ruchów wzdłuż krzywej" obliczając w każdym z nich wartość funkcji $F(x,y)$ i obserwując, czy nie oddalamy się od krzywej;

3/ operację "ruchów wzdłuż krzywej" kończymy, gdy kolejny "ruch wzdłuż krzywej" przetnie krzywą;

4/ z dwóch sąsiednich węzłów siatki leżących po przeciwnej stronie krzywej na węzeł aproksymujący wybierzemy węzeł bliższy krzywej /w sensie metryki generacyjnej $|F(x,y)|$/;

5/ stwierdzając odbieganie od krzywej modyfikujemy kierunek "ruchu wzdłuż krzywej" o kąt $\pm \frac{\pi}{4}$ w taki sposób, aby po wykonaniu następnego "ruchu wzdłuż krzywej" zbliżyć się do niej lub przeciąć ją;

6/ wyznaczony węzeł aproksymujący jest końcem pierwszego odcinka aproksymującego oraz początkiem drugiego, koniec drugiego odcinka aproksymującego oraz wszystkie następne węzły aproksymujące wyznaczamy sposobem opisanym powyżej.

Sprecyzujemy dokładniej warunki "ruchu od krzywej", warunki "ruchu wzdłuż krzywej", modyfikację kierunku "ruchu wzdłuż krzywej" oraz oddalanie się od krzywej i jej przecięcie.

Wykorzystując zależności między pochodnymi cząstkowymi I-rzędu F_x i F_y i innymi wielkościami

można określić szereg różnych sposobów śledzenia krzywej.

W dalszej części pracy opiszemy dwa takie sposoby, dość istotnie różniące się między sobą /rys. 2.2 i rys. 2.3/.

Sposób I ruchów śledzących przebieg krzywej

Określona kombinacja "ruchu od krzywej" oraz "ruchów wzdłuż krzywej" następuje w zależności od spełnienia się jednego z następujących /rozłącznych/ warunków:

1/ "ruch od krzywej" należy wykonać wzdłuż przekątnej p, a "ruchy wzdłuż krzywej" - w kierunku osi x, gdy spełniony jest poniższy warunek:

$$|F_y| > 2|F_x|$$

a także gdy spełniony jest warunek:

$$|F_y| = 2|F_x| \quad \text{i równocześnie}$$

- pochodne F_x i F_y mają znaki zgodne a obieg krzywej jest ujemny lub gdy
- pochodne te mają znaki przeciwne a obieg krzywej jest dodatni;

2/ "ruch od krzywej" należy wykonywać wzdłuż przekątnej p a "ruchy wzdłuż krzywej" - w kierunku osi y gdy spełniony jest warunek:

$$|F_x| > 2|F_y|$$

a także, gdy spełniony jest warunek:

$$|F_x| = 2|F_y| \quad \text{i równocześnie}$$

- pochodne F_x i F_y mają zgodne znaki a obieg krzywej jest dodatni lub gdy
- pochodne te mają znaki przeciwne a obieg krzywej jest ujemny;

3/ "ruch od krzywej" należy wykonać wzdłuż osi x a "ruchy wzdłuż krzywej" - wzdłuż przekątnej p, gdy spełniony jest warunek:

$$(|F_y| > |F_x|) \wedge (|F_y| < 2|F_x|) = 1$$

a także w dwu następujących przypadkach:

a/ spełniony jest warunek:

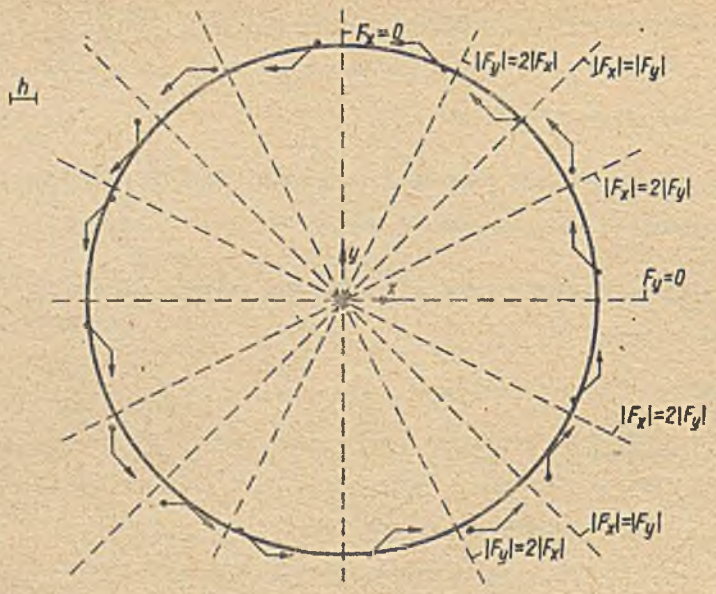
$$|F_y| = |F_x| \quad \text{i równocześnie}$$

- pochodne F_x i F_y mają znaki zgodne przy ujemnym obiegu krzywej lub też gdy
- pochodne F_x i F_y mają znaki przeciwne przy dodatnim kierunku obiegu krzywej,

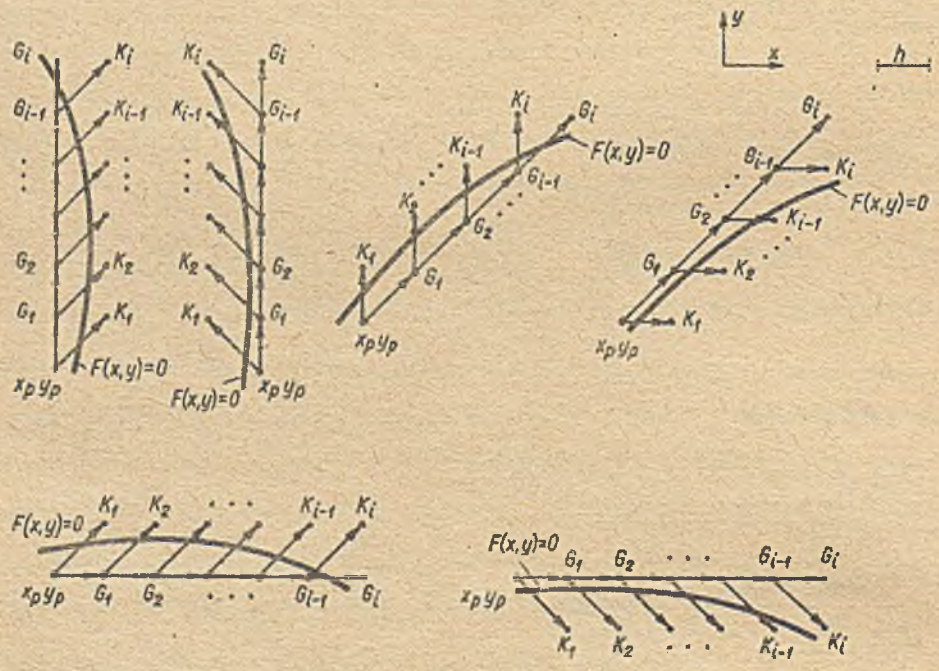
b/ spełniony jest warunek:

$$|F_y| = 2|F_x| \quad \text{i równocześnie}$$

- pochodne F_x i F_y mają znaki zgodne przy dodatnim kierunku obiegu krzywej lub też gdy
- pochodne F_x i F_y mają znaki przeciwne przy ujemnym kierunku obiegu krzywej;



Rys. 2.2. Ilustracja sposobu I ruchów śledzących krzywą



Rys. 2.3. Ilustracja sposobu II ruchów śledzących krzywą

4/ "ruch od krzywej" należy wykonać wzdłuż osi y a "ruchy wzdłuż krzywej" - wzdłuż przekątnej p, gdy spełniony jest warunek

$$(|F_x| < 2\ |F_y|) \wedge (|F_x| > |F_y|) = 1$$

a także w dwóch następujących wypadkach:

a/ spełniony jest warunek

$$|F_x| = 2\ |F_y|$$ i równocześnie

- pochodne F_x i F_y mają znaki zgodne przy ujemnym kierunku obiegu krzywej lub też gdy
- pochodne te mają znaki przeciwne przy dodatnim kierunku obiegu krzywej

b/ spełniony jest warunek:

$$|F_x| = |F_y|$$ a równocześnie

- pochodne F_x i F_y mają znaki zgodne przy dodatnim kierunku obiegu krzywej lub też gdy
- pochodne te mają znaki przeciwne przy ujemnym kierunku obiegu krzywej.

Wprowadźmy następujące oznaczenia:

$$\begin{aligned}
A &= D(\Delta x \neq \Delta y) \vee \bar{D}(\Delta x = \Delta y) \\
E &= |2\ F_x| > |F_y| \\
F &= |2\ F_y| > |F_x| \\
G &= |2\ F_x| < |F_y| \\
H &= |2\ F_y| < |F_x| \\
I &= |F_x| > |F_y| \\
J &= |F_y| > |F_x|
\end{aligned} \qquad /2.6/$$

$$W = \begin{cases} 1 & \text{punkt startowy leży po wypukłej stronie krzywej} \\ \phi & \text{punkt startowy leży po wkięsłej stronie krzywej.} \end{cases}$$

Uwzględniając oznaczenia /2.6/ warunki 1/, 2/, 3/, 4/ zapiszemy w następującej, symbolicznej postaci:

$$\begin{aligned}
&1/\ ER := p, x, x, \ldots \text{ gdy zachodzi } G \vee \overline{GEA} = 1 \\
&2/\ ER := p, y, y, \ldots \text{ gdy zachodzi } H \vee \overline{HFA} = 1 \\
&3/\ ER := y, p, p, p, \ldots \text{ gdy zachodzi } FI \vee \overline{HFA} \vee \overline{IIA} = 1 \\
&4/\ ER: x, p, p, \ldots \text{ gdy zachodzi } JE \vee \overline{IIA} \vee \overline{GEA} = 1
\end{aligned} \qquad /2.7/$$

Modyfikacja kierunku "ruchów wzdłuż krzywej"

Kierunki elementarnych "ruchów wzdłuż krzywej" należy zmodyfikować, gdy w trakcie ich wykonywania stwierdzamy oddalanie się od krzywej a dokładniej - gdy nie zbliżamy się do niej. W przyjętej tu matryce generacyjnej wyraża się to brakiem zmniejszania modułu wartości funkcji w kolejno osiąganych węzłach. Modyfikacja kierunku "ruchu wzdłuż krzywej" uzależniona jest od położenia /w stosunku od krzywej/ osiągniętego węzła, oraz od kierunku obiegu krzywej L, a także od modyfikowanego kierunku.

Modyfikacja kierunku odbywa się przez obrót tego kierunku o kąt $\pm \frac{\pi}{4}$.

Modyfikacja odbywa się o kąt $+\frac{\pi}{4}$, gdy osiągnięty węzeł znajduje się po wypukłej stronie krzywej, a obieg krzywej jest dodatni lub też gdy węzeł ten znajduje się po wklęsłej stronie krzywej, a obieg krzywej jest ujemny.

Modyfikacja odbywa się o kąt $-\frac{\pi}{4}$ gdy osiągnięty węzeł znajduje się po wklęsłej stronie krzywej, a obieg krzywej jest dodatni lub też gdy węzeł ten znajduje się po wypukłej stronie krzywej, a obieg krzywej jest ujemny. Zapiszemy to w następującej postaci symbolicznej:

$$M(ER) : ER^{\nwarrow} + \frac{\pi}{4} \quad \text{gdy zachodzi } DW \vee \bar{D}\bar{W} = 1$$
$$M(ER) : = ER^{\nearrow} - \frac{\pi}{4} \quad \text{gdy zachodzi } \bar{D}W \vee D\bar{W} = 1 \qquad /2.9/$$

Składowe $\Delta x'$, $\Delta y'$ zmodyfikowanego wektora "ruchu wzdłuż krzywej" otrzymujemy ze składowych Δx, Δy dotychczasowego wektora w wyniku następujących przekształceń:

1/ przy obrocie o $+\frac{\pi}{4}$:

- w przypadku modyfikacji kierunku ruchu wzdłuż przekątnej:

a/ jeżeli zachodzi $\Delta x = \Delta y$ to należy przyjąć:

$$\Delta x' := \phi$$
$$\Delta y' := \Delta y$$

b/ jeżeli zachodzi $\Delta x = -\Delta y$ to należy przyjąć:

$$\Delta x' := \Delta x$$
$$\Delta y' := \phi \qquad /2.10a/$$

- w przypadku modyfikacji kierunku ruchu wzdłuż osi x należy przyjąć:

$$\Delta x' := \Delta x$$
$$\Delta y' := \Delta x$$

- w przypadku modyfikacji kierunku ruchu wzdłuż osi y należy przyjąć:

$$\Delta x' := -\Delta y$$
$$\Delta y' := \Delta y$$

2/ przy obrocie o kąt $-\frac{\pi}{4}$:

- w przypadku modyfikacji kierunku ruchu wzdłuż przekątnej:

a/ jeżeli zachodzi $\Delta x = \Delta y$ to należy przyjąć:

$$\Delta x' := \Delta x$$
$$\Delta y' := \phi \qquad /2.10b/$$

b/ jeżeli zachodzi $\Delta x = -\Delta y$ to należy przyjąć:

$$\Delta x' := \phi$$

$$\Delta y' := \Delta y$$

- w przypadku modyfikacji kierunku ruchu wzdłuż osi x należy przyjąć:

$$\Delta x' := \Delta x$$

$$\Delta y' := -\Delta x$$

- w przypadku modyfikacji kierunku ruchu wzdłuż osi y należy przyjąć:

$$\Delta x' := \Delta y$$

$$\Delta y' := \Delta y$$

Przecięcie krzywej /przejście na jej drugą stronę/ powoduje zmianę znaku wartości $F(x,y)$, stąd też wymaga się jedynie sprawdzenia znaków jej wartości w kolejno osiąganych węzłach.

Warunki wybierające jeden z dwóch węzłów /leżących po przeciwnych stronach krzywej/ na węzeł aproksymujący /tj. na węzeł będący końcem odcinka łamanej/, obejmujące przypadek graniczny równości modułów funkcji w węzłach po przeciwnych stronach krzywej, zapisujemy w postaci następującej relacji:

$x_{k_l} y_{k_l} := x_l y_l$ gdy jest spełnione

$$\left[\left(|F_{l+1}| > |F_l|\right) \vee \left(|F_{l+1}| = |F_l|\right) \; (F_l \geq \phi)\right] \quad (\operatorname{Sgn} F_{l+1} \neq \operatorname{Sgn} F_l) = 1$$

$x_{k_l} y_{k_l} := x_{l+1} y_{l+1}$ gdy jest spełnione /2.11/

$$\left[\left(|F_{l+1}| < |F_l|\right) \vee \left(|F_{l+1}| = |F_l|\right) \; (F_l < \phi)\right] (\operatorname{Sgn} F_l \neq \operatorname{Sgn} F_{l+1}) = 1$$

Sposób II ruchów śledzących położenia krzywej

Sformułujemy drugi, odmienny wariant ruchów śledzących położenie krzywej, w którym moduł wartości funkcji $|F(x,y)|$ /metryka generacyjna/ spełnia jedynie rolę pomocniczą, a oddalanie się ruchu śledzącego od krzywej nie powoduje zmiany jego kierunku /modyfikacji kierunku ruchu/, jeżeli odchylenie to nie przekracza określonej wartości odległości - jeżeli krzywa nie wychodzi z pewnego, określonego pasma tolerancji /o szerokości h lub $h\sqrt{2}/2$ w zależności od kierunku nachylenia stycznej do krzywej/. Zgodnie z tym wariantem śledzenie krzywej odbywa się za pomocą dwóch typów ruchów śledzących: tzw. "ruchu głównego" oraz "ruchu kontrolnego". Kierunki "ruchu głównego" i "ruchu kontrolnego" różnią się od siebie o $\frac{\pi}{4}$ i wyznaczają dwa sąsiednie węzły G i K leżące po przeciwnych stronach krzywej.

