

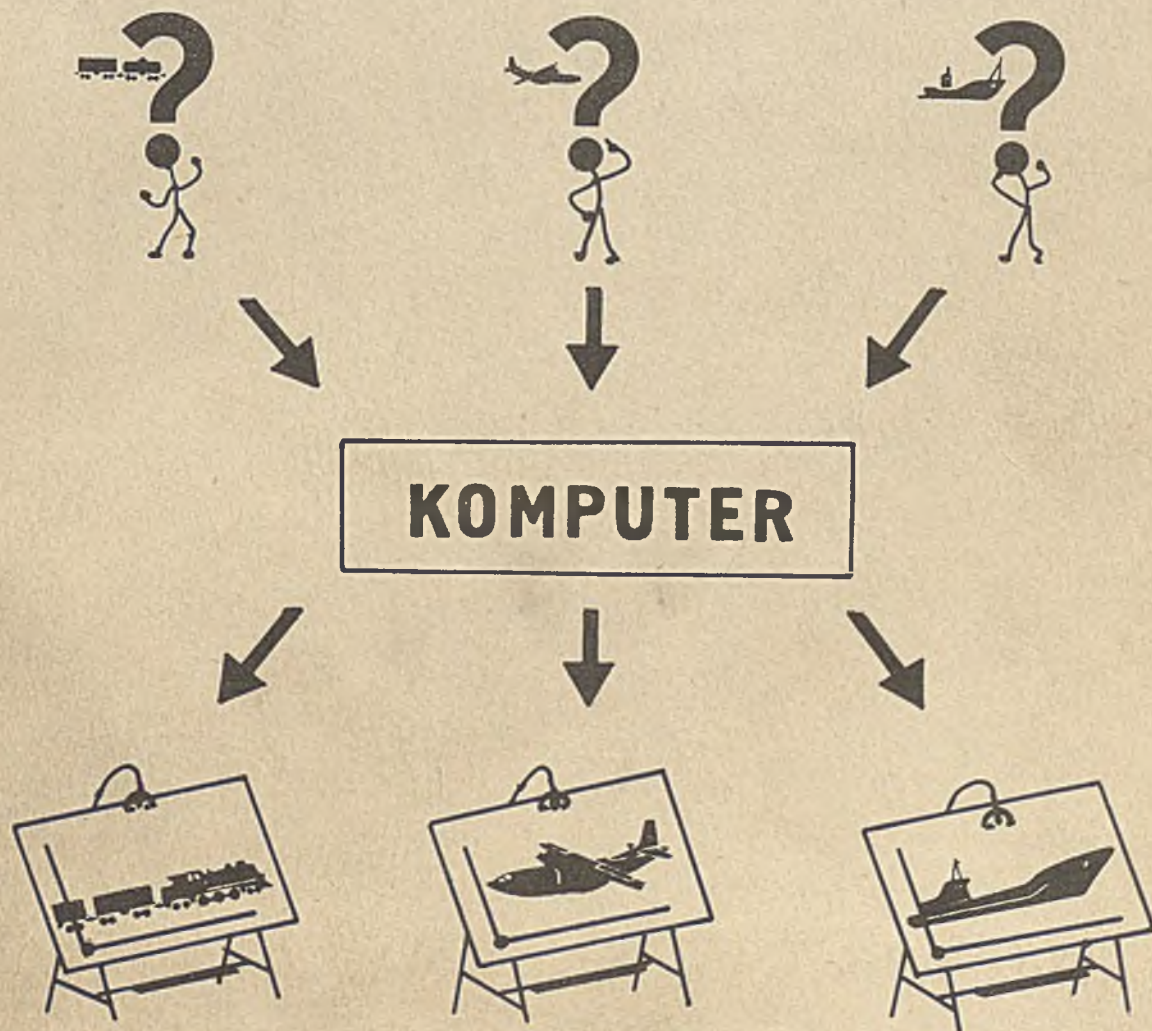
# biuletyn informacyjny



5-6  
'77



OBIEKTOWE  
SYSTEMY  
KOMPUTEROWE



Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej „MERA”  
Instytut Maszyn Matematycznych „MERA IMM” Branżowy Ośrodek INTE

DWUMIESIĘCZNIK

Wydaje: CENTRUM NAUKOWO-PRODUKCYJNE TECHNIK KOMPUTEROWYCH I