W trakcie śledzenia krzywej dokonywane są elementarne przemieszczenia wzdłuż kierunku "ruchu głównego" oraz lokalne ruchy kontrolne wzdłuż kierunku "ruchu kontrolnego". Śledzenie krzywej /elementarne ruchy wzdłuż krzywej/ następuje dopóty, dopóki dwa węzły: G_l - osiągnięty za pomocą "ruchu głównego" oraz K_l - osiągnięty za pomocą "ruchu kontrolnego" leżą po przeciwnych stronach śledzonej krzywej.

Kolejne czynności doprowadzające do wyznaczenia węzła x_{p+1}, y_{p+1} aproksymującego krzywą są następujące:

1/ określenie kierunku "ruchu głównego" : kierunku wzdłuż osi x lub y lub przekątnej p tego, który z wektorem kierunkowym stycznej, wystawionym w punkcie początkowym x_p, y_p tworzy mniejszy kąt;

2/ określenie kierunku "ruchu kontrolnego" /różniącego się od kierunku ruchu głównego o $\pm \frac{\pi}{4}$/ wskazującego węzeł sąsiedni K leżący po przeciwnej stronie krzywej niż węzeł G, wskazany kierunkiem "ruchu głównego";

3/ wykonywanie kolejnych elementarnych ruchów wzdłuż kierunku "ruchu głównego" /osiąganie kolejnych węzłów G_l/ oraz lokalnych ruchów wzdłuż kierunku "ruchu kontrolnego" /osiąganie kolejnych węzłów K_l/ i badanie położenia tych węzłów w stosunku do krzywej /badanie znaków wartości funkcji $F(x,y)$ w tych węzłach/, aż do stanu, gdy para węzłów G_l, K_l znajdą się po tej samej stronie krzywej;

4/ wybranie węzła aproksymującego x_{p+1}, y_{p+1} - jednego z węzłów G_l, K_l, G_{l-1}, K_{l-1} według następującej zasady:

- gdy węzły G_l oraz G_{l-1} leżą po przeciwnych stronach krzywej, to na węzeł aproksymujący wybrany zostaje ten z nich, który jest bliższy krzywej w sensie min $|F(x,y)|$,
- gdy węzeł G_l oraz G_{l-1} leżą po tej samej stronie krzywej to na węzeł aproksymujący wybrany zostaje ten z węzłów K_l lub K_{l-1}, który jest bliższy krzywej w sensie min $|F(x,y)|$;

5/ kolejne punkty x_{p+2}, y_{p+2} i dalsze, aproksymujące krzywą wyznacza się identycznie jak punkt x_{p+1}, y_{p+1}.

Ilustracje do tego wariantu śledzenia położenia krzywej pokazano na rys. 2.3.

Omówimy dokładniej poszczególne operacje oraz zależności warunkujące ruchy śledzące przebieg krzywej.

Określenie kierunku "ruchu głównego" warunkowane jest poniższymi zależnościami:

- jeżeli zachodzi nierówność

$$|F_x| > 2|F_y| \qquad /2.12a/$$

to kierunkiem "ruchu głównego" jest kierunek osi y,

- jeżeli zachodni nierówność

$$|F_y| > 2|F_x| \qquad /2.12b/$$

to kierunkiem "ruchu głównego" jest kierunek osi x,

- jeżeli nie zachodzi żadna z powyższych relacji

to kierunkiem "ruchu głównego" jest kierunek przekątnej p.

Określenie kierunku "ruchu kontrolnego" uwarunkowane jest następującymi zależnościami:

1/ dla kierunku "ruchu głównego" zgodnego z kierunkiem osi x kierunkiem "ruchu kontrolnego" jest kierunek przekątnej, różniący się od kierunku głównego o kąt $\pm \frac{\pi}{4}$ to znaczy ten, który doprowadza do węzła K leżącego po przeciwnej stronie krzywej w stosunku do węzła G, wyznaczonego przez kierunek "ruchu głównego"; zapisujemy to w następującej postaci symbolicznej:

- jeżeli zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^X(G) \neq \mathrm{Sgn}\, F^P(K) \qquad /2.13a/$$

wtedy kierunkiem "ruchu kontrolnego" jest kierunek przekątnej, określony przez kierunki elementarnych

ruchów wzdłuż głównych linii siatki /relacji /2.5//,

- w przeciwnym przypadku przyjmij

$\Delta y' := -\Delta y$

i oblicz $F^{p'}(K')$ w punkcie K' wyznaczonym przez nowy kierunek przekątnej; jeżeli wówczas zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{x}(G) \neq \mathrm{Sgn}\, F^{p'}(K') \qquad /2.13b/$$

to kierunkiem "ruchu kontrolnego" jest ostatnio określony kierunek przekątnej,

- natomiast gdy zachodzi zgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{x}(G) = \mathrm{Sgn}\, F^{p'}(K') = \mathrm{Sgn}\, F^{p}(K) \qquad /2.13c/$$

to przerwij wyznaczanie kierunku ruchu kontrolnego, wybierz na węzeł aproksymujący x_{p+1}, y_{p+1} jeden z węzłów K lub K' wyznaczonych przez ostatni i poprzedni kierunki przekątnych zgodnie z zasadą $\min|F(x,y)|$ w tych węzłach i przystąp do wyznaczania kolejnego węzła aproksymującego;

2/ dla kierunku "ruchu głównego" wzdłuż osi y kierunkiem "ruchu kontrolnego" jest również /jeden/ kierunek przekątnej, różniący się od kierunku "ruchu głównego" o $\pm \frac{\pi}{4}$, ten, który doprowadza do węzła K, leżącego po przeciwnej stronie krzywej niż węzeł G wyznaczony przez kierunek "ruchu głównego"; zapiszemy w następującej postaci:

- jeżeli zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{y}(G) \neq \mathrm{Sgn}\, F^{p}(K) \qquad /2.14a/$$

wtedy kierunkiem "ruchu kontrolnego" jest kierunek przekątnej, określony przez kierunki elementarnych ruchów wzdłuż głównych linii siatki /relacji /2.5//,

- w przeciwnym przypadku przyjmij

$\Delta x' := -\Delta x$

i oblicz $F^{p'}(K')$ w punkcie K' wyznaczonym przez nowy kierunek przekątnej, jeżeli wówczas zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{x}(G) \neq \mathrm{Sgn}\, F^{p'}(K') \qquad /2.14b/$$

to kierunkiem "ruchu kontrolnego" jest ostatnio określony kierunek przekątnej,

- jeżeli natomiast w przypadku równości znaków wartości funkcji w trzech węzłach

$$\mathrm{Sgn}\, F^{y}(G) = \mathrm{Sgn}\, F^{p'}(K') = \mathrm{Sgn}\, F^{p}(K) \qquad /2.14c/$$

przerwij wyznaczanie kierunku "ruchu kontrolnego", wybierz na węzeł aproksymujący (x_{p+1}, y_{p+1}) jeden z węzłów K lub K' wyznaczonych przez rozważane kierunki "ruchu kontrolnego", zgodnie z zasadą $\min|F(x,y)|$ w tych węzłach i przystąp do wyznaczania kolejnego węzła aproksymującego;

3/ dla kierunku "ruchu głównego" wzdłuż przekątnej p kierunkiem "ruchu kontrolnego" jest ten z kierunków wzdłuż osi x bądź y elementarnych ruchów /relacji /2.5//, który doprowadza do węzła K leżącego po przeciwnej stronie krzywej niż węzeł G, wyznaczony przez kierunek "ruchu głównego"; zapisujemy to w następującej postaci:

- jeżeli zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{p}(G) \neq \mathrm{Sgn}\, F^{x}(K) \qquad /2.15a/$$

wtedy kierunkiem "ruchu kontrolnego" jest kierunek osi x,

- w przeciwnym przypadku, jeżeli zachodzi niezgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{p}(G) \neq \mathrm{Sgn}\, F^{y}(K'), \qquad /2.15b/$$

to kierunkiem "ruchu kontrolnego" jest kierunek osi y,

- jeżeli natomiast zachodzi zgodność znaków wartości funkcji

$$\mathrm{Sgn}\, F^{p}(G) = \mathrm{Sgn}\, F^{x}(K) = \mathrm{Sgn}\, F^{y}(K') \qquad /2.15c/$$

to przerwij wyznaczanie kierunku "ruchu kontrolnego", wybierz na węzeł aproksymujący (x_{p+1}, y_{p+1}) jeden z węzłów K lub K', wyznaczonych przez kierunki osi x oraz y zgodnie z zasadą $\min|F(x,y)|$ w tych węzłach i przystąp do wyznaczania kolejnego węzła aproksymującego.

Wykonanie kolejnych elementarnych ruchów wzdłuż kierunku "ruchu głównego" polega na przemieszczaniu się do kolejnych węzłów G_i wyznaczanych przez kierunek "ruchu głównego" i obliczaniu w tych węzłach wartości funkcji $F_i(x,y)$ i na obliczaniu równolegle wartości funkcji $F_i(x,y)$ w węzłach K_i wyznaczanych przez kierunek "ruchu kontrolnego" oraz na badaniu /tylko/ niezgodności znaków wartości funkcji $F(G_{i-1})$ i $F(G_i)$ oraz $F(K_{i-1})$ i $F(K_i)$ w odpowiadających sobie węzłach G_{i-1} i G_i oraz K_{i-1} i K_i. Ruchy śledzące położenie krzywej zostają zaniechane wówczas, gdy nastąpi niezgodność znaków wartości funkcji chociażby w jednej z par węzłów (G_{i-1}, G_i) oraz $K_{i-1}, K_i)$. Na tym etapie śledzenia położenia krzywej zostaje określony kolejny węzeł aproksymujący.

Wybranie węzła na węzeł aproksymujący odbywa się podobnie jak w przypadku poprzednich wariantów ruchów śledzących położenie krzywej, a uwarunkowane jest następującymi zależnościami:

a/ jeżeli węzły G_{i-1} oraz G_i leżą po przeciwnych stronach krzywej, to na węzeł aproksymujący zostaje wybrany jeden zgodnie z zasadą $\min|F(x,y)|$; zapiszemy to w następującej symbolicznej postaci:

$$(x_{k_i}, y_{k_i}) := G_{i-1}$$

gdy jest spełnione

$$\left[(|F(G_i)| > |F(G_{i-1})|) \vee (|F(G_i)| = |F(G_{i-1})|)\ (F(G_{i-1}) \geq \phi)\right]\ (\mathrm{Sgn}\, F(G_i) \neq \mathrm{Sgn}\, F(G_{i-1})) = 1,$$

$$(x_{k_i}, y_{k_i}) := G_i \qquad /2.16a/$$

gdy jest spełnione

$$\left[(|F(G_i)| < |F(G_{i-1})|) \vee (|F(G_i)| = |F(G_{i-1})|)\ (F)G_{i-1}) < \phi)\right]\ (\mathrm{Sgn}\, F(G_i) \neq \mathrm{Sgn}\, F(G_{i-1})) = 1$$

b/ jeżeli węzły G_{i-1} oraz G_i leżą po tej samej stronie krzywej to na węzeł aproksymujący zostaje wybrany jeden z węzłów K_{i-1} lub K_i, który jest bliższy krzywej w sensie $\min|F(x,y)|$ a wyznaczony został za pomocą ostatnich dwóch "ruchów kontrolnych", zapiszemy to w następującej postaci symbolicznej:

$$(x_{k_i}, y_{k_i}) := K_{i-1}$$

gdy jest spełnione

$$\left[(|F(K_i)| > |F(K_{i-1})|) \vee (|F(K_i)| = |F(K_{i-1})|)\ (F(K_{i-1}) \geq \phi)\right]\ (\mathrm{Sgn}\, F(G_i) = \mathrm{Sgn}\, F(G_{i-1}) = 1 \qquad /2.16b/$$

$$(x_{k_i}, y_{k_i}) := K_i$$

gdy jest spełnione

$$\left[(|F(K_i)| < |F(K_{i-1})|) \vee (|F(K_i)| = |F(K_{i-1})|)\ (F(K_{i-1}) < \phi)\right]\ (\mathrm{Sgn}\, F(G_i) = \mathrm{Sgn}\, F(G_{i-1})) = 1$$

Obliczanie wartości funkcji F(x,y)

Wartość funkcji F(x,y) w węzłach osiąganych ruchami śledzącymi położenie krzywej można wyznaczać różnymi sposobami. Jednym z nich jest wyznaczenie F(x,y) wprost z równania /2.1/. Sposób ten wymaga jednak dla każdego z węzłów wykonania dużej liczby operacji potęgowania. Z tego powodu jest mało efektywny. Podamy więc sposób inny, wydajniejszy z punktu widzenia złożoności obliczeniowej, wykorzystujący rozwinięcie funkcji F(x,y) w szereg Taylora.

Wprowadźmy operatory kierunkowe funkcji F(x,y):

$$L^{\bar{x}}F(x_1,y_1) \overset{df}{=} F(x_1+\Delta x_1,y_1) - F(x_1,y_1)$$

$$L^{y}F(x_1,y_1) \overset{df}{=} F(x_1,y_1+\Delta y_1) - F(x_1,y_1) \qquad /2.17/$$

gdzie $|\Delta x_1| = |\Delta y_1| = h$, h - moduł siatki

Korzystając z rozwinięcia w szereg Taylora funkcji F(x,y) w otoczeniu punktu x,y mającego postać:

$$F(x+\Delta x,y+\Delta y) = \sum_{l=0}^{n} \frac{1}{l!} \left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y}\right)^{l} F(x,y)$$

/n- stopień krzywej/ i uwzględniając oznaczenia /2.4/ otrzymamy poniższe wzory:

$$L^{x}F(x,y) = \sum_{l=1}^{n} \frac{(\Delta x)^{l}}{l!} F_{lx}(x,y) \qquad /2.18/$$

$$L^{y}F(x,y) = \sum_{l=1}^{n} \frac{(\Delta y)^{l}}{l!} F_{ly}(x,y)$$

$$L^{p}F(x,y) = \sum_{k=1}^{n} \sum_{l=0}^{k} \frac{(\Delta x)^{k-l}}{(k-l)!} \frac{(\Delta y)^{l}}{l!} F_{(k-l)xly}(x,y)$$

oraz zależność:

$$L^{p}F(x,y) = L^{x}F(x,y) + L^{y}F(x,y) + L^{R}F(x,y) \qquad /2.19/$$

gdzie wyrażenie:

$$L^{R}F(x,y) = \sum_{k=1}^{n-1} \sum_{l=1}^{k} \frac{(\Delta x)^{k-l+1}}{(k-l+1)!} \frac{(\Delta y)^{l}}{l!} F_{(k-l+1)xly}(x,y)$$

jest prostsze do wyznaczenia od $L^{p}F(x,y)$ na podstawie wzoru /2.18/. Stąd na podstawie wyrażeń /2.17/ można wyznaczyć wartości funkcji F(x,y) w punktach $(x_1+\Delta x_1,y_1)$, $(x_1,y_1+\Delta y_1)$ oraz $(x_1+\Delta x_1+\Delta y_1)$ pod warunkiem, że znane są pochodne cząstkowe rzędów 1 - n w punkcie x_1,y_1. Te ostatnie można wyznaczyć w podobny sposób, korzystając z rozwinięcia w szereg Taylora pochodnych cząstkowych w punkcie poprzednim oraz z poniższych wzorów, analogicznych do wzorów /2.17/, /2.18/ i /2.19/:

$$L^{x}F_{rxsy}(x_{l-1},y_{l-1}) \overset{df}{=} F_{rxsy}(x_{l-1},y_{l-1}+\Delta y_{l-1}) - F_{rxsy}(x_{l-1},y_{l-1})$$

$$L^{p}F_{rxsy}(x_{l-1},y_{l-1}) \overset{df}{=} F_{rxsy}(x_{l-1}+\Delta x_{l-1},y_{l-1}+\Delta y_{l-1}) - F_{rxsy}(x_{l-1},y_{l-1})$$

oraz

$$L^{x}F_{rxsy} = \sum_{k=1}^{n-r-s} F_{(r+k)xsy} \frac{(\Delta x)^{k}}{k!}, \qquad /2.20/$$

$$L^{y}F_{rxsy} = \sum_{k=1}^{n-r-s} F_{rx(s+k)y} \frac{(\Delta y)^{k}}{k!},$$

$$L^{p}F_{rxsy} = \sum_{k=1}^{n-r-s} \sum_{l=0}^{k} F_{(r+k-l)x(s+l)y} \frac{(\Delta x)^{k-1}}{(k-l)!} \frac{(\Delta y)^{k}}{l!}$$

dla $1 \leqslant r+s \leqslant n-1, \quad \phi \leqslant r,s \leqslant n-1.$

Wartości pochodnych cząstkowych w punkcie x_p,y_p wyznacza się, gdy znane są współczynniki równania uwikłanego krzywej $a_{i,j}$ $(i,j=\phi-n)$, ze wzoru następującego:

$$F_{rxsy} = \sum_{k=\phi}^{n-r-s} \sum_{l=s}^{n-r-k} \frac{(m-k-l)!}{(n-k-l-r)!} \frac{!!}{(l-s)!} a_{n-k-l,l} x^{n-k-l-r} y^{l-s} \qquad /2.21/$$

dla $1 \leqslant r+s \leqslant 1, \quad \phi \leqslant r,s \leqslant n.$

Ostatnie wzory sprowadzić można do poniższej postaci, wygodniejszej w komputerowej realizacji algorytmu. Dla pochodnych względem jednej zmiennej.

$$F_{rx} = \sum_{k=\phi}^{n-1} \sum_{l=\phi}^{n-r-k} \prod_{p=n-k-r-l+1}^{n-k-l} (p) \cdot a_{n-k-l,l} \cdot x^{n-k-r-l} \cdot y^{l}$$

$$F_{sy} = \sum_{k=\phi}^{n-s} \sum_{l=s}^{n-k} \prod_{q=l-s+1}^{l} (q)\, a_{n-k-l,l}\, x^{n-k-l} y^{l-s} \qquad /2.22/$$

gdzie $1 \leqslant r,s \leqslant n,$

dla pochodnych mieszanych

$$F_{rxsy} = \sum_{k=\phi}^{n-r-s} \sum_{l=s}^{n-k-r} \prod_{p=n-k-r-l+1}^{n-k-l} (p) \prod_{q=l-s+1}^{l} (q)\, a_{n-k-l,l}\, x^{n-k-r-l} y^{l-s}$$

gdzie $1 \leqslant r,s \leqslant n, \qquad 2 \leqslant r+s \leqslant n.$

2.3. Dane początkowe dyskretyzacji

Rozważane metody dyskretyzacji wymagają, jako danych początkowych określenia współrzędnych punktu początkowego x_p,y_p i końcowego x_k,y_k generowanego segmentu krzywej, wartości funkcji w punkcie początkowym $F(x_p,y_p)$, kierunku obiegu krzywej D oraz wartości pochodnych cząstkowych rzędów $1 \div n$ w punkcie początkowym. Punkty: początkowy i końcowy, a szczególnie początkowy powinny bezwzględnie pokrywać się z węzłami siatki /powinny mieć współrzędne całkowitoliczbowe/. Alternatywnym sposobem zadania krzywej, wygodniejszym z punktu widzenia komputerowej realizacji algorytmów jest podanie współczynników $a_{i,j}$ równania /2.1/ krzywej w miejsce pochodnych cząstkowych i wartości funkcji $F(x_p,y_p)$

w punkcie początkowym. Wartość $F(x_p,y_p)$ /metrykę generacyjną/ niezbędną do wyznaczania ciągu aproksymującego można wtedy wyznaczyć wprost z równania /2.2/, natomiast wartości pochodnych cząstkowych w węźle początkowym - z równań /2.18/ ewentualnie z /2.21/.

Węzły: początkowy $x_p y_p$ i końcowy $x_k y_k$ /a szczególnie ten ostatni/ powinny leżeć możliwie najbliżej dyskretyzowanego segmentu krzywej, w sensie minimalnej odległości /od krzywej/ mierzonej w metryce generacyjnej tj. $|F(x,y)|$.

Niezbędny w lokalizacji wektora stycznej kierunek obiegu krzywej D zastąpić można podając $\Delta x_p, \Delta y_p$ - elementarne przyrosty wzdłuż głównych linii siatkowych, z węzła początkowego $x_p y_p$ do węzła sąsiedniego, leżącego możliwie najbliżej krzywej zadanej. Na podstawie pierwszego elementarnego ruchu można bowiem określić kierunek obiegu krzywej D. Ten sposób zadania kierunku ruchu wzdłuż krzywej wydaje się być jednak mniej wygodny, ze względu na konieczność wyznaczania /poza algorytmem generacyjnym/ wartości funkcji w co najmniej dwóch sąsiednich węzłach siatki. Długość dyskretyzowanego segmentu krzywej może również być zadana kilkoma sposobami, np. przez określanie $\Delta x, \Delta y$ - zmian współrzędnych między punktem początkowym i końcowym segmentu lub jeszcze inny sposób. W przypadku, gdy dyskretyzowany segment krzywej określony jest również za pomocą punktu $x_k y_k$, to warunek końca dyskretyzacji ma następującą postać:

$$\left(|x_l - x_k| < h\right)\left(|y_l - y_k| \leqslant h\right) \vee \left(|x_l - x_k| \leqslant h\right)\left(|y_l - y_k| < h\right) = 1 \qquad /2.23/$$

2.4. Izolowanie punktów osobliwych krzywej i uwarunkowania wyboru początkowego i końcowego punktów dyskretyzowanego segmentu krzywej

Niezdegenerowane postaci krzywych n-stopnia, dla $n > 2$ mogą zawierać punkty osobliwe, w szczególności punkty rozgałęzienia, ostrza krzywej, punkty samostyczności lub punkty kątowe. W punktach tych nie jest spełniony warunek regularności krzywej tj. skończoność pochodnych cząstkowych I-rzędu oraz zależność $F_x^2 + F_y^2 > \phi$. Z tego też względu w punkcie osobliwym nie można przeprowadzić lokalizacji nachylenia wektora kierunkowego stycznej do krzywej. A tym samym nie można określić kolejnego punktu ciągu aproksymującego krzywą. Punkty osobliwe trzeba więc izolować.

Z dokładniejszej analizy tego problemu wynika, że na dyskretyzację krzywych zawierających punkty osobliwe oraz leżące blisko siebie różne gałęzie krzywej /opisane jednym równaniem /2.1// należy założyć następujące ograniczenia:

1/ dyskretyzowany segment krzywej nie może zawierać punktów osobliwych pomiędzy punktami: początkowym $x_p y_p$ i końcowym $x_k y_k$ /ze względu na niemożliwość lokalizacji na nim wektora kierunkowego stycznej/;

2/ punkt początkowy $x_p y_p$ nie może być punktem osobliwym /ze względu na niemożliwość lokalizacji wektora stycznej w tym punkcie/;

3/ na punkt początkowy nie można wybierać węzłów siatki leżących w bezpośredniej bliskości przylegającej gałęzi krzywej /na odległość $d < h$/, tzn. w tych rejonach krzywej, w których przez jedno elementarne pole siatki przechodzi więcej niż jedna gałąź krzywej; rozpoczynając z takiego węzła dyskretyzację otrzymuje się łamaną z defektem, z dużymi zniekształceniami lub też może zostać wygenerowana niewłaściwa gałąź krzywej.

4/ na punkt końcowy segmentu $x_k y_k$ nie można /podobnie jak w przypadku punktu $x_p y_p$/ wybierać punktów w tych rejonach krzywej, w których leżą blisko siebie różne gałęzie krzywej /tzn. gdzie przez jedno elementarne pole siatki przechodzi więcej niż jedna gałąź krzywej/; w trakcie generacji może nastąpić bowiem przejście na przylegającą gałąź krzywej przed osiągnięciem węzła $x_k y_k$.

Punkty końcowe generowanych segmentów krzywych mogą natomiast być punktami osobliwymi. Podstawowym warunkiem wyboru punktu $x_k y_k$ jest , by wybrany był on w takim miejscu krzywej, w którym inne gałęzie krzywej są dostatecznie odległe /co najmniej na odległość równą modułowi siatki h/.

3. ALGORYTMY DYSKRETYZACJI

Na bazie przedstawionych wyżej dwóch różnych wariantów ruchów śledzących algebraiczne krzywe płaskie n-stopnia - skonstruujemy dwa algorytmy dyskretyzacji tych krzywych na jednorodnej kwadratowej siatce k-spójnej.

Chociaż każdy z algorytmów jest oparty na odmiennej koncepcji śledzenia przebiegu krzywej, to tym niemniej mają one również pewne cechy wspólne. Obydwa algorytmy mają po jednej pętli głównej i po kilka pętli małych. Przejście przez pętlę główną doprowadza zawsze do wyznaczenia jednego /kolejnego/ punktu aproksymującego. W pętlach małych są wykonywane ruchy śledzące przebieg krzywej.

W pierwszym algorytmie są trzy małe pętle, a wejście do każdej z nich wymaga spełnienia odmiennych warunków.

W algorytmie II ruchy śledzące przebieg krzywej wykonywane są w czterech małych pętlach. Nie ma również wzajemnych sprzężeń między tymi pętlami. Wariant ruchów śledzących krzywą, na którym oparty jest algorytm II nie zawiera możliwości modyfikacji kierunku ruchów śledzących.

Operacje początkowe w obydwu algorytmach są jednakowe. Są to operacje zapamiętania danych początkowych, wyznaczenia pochodnych cząstkowych rzędów 1÷n funkcji $F(x,y)$ w punkcie początkowym oraz operacje wyznaczenia kierunków elementarnych ruchów wzdłuż głównych linii siatki /lokalizacja położenia wektora stycznej z dokładnością do $\frac{\pi}{2}$ /.

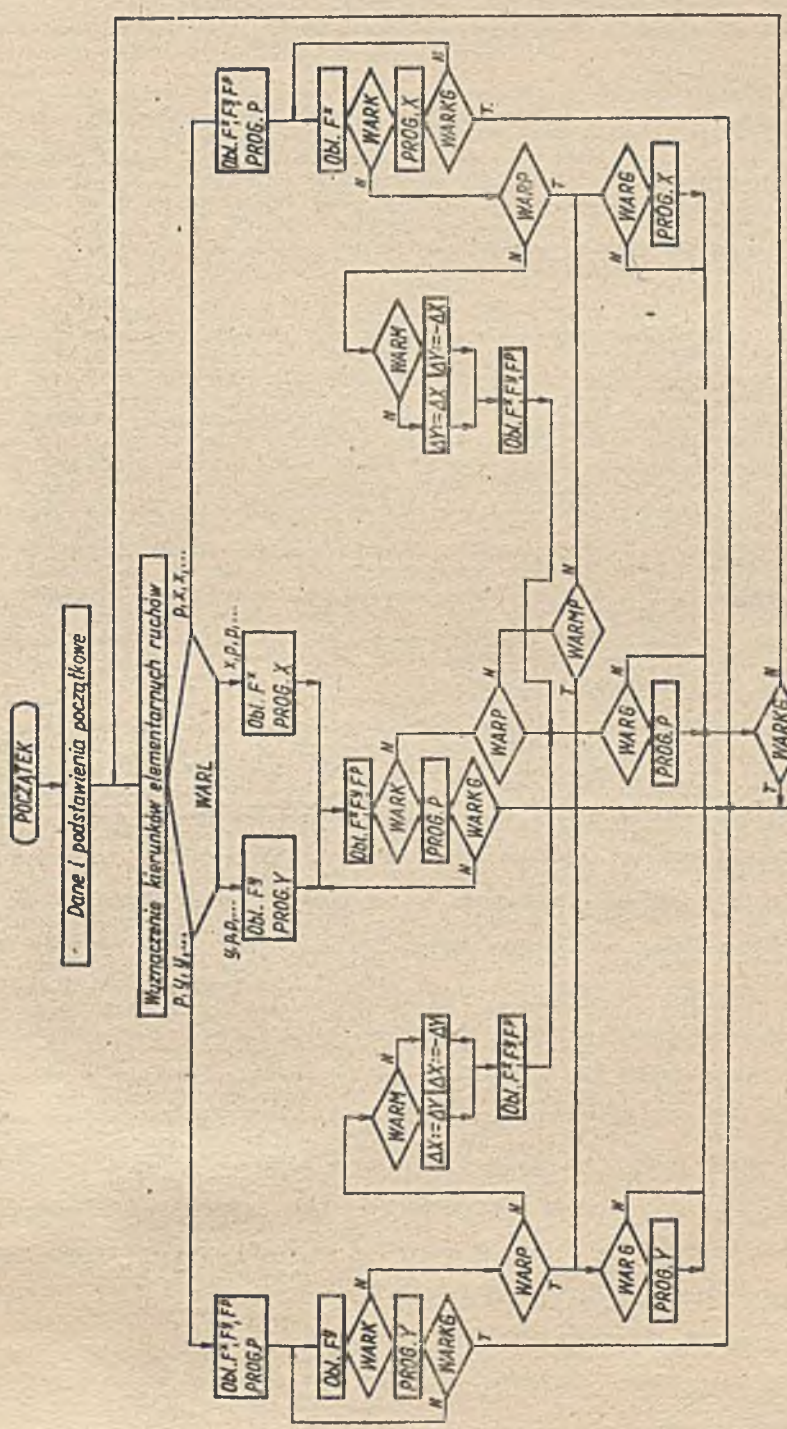
Dalsze operacje są już różne dla poszczególnych algorytmów.

Algorytm I

Algorytm I w postaci blokowej został przedstawiony na rys. 3.1.

Pierwszą operacją algorytmu, jak już wspomniano, jest zapamiętanie danych początkowych, będących współrzędnymi punktów: początkowego (x_p, y_p) i końcowego (x_k, y_k), współczynników a_{ij}, równania /2.1/ opisującego krzywą oraz kierunku obiegu krzywej D. Następnie wyznaczane są pochodne cząstkowe funkcji $F(x,y)$ w punkcie początkowym /wzory 2.21/ Dalsze operacje algorytmu, w odróżnieniu od dotychczasowych, będą wykonywane w algorytmie wielokrotnie - w pętli głównej i w jednej z trzech jego małych pętli.

Pierwszą operacją pętli /głównej/ algorytmu jest operacja wyznaczania kierunków elementarnych ruchów wzdłuż głównych linii siatki /lokalizacja kierunku wektora stycznej z dokładnością do $\frac{\pi}{2}$ - wzory /2.5//. Dalsze operacje algorytmu to wyznaczenie kierunków ruchów śledzących przebieg krzywej:



Rys. 3.1. Algorytm I dyskretyzacji krzywych algebraicznych płaskich na jednorodnej kwadratowej siatce k-spójnej

"ruchu od krzywej" i sprzężonych z nim "ruchów wzdłuż krzywej" /oznaczone w algorytmie jako WARL/, które przebiegają zgodnie z opisanym wcześniej wariantem I tych ruchów /wzory /2.7//. Występują tu różne kombinacje kierunków "ruchu od krzywej" oraz "ruchów wzdłuż krzywej". Relacje /2.7/ lokalizując nachylenie wektora stycznej z dokładnością do $\frac{\pi}{8}$ wybierają jedną z tych kombinacji. Uwzględniają przy tym takie zależności typu $>$, $<$ między modułami pochodnych cząstkowych F_x i F_y jak również i znaki tychże pochodnych oraz kierunek obiegu krzywej D.

Dalej algorytm rozwidla się na trzy gałęzie /będące wejściami do trzech jego małych pętli/, w każdej z nich realizowane są "ruch od krzywej" oraz "ruchy wzdłuż krzywej" - wzdłuż jednego z kierunków: osi x, osi y lub wzdłuż przekątnej układu p. Po wykonaniu "ruchu od krzywej" algorytm aktualizuje wartość F(x,y) oraz wartości pochodnych cząstkowych rzędów 1 - n w osiągniętym węźle /wzory /2.17/ i /2.20//.

Każdy elementarny "ruch wzdłuż krzywej" rozpoczyna się operacją obliczania wartości funkcji w punkcie sąsiednim, wyznaczonym przez określony wcześniej kierunek tego ruchu. W tym algorytmie moduł wartości funkcji jest metryką generacyjną, a porównanie modułów wartości funkcji w punkcie bieżącym oraz w punkcie sąsiednim, wyznaczonym przez kierunek "ruchu wzdłuż krzywej", określa, czy w wyniku wykonania zamierzonego "ruchu wzdłuż krzywej" nie nastąpi oddalenie się od niej. W algorytmie warunek ten oznaczony jest symbolem WARK /warunek kontynuacji "ruchu wzdłuż krzywej"/.

Określa on, czy kolejny krok w "ruchu wzdłuż krzywej" nie spowoduje oddalenia się od niej. W tym celu porównuje się moduły wartości funkcji w punkcie bieżącym i w punkcie sąsiednim, osiągalnym w wyniku wykonania elementarnego "ruchu wzdłuż krzywej".

Gdy z tych dwóch modułów większy jest w punkcie bieżącym, a oba punkty leżą po tej samej stronie krzywej /WARK=TRUE/, to wykonany zostanie "ruch wzdłuż krzywej"; o ile nie został napotkany końcowy punkt krzywej /WARKG=FALSE/, "ruch wzdłuż krzywej" jest kontynuowany /w małej pętli algorytmu/.

Jeśli "ruch wzdłuż krzywej" nie może być kontynuowany /WARK=FALSE/, testowany jest warunek przecięcia krzywej /WARP/, i jeśli takie przecięcie nie nastąpiło /WARP=FALSE/, to jest wykonywana modyfikacja kierunku ruchu /uzależniona od kierunku obiegu krzywej D za pomocą warunku WARM /WARMP//, /wzory /2.8/ - /2.10//. Natomiast gdy nastąpiło przecięcie krzywej /WARP=TRUE/, bądź gdy została wykonana modyfikacja ruchu, to węzeł aproksymujący zostaje w jednym z dwóch rozważanych węzłów, w którym moduł wartości funkcji /relacje /2.11// jest mniejszy. Wybór ten zostaje dokonany w wyniku testowania warunku granicznego WARG. Jeśli zachodzi WARG=TRUE, to węzłem aproksymującym zostaje wybrany węzeł sąsiedni i wtedy w tym punkcie następuje wyznaczenie wartości funkcji i pochodnych cząstkowych. Gdy zachodzi WARG=FALSE, to węzłem aproksymującym zostaje wybrany węzeł bieżący.

Wybranie węzła aproksymującego spowoduje przejście do początkowych operacji algorytmu w celu wyznaczenia kolejnego punktu aproksymującego, o ile ostatnio wybrany węzeł nie okaże się końcowym węzłem generowanego segmentu krzywej /WARKG=FALSE/.

Spełniony w dowolnym miejscu algorytmu warunek końca generacji WARKG powoduje zakończenie generacji ciągu aproksymującego.

Algorytm II

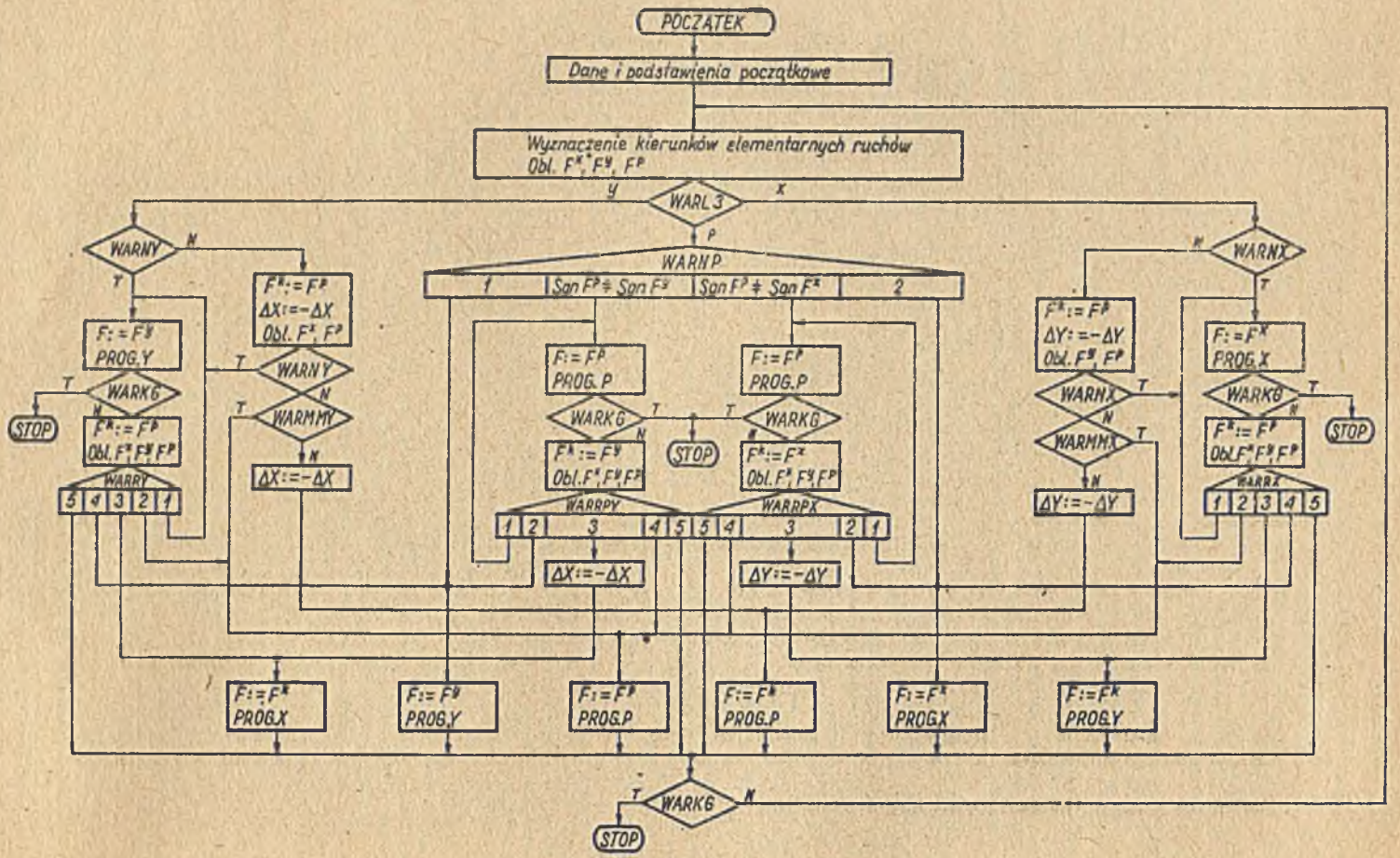
Algorytm II został przedstawiony na rys. 3.2 w postaci blokowej. Jako danych początkowych wymaga on określenia tych samych wielkości co i poprzedni algorytm, tj. zadanie stopnia krzywej, współczynników jej równania uwikłanego /2.1/ oraz wartości współrzędnych punktów: początkowego (x_p, y_p) i końcowego (x_k, y_k) aproksymowanego segmentu a także wartości funkcji $F(x_p, y_p)$ oraz kierunku obiegu krzywej D.

Wszystkie początkowe operacje algorytmu są identyczne jak w poprzednim algorytmie. Następuje więc zapamiętanie danych początkowych oraz wyznaczenie wartości pochodnych rzędów 1-n funkcji $F(x,y)$ w punkcie początkowym /wzory /2.21//. Kolejną operacją algorytmu, która w odróżnieniu od operacji początkowych jest pierwszą operacją pętli głównej algorytmu, jest, tak jak i w algorytmie I operacja wyznaczenia kierunków elementarnych ruchów /relacji /2.5// oraz operacja wyznaczenia wartości funkcji $F(x,y)$ dla trzech sąsiednich węzłów siatki $F^x(x,y)$, $F^y(x,y)$ i $F^p(x,y)$, wyznaczonych przez kierunki elementarnych ruchów i przekątną /wzory /2.17/ i /2.19//. Wartości funkcji F^x, F^y oraz F^p wykorzystywane są w dalszych fazach algorytmu. Po tej operacji następuje testowanie warunku WARL 3, który rozwidla algorytm na trzy gałęzie "ruchu głównego" wzdłuż krzywej: gałąź ruchu śledzącego wzdłuż osi x /gałąź prawa/, gałąź ruchu śledzącego wzdłuż osi y /gałąź lewa/, gałąź ruchu śledzącego wzdłuż przekątnej /gałąź środkowa/ /relacji /2.12//. Równocześnie, dla każdego z kierunków "ruchu głównego" zostaje określony kierunek "ruchu kontrolnego" którym jest kierunek przekątnej dla "ruchu głównego" wzdłuż osi x bądź y /gałęzie skrajne/ oraz kierunek osi x bądź y dla "ruchu głównego" wzdłuż przekątnej /gałąź środkowa/ /relacji /2.13/ - /2.15//.

Początkową operacją każdej ze skrajnych gałęzi algorytmu jest sprawdzenie poprawności określenia kierunku "ruchu kontrolnego". Następuje więc testowanie warunków WARN - warunków niezgodności znaków wartości funkcji w węzłach sąsiednich wyznaczonych przez kierunki "ruchu głównego" oraz "ruchu kontrolnego". Pozytywny wynik testowania tych warunków powoduje przejście do małych pętli algorytmu, w których wykonywane są ruchy śledzące przebieg krzywej. Negatywny wynik powoduje wyznaczenie nowego, różniącego się o $\pm \frac{\pi}{2}$ kierunku "ruchu kontrolnego", obliczenie wartości funkcji w węźle wyznaczonym przez nowy kierunek "ruchu kontrolnego" oraz powtórne sprawdzenie poprawności określenia kierunku "ruchu kontrolnego". Pozytywny wynik tego testu powoduje, jak poprzednio, przejście do małej pętli algorytmu. Negatywny - doprowadza do operacji wyznaczenie węzła aproksymującego /spośród dwóch węzłów wyznaczonych przez poprzedni i aktualny kierunki "ruchu kontrolnego"/ oraz wejścia do głównej pętli algorytmu.

W gałęzi środkowej /gałęzi "ruchu głównego" wzdłuż przekątnej p/ pierwszą operacją jest również dokładne określenie kierunku "ruchu kontrolnego", którym jest kierunek osi x bądź y ten, który doprowadza do węzła leżącego po przeciwnej stronie krzywej niż węzeł wyznaczony przez kierunek "ruchu głównego". Następuje więc testowanie znaków wartości funkcji F^x, F^y i F^p w węzłach sąsiednich. Dwa wewnętrzne wyjścia z warunku WARNP doprowadzają do dwóch środkowych pętli algorytmu, w których kierunkiem "ruchu głównego" jest kierunek przekątnej a kierunkiem "ruchu kontrolnego" jest kierunek osi x w gałęzi prawej oraz kierunek osi y - w lewej. Pozostałe dwa skrajne wyjścia z warunku WARNP doprowadzają do operacji wyznaczenia węzła aproksymującego /spośród węzłów sąsiednich, wyznaczonych przez kierunki osi x i y, zgodnie z zasadą min $|F(x,y)|$ oraz wejście do głównej pętli algorytmu. Wyznaczenie kierunku "ruchu kontrolnego" spełniającego określone wymagania nie jest bowiem możliwe.

W każdej z małych pętli algorytmu wykonany zostaje elementarny ruch wzdłuż kierunku głównego /obliczenie wartości pochodnych cząstkowych wszystkich stopni i uaktualnienie współrzędnych oraz



Rys. 3.2. Algorytm II dyskretyzacji krzywych algebraicznych płaskich na jednorodnej kwadratowej siatce k-spójnej

wartości funkcji/ oraz sprawdzenie kryterium końca generacji WARKG. Postać tego kryterium jest identyczna jak w poprzednim algorytmie. Pozytywny wynik testowania WARKG oznacza zakończenie działania algorytmu /zakończenie procesu dyskretyzacji/. Negatywny - prowadzi do operacji zapamiętania wartości funkcji w ostatnio wyznaczonym węźle kontrolnym /operacja podstawienia $F^k := .../$ oraz do operacji obliczenia wartości funkcji w sąsiednich węzłach wyznaczonych przez kierunki: "ruchu głównego" oraz "ruchu kontrolnego".

Kolejną operacją /w każdej z czterech/ pętli małej algorytmu jest testowanie warunku rozwidlenia pętli WARR (x,p,y). Warunki WARR rozwidlają pętlę na 5 gałęzi. Gałąź oznacza numerem 1 jest gałęzią kontynuacji śledzenia krzywej /przejście do początkowych operacji małych pętli algorytmu/. Śledzona krzywa przebiega bowiem pomiędzy węzłami wyznaczonymi przez kierunki: "ruchu głównego" oraz "ruchu kontrolnego". Gałęzie oznaczone pozostałymi numerami doprowadzają do operacji wyznaczenia węzła aproksymującego krzywą na osiągniętym etapie jej śledzenia. Istnieje bowiem przypadek, w którym dwa węzły: wyznaczony przez "ruch główny" oraz "ruch kontrolny" znajdują się po tej samej stronie krzywej. Proces śledzenia przebiegu krzywej został więc zakończony i należy wybrać węzeł aproksymujący. Gałęzie oznaczone numerami 2 i 3 są gałęziami, w których na węzeł aproksymujący wybrany zostaje /zgodnie z zasadą $\min |F(x,y)|$ / jeden z węzłów określonych przez aktualny /gałąź 2/ i poprzedni /gałąź 3/ kierunki "ruchu kontrolnego". Gałęzie oznaczone numerami 4 i 5 są gałęziami, w których na węzeł aproksymujący wybrany zostaje /również zgodnie z zasadą $\min |F(x,y)|$ /jeden z węzłów wyznaczonych przez kierunek "ruchu głównego" aktualny /gałąź 4/ bądź poprzedni /gałąź 5/.

Każde z operacji wyznaczenia węzła aproksymującego zawiera w sobie uaktualnienie wartości współrzędnych (x,y) oraz wartości funkcji F(x,y) w tym węźle a także obliczenie wartości pochodnych cząstkowych wszystkich stopni /oprócz gałęzi WARR o numerze 5, w których wartości te określone zostały wcześniej/ i każda z nich powoduje przejście do pętli głównej algorytmu, w której następuje sprawdzenie warunku WARKG i przejście do początkowych operacji algorytmu - bądź też zakończenie procesu dyskretyzacji.

4. ZAKOŃCZENIE

4.1. Implementacja algorytmów

Zaproponowane w rozdz. 3 algorytmu dyskretyzacji krzywych algebraicznych płaskich zostały zaprogramowane na komputerze MERA 400 w FORTRANIE. Implementacja ta wykonana została w celu praktycznej weryfikacji poprawności działania algorytmów, porównania założeń z otrzymanymi wynikami i sformułowania wynikających z implementacji wniosków. W ramach tej implementacji przeprowadzono dyskretyzację wybranej grupy krzywych a jej wyniki posłużyły do potwierdzenia poprawności konstrukcji oraz określenia zalet i wad algorytmów, ujawnienia innych cech algorytmów niż te, które były przewidziane na etapie syntezy oraz wyciągnięcia praktycznych wniosków i wskazówek do wykorzystania ich w praktycznych zastosowaniach.

Implementacje algorytmów wykonano początkowo w arytmetyce całkowitoliczbowej. Jednakże, w przypadku krzywych od 4-stopnia wzwyż i rozmiarach geometrycznych krzywych rzędu $\pm$ 40 jednostek,

wystąpowały w procesie działania algorytmu wyniki liczbowe przekraczające zakres liczb całkowitych /± 32767/ MERA 400. Zaistniała więc konieczność przejścia na arytmetykę zmiennoprzecinkową. Wyeliminowana została jednak w ten sposób jedna z podstawowych zalet zastosowanej metody dyskretyzacji: szybkość obliczeń i brak kumulacji błędów - wprowadzony został do obliczeń błąd systematyczny /błąd reprezentacji/.

Negatywną cechą metody będącej podstawą proponowanych algorytmów jest więc operowanie na dużych liczbach całkowitych /dużych w stosunku do zakresu liczb całkowitych minikomputerów/, zwiększającymi się wraz ze stopniem i wymiarami geometrycznymi krzywej.

W testowanych przykładach krzywych stopnia $n \leq 6$ i wymiarach geometrycznych krzywych rzędu $\pm 10^4$ jednostek /1 jednostka = 1 h - moduł siatki/ zakres liczbowy pojedynczej precyzji $\pm 0.17.10^{39}$ w zupełności wystarczał do poprawnego wykonywania niezbędnych w algorytmach operacji. Nie stwierdzało się również "degeneracji" krzywej z powodu stosowania liczb zmiennoprzecinkowych, mimo wnoszonych przez tę reprezentację błędów.

W algorytmie I implementacja pozwoliła stwierdzić istotną wadę. W dyskretyzacji krzywych wielogałęziowych, w rejonach krzywej od strony innych jej gałęzi następuje, już w niewielkiej odległości od konturu krzywej duży spadek wartości funkcji w kierunku innych gałęzi tej samej krzywej /patrz tab. 4.1/. Oznacza to fakt, że we wskazanych rejonach krzywej występuje zawężenie przedziału, w którym $|F(x,y)|$ spełnia warunki metryki. Ponieważ w algorytmie wykonywany jest tzw. "ruch od krzywej" oraz stosowany - jako metryka generacyjna - moduł wartości funkcji $|F(x,y)|$, dlatego też, dla pewnego /dostatecznie dużego/ modułu siatki h i określonego kierunku obiegu krzywej spełnione są warunki, przy których kierunek "ruchu wzdłuż krzywej" odbiega znacznie od kierunku krzywej i doprowadza do "ześlizgu" kierunku "ruchu wzdłuż krzywej" na przylegającą gałąź krzywej /lub punktu izolowanego krzywej/, powodując jej degenerację.

Podobny przypadek został opisany w [10]. Natomiast tutaj został on przedstawiony na rys. 4.1, a wystąpił w trakcie dyskretyzacji krzywej 4-stopnia $F(x,y) = (x^2+y^2)^2 - 100\,x^3 + \ldots - 3\,xy^2 - 900\,x^2 + y^2 = 0$ /koniczyna trójlistna/. Dla tego przypadku kolejność operacji algorytmu jest następująca: Punkt A został wybrany kolejnym węzłem aproksymującym. Rozpoczyna się więc wyznaczanie węzła kolejnego. "Ruch od krzywej" doprowadza do węzła B, a pierwszy "ruch wzdłuż krzywej" doprowadza do węzła C i następnie do węzłów: D, E, F itd. W każdym z nich wartość funkcji $|F(x,y)|$ jest mniejsza niż w punkcie poprzednim. Powoduje to jednak oddalanie się punktu bieżącego od krzywej. Dalsze kroczenie "ruchami wzdłuż krzywej" doprowadza do punktu (0,0), który jest punktem izolowanym /i osobliwym/ krzywej. W punkcie (0,0) następuje zapętlenie się algorytmu: dokonują się ciągłe zmiany kierunku "ruchu wzdłuż krzywej". Każdy z punktów sąsiednich ma bowiem większy moduł $F(x,y)$.

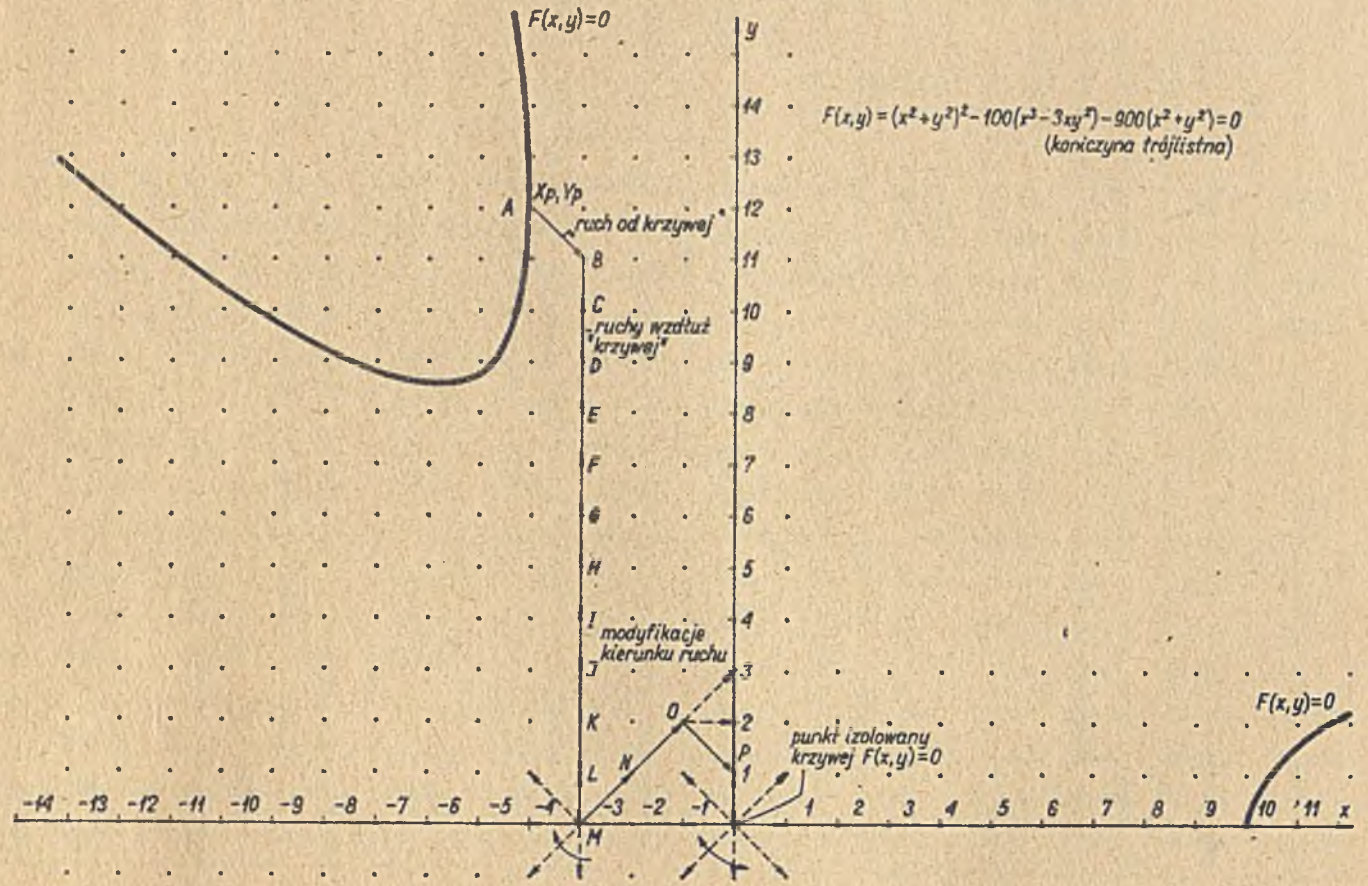
Tę wadę algorytmu można ominąć poprzez zmianę kierunku obiegu krzywej na przeciwny /zmieniając miejscami punkty: początkowy (x_p,y_p) i końcowy (x_k,y_k) oraz kierunek obiegu krzywej D/, lub zmniejszeniem modułu siatki h. Nie znana jest jednak zasada doboru kierunku obiegu krzywej ani też wartości modułu siatki zapewniające poprawną dyskretyzację krzywej /zapobiegające wystąpieniu przypadku wyżej opisanego/.

Algorytm II z założenia nie ma opisanej wady, z powodu tzw. "ruchu kontrolnego" położenia krzywej.

Inną wadą algorytmu I jest ścinanie ostrych fragmentów krzywej w przypadku aproksymacji krzywych o promieniach krzywizn zbliżonych wartością do modułu siatki h. Powstaniu tego zjawiska sprzyjała

y \ x	- 15	- 14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
15	-67500	-114459	-147964	-168939	-178284	-176875	-165564	-145179	-116524	-80379	-37500	11381	65556	124341	187076	253125
14	11641	- 42336	- 82975	-111200	-127911	-133984	-130271	-117600	- 96775	-68576	-33759	6944	52825	103200	157409	214816
13	86836	26325	-20956	-55931	-79500	-92539	-95900	-90411	-76876	-56075	-28764	4325	42484	85029	131300	180661
12	157761	91200	37769	-3456	-33375	-52864	-62775	-63936	-57151	-43200	-22839	3200	34209	69304	108425	150336
11	224116	151989	92900	45925	10164	-15259	-31196	-38475	-37900	-30251	-16284	3269	27700	56325	88484	123541
10	285625	208416	144161	91936	50841	20000	-1439	-14304	-19399	-17504	- 9375	4256	22681	45216	71201	100000
9	342036	260229	191300	134325	88404	52661	26244	8325	-1900	-5211	-2364	5909	18900	35925	56324	79461
8	393121	307200	234089	172864	122625	82496	51625	29184	14369	6400	4521	8000	16129	28224	43625	61696
7	438676	349125	272324	207349	153300	109301	74500	48069	29204	17125	11076	10325	14164	21909	32900	46501
6	478521	385824	305825	237600	180249	132896	94689	48069	42425	17125	17121	10325	12825	21909	23969	33696
5	512500	417141	334436	263461	203316	153125	112036	79221	53876	35221	22500	14981	11956	12741	16676	23125
4	540481	442944	358025	284800	222369	169856	126409	91200	63425	42304	27081	17024	11425	9600	10889	14656
3	562356	463125	376484	301509	237300	182981	137700	100629	70964	47925	30756	18725	11124	7269	6500	8181
2	578041	477600	389729	313504	249025	192416	145825	107424	76409	52000	33341	20000	10969	5664	3425	3616
1	587476	486309	397700	320725	254484	198101	150724	111525	79700	54462	35076	20789	10900	4725	1604	901
0	590625	489216	400361	323136	256641	200000	152361	112896	80801	55296	35625	21056	10881	4416	1001	0

Tab. 4.1 Wartość funkcji $F(x,y) = (x^2+y^2)^2 - 100\,x^3 - 3xy^2 - 900x^2 + y^2 = 0$ w określonych węzłach siatki



Rys.4.1. Ilustracja "ześlizgiwania się" z krzywej ruchów śledzących przebieg krzywej

możliwość wykonywania w algorytmie szeregu występujących po sobie modyfikacji kierunku ruchu/ w ramach wyznaczania jednego wektora aproksymującego/. Zjawisko to zostało z algorytmu wyeliminowane na etapie jego testowania przez wprowadzenie do działania algorytmu zasady, zgodnie z którą bezpośrednio po operacji modyfikacji kierunku ruchu następuje wyznaczenie końca wektora aproksymującego.

Między algorytmami I i II istnieją i inne różnice. W szczególności każdy z nich generuje, z reguły inną łamaną, a ponadto algorytm I składa łamaną na ogół z większej liczby krótszych odcinków.

W porównaniu z algorytmami znanymi z literatury, algorytm I jest podobny do algorytmu aproksymacji krzywych 2-stopnia opisanego w [12]. Wykorzystuje bowiem tę samą zasadę śledzenia przebiegu krzywej: "ruch od krzywej" oraz szereg "ruch wzdłuż krzywej", a ponadto wyznaczanie wartości funkcji jako miary odległości punktu od konturu. Różnica między tymi algorytmami polega na możliwości wykonywania /w opisywanym algorytmie I/ "ruchu od krzywej" oraz "ruchów wzdłuż krzywej" tak wzdłuż obydwu współrzędnych x bądź y jak też wzdłuż przekątnej a także na możliwości modyfikacji kierunku ruchu o kąt $\pm$ 45°/a nie o $\pm$ 90°/. W rezultacie generowana przez opisywany algorytm łamana jest bardziej "gładka" gdyż składa się z mniejszej liczby wektorów, a dokładność jej aproksymacji jest wyższa.

Algorytm II nie jest podobny do żadnego ze znanych w literaturze całkowitoliczbowych algorytmów aproksymacji płaskich krzywych algebraicznych.

Rysunki 4.2 - 4.15 ilustrują krzywe algebraiczne różnych stopni aproksymowane na siatce kwadratowej z modułem h = 1 mm za pomocą opisanych algorytmów I i II.

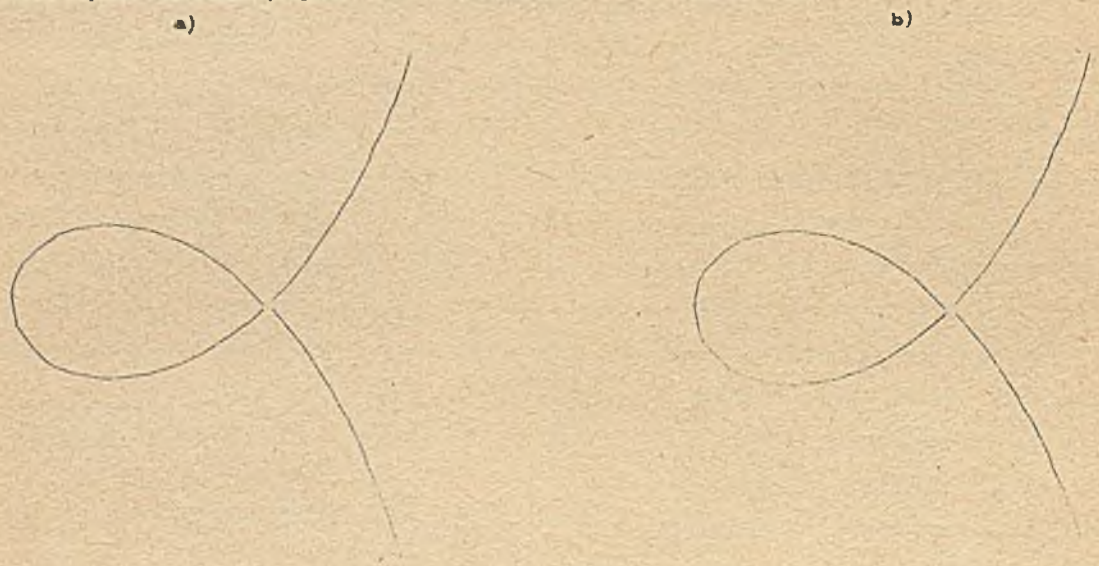
4.2. Określenie kierunku dalszych badań

Istnienie i poprawność działania zaproponowanych algorytmów dyskretyzacji krzywych algebraicznych dowolnego stopnia dowodzi, że stopień krzywej nie jest podstawową przeszkodą w opracowaniu poprawnego algorytmu dyskretyzacji, przynajmniej w przypadku krzywych płaskich algebraicznych. Co więcej, istnieją teoretyczne przesłanki do stwierdzenia, że podobnego typu algorytmy można opracować nie tylko dla krzywych mających skończone rozwinięcia potęgowe, ale w ogóle dla funkcji ciągłych /odcinki ciągłych/. Jako generalną zasadę konstrukcji tych algorytmów należałoby wykorzystać różne znaki wartości funkcji F(x,y) w punktach leżących po przeciwnych stronach krzywej oraz moduł wartości funkcji jako metrykę generacyjną. Problem kroczenia wzdłuż krzywej można realizować, nie stosując rozwinięć funkcji w szeregi potęgowe. Wartość funkcji F(x,y) w punktach-węzłach można bowiem wyznaczać wprost z równania ogólnego krzywej, a niekoniecznie za pomocą rozwinięć potęgowych. Zakładając różniczkowalność funkcji i ciągłość jej pochodnych oraz to, że pochodne I-rzędu nie są równocześnie równe zero, można, w oparciu o nie, wyznaczyć kierunek ruchu wzdłuż stycznej do krzywej, a obliczenia wartości funkcji dokonywać wprost z równania ogólnego.

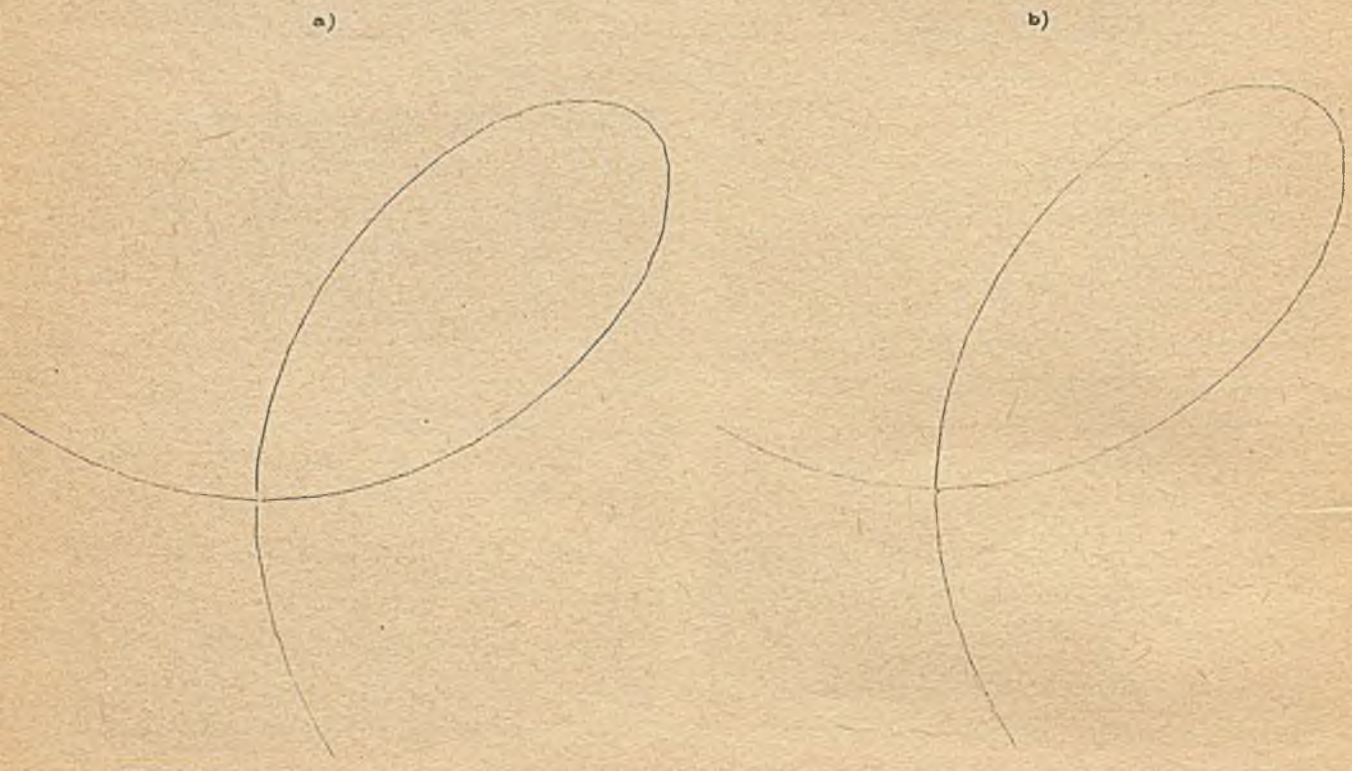
Podstawową wadą opisanych algorytmów jest konieczność operowania na bardzo dużych liczbach, rosnących /w przypadku krzywych algebraicznych/ wraz ze wzrostem stopnia krzywej i ich wymiarów geometrycznych. Wynika stąd konieczność stosowania /w komputerowej realizacji algorytmów/ zmiennoprzecinkowej reprezentacji liczb, wnoszącej /ze swojej natury/ błędy reprezentacji a w konsekwencji wielokrotnych obliczeń - również błędy obliczeń i zaokrągleń. Zjawisko to związane jest z przyjęciem za metrykę generacyjną modułu wartości funkcji F(x,y) w węzłach z otoczenia krzywej. Metryka ta jest przyczyną licznych niedogodności. Narzuca bowiem w zasadzie wartość modułu siatki, na którym należy przeprowa-

dzać dyskretyzację . Przy dostatecznie dużym module siatki /w siatce rzadkiej/ mogą występować przypadki "ześlizgiwania" się ruchów śledzących z krzywej, tak jak to zaistniało w przedstawionej tu implementacji.

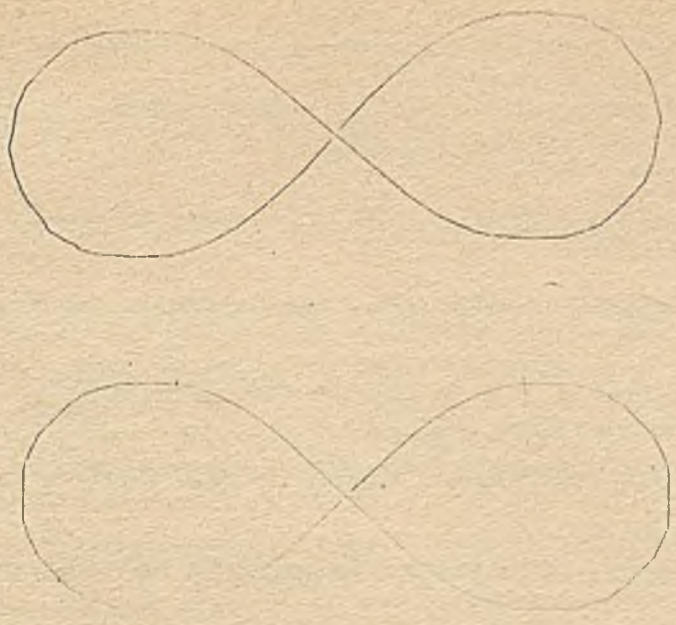
Istnieje więc potrzeba dalszych badań nad algorytmami, by doprowadzić je do takiej postaci, w której możliwe będzie działanie jedynie na krótkich liczbach stałoprzecinkowych.



Rys.4.2. Aproksymacja strofoidy (krzywa 3-stopnia) na siatce k-spójnej z modułem h=1 [mm] za pomocą algorytmu I (a) i II (b)



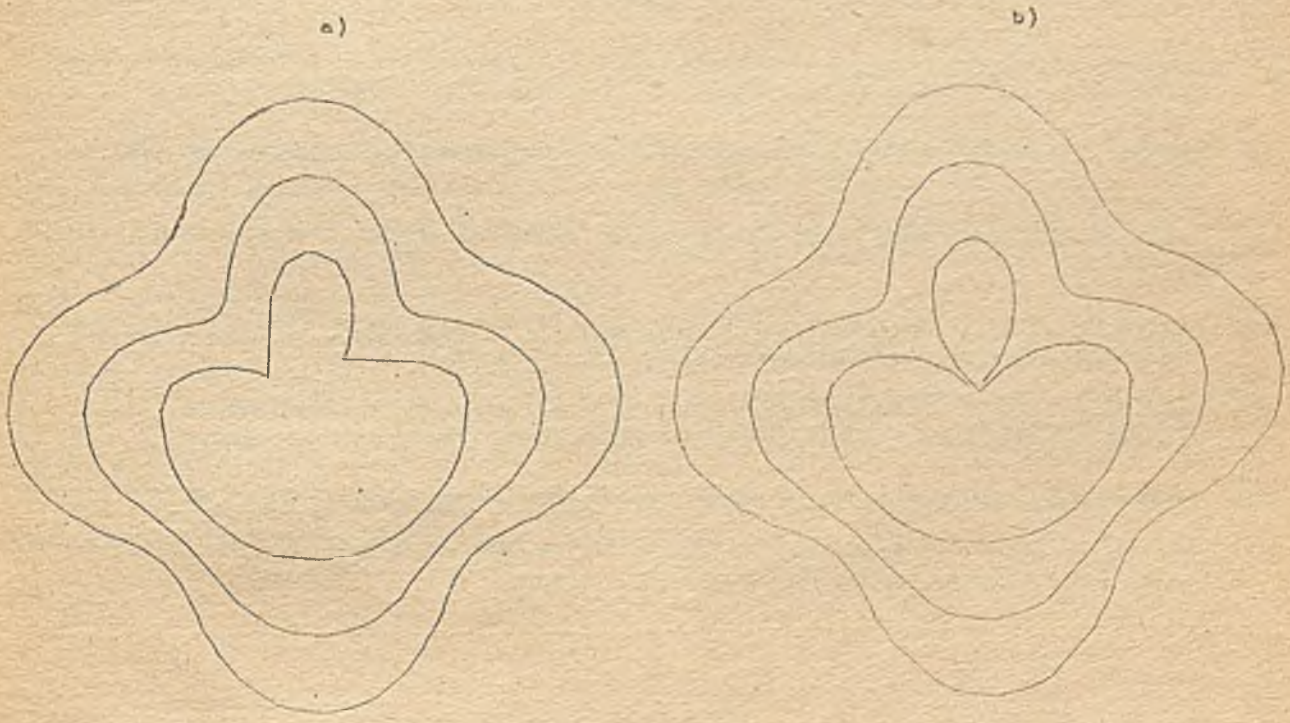
Rys.4.3. Aproksymacja liścia Kartezjusza (krzywa 3-stopnia) na siatce k-spójnej z modułem 1 [mm] za pomocą algorytmu I (a) i II (b)



a)

b)

Rys.4,4. Aproksymacja lemniskaty Bernulliego (krzywa 4-stopnia) na siatce k-spójnej z modułem h= 1 [mm] za pomocą algorytmu I (a) i II (b)



a)

b)

Rys.4,5. Aproksymacja rodziny krzywych 6-stopnia na siatce k-spójnej z modułem h=1 [mm] za pomocą algorytmu I (a) i II (b)

Literatura

[1] Lappalainen P., Ojala L.: A novel digital method generating a circle test pattern for television, The Radio and Electronic Engineer 1972 t. 42 nr 1 s. 21-27

[2] Denert E.: A method for computing points of a circle using only integers, Computer Graphics and Image Processing 1973 t. 2 nr 1 s. 83-91

[3] Pitteway M.L.V.: Integer circles, etc. - Some fourther thinghts, Computer Graphics and Image Processing 1974 t. 3 nr 3 s. 262-265

[4] Chaikin C.M.: Algorithm for high-speed curve generation, Computer Graphics and Image Processing 1974 t. 3 nr 4 s. 346-349

[5] Athani V.V.: The 16-vector algorithm for computer controlled digital X-Y plotter, IEEE Transactions on Computers 1974 t. C-24 nr 8 s. 831-835

[6] Riesenfeld F.R.: On Chaikin's algorithm, Computer Graphics and Image processing 1975 t. 4 nr 4 s. 304-310

[7] Kameswara Rao C.V.: Comment on a method for computing points of a circle using only integers, Computer Graphics and Image Processing 1975 t. 4 nr 1 s. 79

[8] Horn B.K.P.:Circle generators for display devices, Computer Graphics an Image, Processing 1976 t. 5 nr 3 s. 280-288

[9] Sutcliffe D.C.: An algorithm for drawing the curve $f(x,y)=0$, The Computer Journal 1976 t. 19 nr 3 s. 246-249

[10] Sutcliffe D.C.: A remark on a contouring algorithm, The Computer Journal 1976 t. 19 nr 4 s. 333-335

[11] Mokrzycki W.: Algorytmy i ocena dokładności układowych generacji krzywych 2-stopnia na siatkach kwadratowych, Biuletyn Informacyjny MERA-IMM 1977 RXVI nr 5-6 s. 89-109

[12] Mokrzycki W.: Analiza cyfrowych metod generacji układowej krzywych drugiego stopnia, Prace naukowo-badawcze IMM 1978

[13] Ward A.S.: Real time plotting of approximate contour maps, Communications of ACM 1978 t. 21 nr 9 s. 788-790

[14] Maxwell P.C., Baker P.W.: The generation of polygons representing circles ellipses and hyperbolas, Computer Graphics and Image Processing 1979 t. 10 nr 1 s. 84-93

[15] Moss R., Lindgard A.: Parametric spline curves in integer arithmetic designed for use in microcomputer controlled plotters, Computers and Graphics 1979 t. 4 nr 1 s. 51-61

[16] Earnshaw R.A.: Line tracking for incremental plotters, The Computer Journal 1980 t. 23 nr 1 s.46-52

[17] Freeman H., Saghri J.A.: Comparative analysis of line -drawing modeling schemes, Computer Graphics and Image Processing 1980 t. 12 nr 2 s. 203-223

Дискретизация алгебраических кривых на однородных квадратных к-связаных сетках

Р е з ю м е

В публикации анализируется т.н. целочисленные дискретизации (аппроксимации) алгебраических кривых осуществляемые на к-связаных квадратных сетках. Они заключаются в приближении кривых рядом точек (узлов сетки) имеющих целочисленные координаты, в котором разстояние между очередными точками ряда неограничено сверху. Эти приближения не вносят систематических ошибок. Сделано обзор литературы касающейся методов и алгоритмов целочисленной дискретизации кривых а также ссинтезировано два новые алгоритма дискретизации плоских алгебраических кривых n-степени на однородных, квадратных, к-связаных сетках.

Оговорено результаты внедрения алгоритмов на миникомпьютере MERA 400 приводя чертежи четырех кривых (3-, 4- и 6-степени выполненные чертежным автоматом DIGIGRAF 1612) сдискретизированных с помощью разработанных алгоритмов.

Алгоритмы найдут применение не только в машинной графике и машинной обработке образов и чертежей, но также и в цифровом управлении фрезерными станками (и другими промышленными процессами) и т.п.

An algebraic curve discretization on k-connected homogeneous square grids

A b s t r a c t

In this paper there have been analysed methods of so called integer algebraic curve discretizations (approximations) realized on k-connected square grids.

In all these methods curves are approximated by a sequence of points (nodes of a grid), coordinates of which are integers and distance between the adjacent points is not limited in advance. Results are not charged with systematical errors.

The literature concerning methods and algorithms of integer curve discretization has been reviewed. Two new algorithms of 2D n-degree algebraic curve discretization on homogenous k-connected square grids have been synthesized. Results of on implementation of the algorithms on MERA 400 minicomputer on corresponding drawings of four curves (3-, 4- and 6-degree plotted on the DIGIGRAF 1612 plotter) have been discussed.

These algorithms can be applied in computer graphics image and drawing processing, tool numerical control (or other production proceses), etc.

Prace naukowo-badawcze Instytutu Maszyn Matematycznych

O pewnych problemach projektowania komputerowych systemów graficznych

Wojciech MOKRZYCKI

W wielu praktycznych zastosowaniach komputerowych systemów graficznych istnieje potrzeba prezentacji dość złożonych obrazów, w których liczba elementów jest powyżej tysiąca: punkty, wektory, znaki alfanumeryczne i specjalne. Są to często obrazy dynamiczne, występujące np. w graficznych systemach wspomagania projektowania inżynierskiego. W systemach tych są stosowane - jako urządzenie obrazowania i graficznego współdziałania operatora z systemem - monitory graficzne z ciągłym odnawianiem obrazu. Ciągłość taka oznacza, że tzw. kod kadru /program wyświetlanego obrazu zapisany w postaci ciągu instrukcji monitora graficznego/ musi być w całym czasie prezentacji obrazu cyklicznie /z częstotliwością ok. 50 Hz/ przetwarzany przez monitor. Ponadto prezentowanie obrazów dynamicznych wymaga - również w czasie jednego kadru - zmiany treści obrazu i transmisję tego obrazu do monitora. Z tych względów obrazom bardziej złożonym nie daje się zapewnić dynamiki opierając się na samym oprogramowaniu. Zachodzi zatem konieczność realizacji w monitorze graficznym pewnych funkcji systemu /realizowanych w prostych systemach graficznych za pomocą oprogramowania/ w sposób układowy, np. generowanie znaków specjalnych i niektórych często używanych krzywych. Monitor staje się wówczas urządzeniem złożonym i kosztownym. Można go zaliczyć raczej do małych komputerów niż do rozbudowanych kanałów przesyłowych. W projektowaniu takiego systemu należy rozważyć następujące problemy:

- stopień fizycznej separacji między monitorem i komputerem
- niezbędną moc obliczeniową, jaką powinien mieć monitor graficzny.

Umieszczenie monitora w pobliżu komputera umożliwia wykorzystanie jego pamięci operacyjnej do odnawiania obrazu. Staje się więc możliwe korzystanie /dla celów odnawiania obrazu/ z bloków pamięci, w których informacja graficzna jest przygotowywana. Nie ma wówczas potrzeby przesyłania jej do lokalnej pamięci monitora. Tego typu sprzężenie komputer - monitor w komunikacji system - operator jest wygodniejsze. Zapewnia ono bowiem elastyczność i prostotę sterowania danymi graficznymi. Jeżeli jednak monitor jest zbyt związany z komputerem, to współzawodnictwo o dostęp do pamięci, jak i wymagania monitora odnośnie współpracy z komputerem mogą obniżyć efektywność zarówno jego jak i monitora. Jeżeli jednak komputer pracuje z "podziałem czasu" a monitor może w tym czasie zmieniać informację w pamięci, to istnieje niebezpieczeństwo zniszczenia tych programów. Z tych właśnie względów istnieją przesłanki do odłączania monitora od komputera pracującego z "podziałem czasu" i wyposażenia go w pamięć lokalną, przeznaczoną do pamiętania kodu kadru. Jeśli monitor ma być umieszczony w dużej odległości od komputera, to wymagania odnośnie takiej pamięci są identyczne. Powoduje to jednak istotną rozbudowę systemu, pociągając za sobą znaczne zwiększenie kosztów wykonania. Ponadto komplikuje się również problem komunikacji między komputerem i monitorem, szczególnie w przypadku, gdy oddalony monitor ma małą moc obliczeniową.

z wyświetlany obraz stosunkowo często się zmienia. W tym przypadku wymagane jest szybkie opracowywanie złożonych funkcji komunikacyjnych. Z tych też względów lepszym rozwiązaniem jest umieszczenie monitora w pobliżu komputera, tak aby mógł on korzystać /dla celów odnawiania obrazu/ z pamięci operacyjnej. Chociaż stwarza to problemy w systemach z "podziałem czasu" tym nie mniej wydaje się, że uzyskane korzyści uzasadniają takie właśnie rozwiązanie tych problemów. Monitor z własną pamięcią ma istotne wady, można je przyjąć tylko wtedy, gdy zmusza nas do tego zapewnienie szerokiego pasma łącz.

Problem mocy obliczeniowej monitora występuje w przypadku, gdy znajduje się on w dużej odległości od komputera. Jeżeli jednak przewidujemy jedynie wyświetlanie prawie statycznego obrazu, a ponadto można dopuścić /stosunkowo/ długie opóźnienia w komunikacji użytkownik - system, to nie ma wówczas potrzeby włączania do monitora uniwersalnej mocy obliczeniowej. Moc ta potrzebna jest w przypadku, gdy przewiduje się wyświetlanie dynamicznych obrazów i szybkie opracowywanie złożonych funkcji komunikacyjnych. Wówczas problemem jest, czy włączyć moc obliczeniową w monitor, czy też wykonać w komputerze specjalny kanał dla niego. Zależy to od typu i złożoności zadań do realizacji których system jest projektowany. Należy tu wyróżnić następujące możliwości:

1. monitor musi generować obrazy na podstawie pewnej wewnętrznej reprezentacji danych, która może wymagać wielokrotnego wywoływania podprogramów;

2. monitor może tworzyć obrazy lub ich elementy na podstawie obliczeń, a nie na podstawie ich statycznego przedstawienia w pamięci, bowiem wiele elementów /np. śledzenie pióra świetlnego, siatki punktowe, punkty przypadkowe/ można łatwo opisywać za pomocą procedur generacyjnych niż przez wyliczanie ich elementów;

3. monitor może mieć możliwość bezpośredniego sprzężenia zwrotnego z użytkownikiem lub obsługiwanie prostych zdarzeń /funkcji/ komunikacyjnych jak redagowanie lub śledzenie pióra świetlnego;

4. monitor może tworzyć reprezentację wyświetlanego obrazu na podstawie danych wyższego rzędu wziętych z programu użytkowego; powoduje to konieczność włączenia procedur obliczeniowych dla obrotów, skalowania, translacji, generowania krzywych i innych, o ile nie są one wykonywane przez układy elektroniczne monitora.

W przypadku 1, gdy monitor generuje obrazy na bazie pewnej wewnętrznej reprezentacji danych, to musi on interpretować strukturę danych z pamięci. Pożądane jest natomiast wyeliminowanie wtórnego pakietu do wyświetlania, generowanego zwykle na podstawie danych wyższego rzędu. Im bardziej złożone struktury monitor może przetwarzać, tym bliżej jesteśmy postawionego celu. Tym niemniej dla zwiększenia szybkości generacji obrazu celowe jest śledzenie tych struktur i wykonywanie rozkazów wyświetlania umieszczonych w tych strukturach. Nie jest natomiast celowe dawanie monitorowi uniwersalnej mocy obliczeniowej jedynie dla interpretacji takich struktur.

W przypadkach 2 oraz 3 tj. wówczas, gdy monitor tworzy obrazy za pomocą procedur generacyjnych lub gdy ma on bezpośrednie sprzężenie zwrotne z użytkownikiem, to nie jest istotne skąd moc ta pochodzi. Istotne jest tylko, aby była ona dostępna na każde żądanie. Można przewidzieć bądź wysoki priorytet przerwania w komputerze, ryzykując obniżenie się wówczas jego efektywności lub też zainstalować niezbędną moc obliczeniową w monitorze.

W przypadku 4, tworzenie reprezentacji wyświetlanego obrazu na podstawie danych wyższego rzędu

wziętych z programu użytkowego nie powinno być realizowane w monitorze. Monitor powinien być urządzeniem specjalizowanym, mającym za zadanie wyświetlanie obrazu na podstawie odczytanych z pamięci danych. Struktura danych użyteczna dla manipulacji na wysokim poziomie przedstawia obiekty abstrakcyjnie i posiada jako parametry dane numeryczne niezbędne przy generacji poszczególnych obrazów. Monitor powinien być zdolny do śledzenia takich struktur bezpośrednio, lecz bez generacji wtórnych pakietów do wyświetlania informacji zawartych w tych strukturach. Generacja ich powinna być realizowana w komputerze.

Uniwersalna moc obliczeniowa, niezależnie od celu jakiemu ma służyć, powinna pochodzić z systemu centralnego. Jeżeli jest ona niedostateczna, to system a nie monitor wymaga jej zwiększenia.

Obecnie kierunek rozwojowy systemów graficznych jest taki, że małe samodzielne komputery zastępowane są przez specjalizowane, które mają procesory graficzne dołączane bezpośrednio do szybkiego uniwersalnego komputera, o znacznej możliwości przetwarzania.

Prace naukowo-badawcze Instytutu Maszyn Matematycznych

Rozwiązywanie niesymetrycznych i rozrzedzonych układów równań liniowych

Adam FAUDROWICZ

Wstęp

Najczęściej rozwiązywanymi rozrzedzonymi układami równań liniowych są takie układy, których macierze współczynników są albo symetryczne, albo przynajmniej uporządkowane. Mogą to być macierze pasmowe, blokowe, pseudosymetryczne i in. W pracach [1],[2],[5],[9],[10] opisano wiele wyspecjalizowanych metod rozwiązywania takich układów. Niniejsza praca dotyczy rozwiązywania układów równań, których macierze współczynników nie mają żadnych cech regularności. Macierze takie występują przy rozwiązywaniu układów równań Kirchoffa (elektronika, energetyka) przy zagadnieniach przepływów międzygałęziowych (ekonomia) oraz przy rozwiązywaniu układów równań nieliniowych metodą Newtona [4].

Metoda eliminacji Gaussa

Rozwiązanie układu równań liniowych

$$Ax=b \tag{1}$$

metodą eliminacji Gaussa polega na znalezieniu takiej macierzy L z samymi zerami nad przekątną (lower triangle) aby jej iloczyn z macierzą współczynników A dawał w wyniku macierz trójkątną górną (upper triangle) U z samymi zerami pod przekątną i jedynkami na przekątnej.

Mnożąc równanie (1) lewostronnie przez L, otrzymujemy:

$$U x = b1 \tag{2}$$

gdzie $b1 = L b$.

Widać więc, że rozwiązywanie układu równań liniowych metodą eliminacji Gaussa składa się z dwóch etapów:

- znalezienie macierzy L - tzw. przebieg prosty
- rozwiązanie układu(2)- tzw. przebieg odwrotny

Jeżeli problem rachunkowy wymaga wielokrotnego rozwiązywania układu (1) przy zmieniającej się jedynie kolumnie prawych stron b, warto raz obliczone macierze L i U zapamiętać i używać wielokrotnie. Zajmuje to wprawdzie pamięć komputera, ale znacznie skraca czas wykonywania obliczeń. Gdy przy każdym nowym przypadku rozwiązywania układu (1) zmienia się macierz A (sytuacja taka występuje np. przy rozwiązywaniu równań nieliniowych metodą Newtona), zapamiętywanie macierzy L jest zbędne. Można jej nawet w ogóle nie obliczać, lecz krok po kroku przekształcać macierz A w macierz U, obliczając jednocześnie wektor wyrazów wolnych b1. Taki jest właśnie algorytm programu opisywanego w tym opracowaniu.

Ponieważ metoda eliminacji Gaussa jest dość rozpowszechnionym sposobem rozwiązywania równań liniowych [1],[2],[8],[11], szczegółowo przedstawiono tylko dwa zagadnienia charakterystyczne w opisywanym programie: wybór elementu głównego oraz sposób zapamiętania macierzy rozrzedzonej. W zakończeniu opisano wyniki działania programu.

Wybór elementu głównego

Najprostszym sposobem wyboru elementu głównego jest poszukiwanie największego co do modułu

dokładność dyskretyzacji jest określona dokładnością wyznaczania pierwiastka kwadratowego z i_1 i jest nie gorsza niż jedna jednostka. Przedstawiony algorytm może jednak dyskretyzować jedynie okręgi.

Oryginalną metodą całkowitoliczbowej dyskretyzacji okręgu zaproponował F.Denert w pracy [2] z 1973 r. Punktem wyjścia w tej metodzie jest spostrzeżenie, że istnieje całkowitoliczbowe rozwiązanie równania okręgu $x^2+y^2=R^2$. Trzy liczby x_1,y_1,R_1 spełniające to równanie są zwane trójką Pitagorasa. Jedna taka trójka dostarcza ośmiu punktów na okręgu, które łącznie z czterema punktami na obu osiach dają 12 punktów aproksymujących okrąg. Ponieważ jednak trójki Pitagorasa istnieją jedynie dla nielicznych okręgów o promieniu całkowitoliczbowym, a aproksymacja dwunastobokami jest niewystarczająca, zwłaszcza dla większych okręgów, więc w tym wypadku wyznacza się również trójki dla okręgów o promieniach mniejszych. Przez punkty te i środek okręgu prowadzi się linie aż do przecięcia z okręgiem zadanym. Przemieszczając się wzdłuż tych linii wg podanego algorytmu /wykonując odpowiednią liczbę ruchów wzdłuż osi x,y oraz wzdłuż przekątnej/ osiąga się dodatkowe punkty w pobliżu okręgu zadanego.

Metoda Denerta jest dość złożona. Szczególnie złożone jest wyznaczanie dodatkowych punktów na okręgu. Poza tym używa się dzielenia całkowitoliczbowego, które, co prawda nie wpływa na dokładność dyskretyzacji, ale jest operacją stosunkowo złożoną /długotrwałą/. Poza tym podział kątowy dokonywany przez wyznaczone na okręgu punkty jest ekstremalnie nierówny, co powoduje duże błędy dyskretyzacji /rzędu kilku jednostek/. W sumie metoda ta jest mało atrakcyjna ze względu na jej złożoność, małą dokładność oraz na brak uniwersalności - dotyczy jedynie dyskretyzacji okręgów.

Powyższy algorytm Denerta ulepszał M.L.V. Pitteway w publikacji [3] z 1974 r. Proponuje on mianowicie, aby kolejne punkty aproksymujące okrąg były wybierane wg ciągu $\left[(R-i)^2,\ NC\ (2i^2R^2-i^4)\right]^{\frac{1}{2}}$ /NC - najbliższa liczba całkowita, R - promień okręgu, i - numer kolejnego punktu/. Postępując tak otrzymuje się całkowitoliczbową aproksymację okręgu odcinkami linii prostej o mniej więcej jednakowej długości. Liczba odcinków aproksymujących jest proporcjonalna do $\sqrt{R}$.

Ostatni algorytm ma, w porównaniu z algorytmem Denerta, następującą przewagę: punkty aproksymujące okrąg rozłożone są na nim równomiernie. Liczba tych punktów jest /dla większych okręgów/ znacznie mniejsza niż w algorytmie Denerta - przy zachowaniu tej samej dokładności aproksymacji, złożoność obliczeniowa algorytmu jest taka sama jak i algorytmu Denerta. W sumie jednak algorytm jest stosunkowo złożony, wymaga bowiem oprócz sumowania również potęgowania oraz pierwiastkowania liczb.

Zasadniczo odmienną metodę od dotychczasowych całkowitoliczbowych metod dyskretyzacji krzywych przedstawia G.M.Chaikin w [4] z 1974 r. W tej metodzie niezamkniętą krzywą bez punktów przegięcia zadaje się czterema całkowitoliczbowymi punktami P1, P2, P3 i P4 wybranymi w następujący sposób: punkty P1 i P4 są odpowiednio punktem początkowym i końcowym krzywej. Przez punkty te prowadzi się styczne do krzywej L12 i L34; punkty P2 i P3 stanowią punkty przecinania się linii L12 i L34 z linią L23 poprowadzoną w ten sposób, że jest ona styczna do krzywej, a punkt styczności dzieli odcinek L23 na połowę. Algorytm generacji krzywej działa następująco: znajduje się środki odcinków L12 i L23; nowym punktem P4 staje się punkt środkowy odcinka L23 /punkt styczności z krzywą/:

$$P4 : = \frac{1}{2}\ (P2+P3)$$

Nowym punktem P2 jest środek odcinka L12, a nowym punktem P3 jest środek odcinka wyznaczonego przez poprzedni punkt P2 i punkt styczności linii L23 z krzywą /czyli nowy punkt P4/:

$$P3_1 = \frac{1}{2}\ (P2+P4)$$

$$P2_1 = \frac{1}{2}\ (P2+P1)\ .$$

Przed wyznaczeniem nowych punktów P2, P3 i P4 poprzednie współrzędne punktów P2 i P4 zapamiętuje się w specjalnym, spychanym do tyłu stosie. Z kolei znajduje się środki nowo wyznaczonych odcinków L12 i L23 oraz wyznacza się nowe punkty kolejno: P4, P3, P2, identycznie jak za pierwszym razem. Operację tę powtarza się tak długo aż odległość pomiędzy sąsiednimi punktami P1, P2,P3 i P4 staje się nie większa od zadanej. Wówczas to łączy się kolejno ze sobą punkty: P1, P2, P3 i P4, a punkt P4 staje się nowym punktem P1; ostatnio umieszczone w stosie punkty P2 i P4 pobiera się jako aktualne i cykl podziału odcinków, łączenia ich środków oraz sprawdzania odległości pomiędzy sąsiednimi punktami powtarza się aż do osiągnięcia punktu P4 zadanego pierwotnie.

Algorytm Chaikina jest stosunkowo prosty. Używane są bowiem tylko operacje sumowania liczb całkowitych, dzielenia przez 2 oraz porównywania liczb. Do tworzenia stosu istnieje jednak potrzeba używania pamięci, której pojemność zależy od rozmiarów generowanej krzywej. Dla pola obrazowania 1024x1024 jednostek są wymagane cztery dziesięciosłowowe stosy. Prędkość generacji jest zmienna i ma charakter narastający. Na początku generacji wymagana jest większa liczba podziałów. W trakcie generacji wykorzystuje się te punkty ze stosu, które wyznaczono w poprzednich cyklach generacji. Do wyznaczania n punktów wymaganych jest n iteracji. Istotną zaletą algorytmu jest to, że daje się on łatwo rozszerzyć na krzywe trójwymiarowe. Autor algorytmu nie podaje jednak żadnych danych na temat dokładności generacji krzywej. Sądzić jednak należy, że nie jest ona zbyt wysoka /błędy rzędu kilku-kilkunastu jednostek/.

R.F.Riesenfeld w [6] /1975 r./ ulepsza algorytm Chaikina z [4] /1974 r./ do postaci, w której może on generować również krzywe zamknięte. Wykazuje ponadto, że krzywe generowane przez ten algorytm są parametrycznymi krzywymi B-giętkimi /B-spline/ 2-stopnia. Pozostaje jednak nierozwiązany problem dokładności generacji zdefiniowanych w podobny sposób krzywych.

V.V.Athani w [5] /1975 r./, a następnie R.A. Earnshaw w [16]/1980 r./,rozwijają algorytm generacji prostej oparty na metodzie 8-kierunkowych elementarnych przemieszczeń /metodę aproksymacji na 8-spójnej jednorodnej siatce kwadratowej/ wprowadzając 8 dodatkowych kierunków ruchów. Generację prostej za pomocą tego algorytmu uzyskuje się wykonując elementarne ruchy do jednego z szesnastu najbliższych węzłów /aproksymacja na 16-spójnej siatce/. Relacje określające te ruchy wykorzystują zależność między znakami i wielkością ilorazu wartości $\Delta x, \Delta y$ - rzutów generowanego odcinka prostej. Odległość między kolejnymi węzłami ciągu aproksymującego może więc przybierać trzy różne wartości: 1, $\sqrt{2}$ oraz $\sqrt{5}$.

Powyższy algorytm opracowany został do zastosowań w plotakach xy. Może być jednak stosowany również w innych urządzeniach obrazowania czy sterowania, działających na płaszczyźnie /w przestrzeni/ dyskretnej. W porównaniu ze swoim pierwowzorem pozwala on aproksymować proste za pomocą mniejszej liczby przemieszczeń oraz z mniejszym o połowę szczytowym odchyleniem $|\varepsilon| \leq \frac{1}{4}$ i chociaż wymaga dokonania zmian konstrukcyjnych w ploterach, to jednak uzyskane efekty w szybkości i dokładności pracy plotera w pełni rekompensują wynikające z tych zmian niedogodności.

Algorytm dotyczy jednak tylko dyskretyzacji prostej.

W publikacji [8] z 1975 r. Horn opisuje metodę wielomianowej całkowitoliczbowej aproksymacji

Wyniki

Napisany w języku FORTRAN program przetestowano na minikomputerze PDP-15, rozwiązując trzy układy równań. Współczynniki macierzy A mieściły się w granicach $\pm 0.01 \div \pm 100$. Wyniki testów przedstawiono w tabeli 1.

Liczba zmiennych N	Początkowa liczba elementów niezerowych	Czas pracy s	Błąd rozwiązania
7	25		$5 \cdot 10^{-5}$ $8 \cdot 10^{-7}$
20	89	3	$7 \cdot 10^{-7}$ $5 \cdot 10^{-7}$
50	295	47	$5 \cdot 10^{-7}$ $4 \cdot 10^{-6}$

W kolumnie pierwszej podano rząd zadania. W kolumnie drugiej podano wejściową liczbę elementów niezerowych (łącznie z wyrazami wolnymi). W kolumnie trzeciej podano czas pracy maszyny w sekundach. W kolumnie czwartej podano średni błąd wyników obliczony według wzoru:

$$= \frac{1}{N} \sum_{i=1}^{N} \left| x_i^o - x_i \right| \qquad (7)$$

gdzie: N - liczba zmiennych
x_i^o - dokładna znana wcześniej wartość i-tej zmiennej, spełniającej układ równań
x_i - obliczona wartość i-tej zmiennej

W kolumnie piątej podano dla porównania średni błąd wyników otrzymanych przy tradycyjnym sposobie wyboru elementu głównego (pełny wybór elementu największego co do modułu).

W tabeli 2 podano wyniki obliczone dla przykładu trzeciego. Współczynniki macierzy są tak dobrane, że x_i^o jest resztą z dzielenia liczby "i" przez 10, czyli - pisząc w języku FORTRAN

$$x_i^o = \text{FLOAT}(\text{MOD}(I,10)), \qquad (8)$$

Wzór(8) jest również uzasadnieniem wzoru (7): błędu względnego

$$\frac{x_i - x_i^o}{x_i^o}$$

nie można było obliczać, ponieważ niektóre x_i^o są zerami.

Tabela 2

```
KMAX=  295
 0   47
X( 1)=  1.0000001
X( 2)=  1.9999999
X( 3)=  2.9999996
X( 4)=  3.9999987
X( 5)=  5.0000000
X( 6)=  5.9999992
X( 7)=  6.9999979
X( 8)=  7.9999992
X( 9)=  8.9999971
X(10)= -0.0000035
X(11)=  0.9999999
X(12)=  2.0000001
X(13)=  2.9999992
X(14)=  3.9999996
X(15)=  4.9999990
X(16)=  6.0000019
X(17)=  7.0000001
X(18)=  7.9999998
X(19)=  8.9999993
X(20)=  0.0000000
X(21)=  0.9999998
X(22)=  2.0000000
X(23)=  2.9999999
X(24)=  3.9999987
X(25)=  4.9999992
X(26)=  5.9999993
X(27)=  7.0000004
X(28)=  8.0000000
X(29)=  8.9999974
X(30)=  0.0000002
X(31)=  1.0000000
X(32)=  1.9999993
X(33)=  3.0000002
X(34)=  3.9999988
X(35)=  4.9999998
X(36)=  5.9999950
X(37)=  6.9999995
X(38)=  8.0000000
X(39)=  8.9999969
X(40)= -0.0000001
X(41)=  0.9999993
X(42)=  2.0000009
X(43)=  2.9999980
X(44)=  3.9999994
X(45)=  4.9999998
X(46)=  5.9999996
X(47)=  6.9999980
X(48)=  8.0000024
X(49)=  9.0000005
X(50)= -0.0000000
```

Literatura

[1] Bjorck A., Dalquist G.: Numerische Methoden. Oldenbourg Verlag Munchen 1972

[2] Bunch R.J., Rose D.J.: Sparse Matrix Computation. Acad.Press London 1976

[3] Duff I.S.: A survey of sparse matrix research. Proc.IEEE 1977 t.54 nr 4

[4] Faudrowicz A.: Optymalizacja rozmieszczenia elementów układów scalonych. Prace PIE 1978 t.19 nr 68 s.39-51

[5] McNamee J.M.: A sparse matrix package.Comm. ACM 1971 t.14 nr 4 s.265-273

[6] Pankowski F.: Solution of sparse linear equation syatems. Zastosowania Matematyki 1974 t.14 nr 3 s.481-510

[7] Petrenko A.I., Vlasov A.I.: Tablicnye metody modelirovanija elektronnych schem na ECVM. Nauka Kiev 1977

[8] Soherman A.H.: Algorithms for sparse Gaussian elimination with partial pivoting. ACM Trans. of Math.Software 1978 t.4 nr 12 s.330-338

[9] Shacham M., Kehat E.: A direct method for the solution of large sparse systems of linear eqations. The Comp. J. 1976 t.19 nr 4 s.353-359

[10] Rakus A.: Zastosowanie schematu Cholesky'ego do rozwiązywania układów równań liniowych. Algorytmy 1974 t.11 nr 20 s.73-83

[11] Tewarson J.: Sparse Matrices.Acad.Pres 1973

[12] Curtis A.R., Reid J.K.: The solution of large sparse unsymetric system of linear equations. J.Inst. Math.Appl. 1971 t.8 s.344-353

Warunki prenumeraty wydawnictw

Instytutu Maszyn Matematycznych

Prenumeratę na kraj przyjmują Oddziały RSW "Prasa-Książka-Ruch" oraz urzędy pocztowe i doręczyciele w terminie do dnia 25 listopada na rok następny.

Jednostki gospodarki uspołecznionej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW "Prasa-Książka-Ruch", w miejscowościach zaś, w których nie ma Oddziałów RSW - w urzędach pocztowych.

Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO nr 1153-201045. Prenumerata ze zleceniem wysyłki za granicę jest droższa od prenumeraty krajowej o 50% dla zleceniodawców indywidualnych i o 100% dla zlecających instytucji.

Ceny prenumeraty rocznej:

Biuletyn Informacyjny - Nauki i Techniki Komputerowe - 1560,-
Przegląd Dokumentacyjny - Nauki i Techniki Komputerowe 1260,-
Prace Naukowo-Badawcze IMM - cena jednego zeszytu - 220,-
cena prenumeraty rocznej 660,-

Zamówienia na prenumeratę Prac Naukowo-Badawczych IMM przyjmuje Dział Sprzedaży Wysyłkowej Ośrodka Rozpowszechniania Wydawnictw Naukowych PAN, Warszawa Pałac Kultury i Nauki, tel. 20-02-11 w. 2516. Egzemplarze pojedyncze Prac Naukowo-Badawczych IMM są do nabycia w księgarni Ośrodka Rozpowszechniania Wydawnictw Naukowych PAN, Warszawa, Pałac Kultury i Nauki, tel. 20-02-11 w. 2105.

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej
P. 4201/82

a 220 zł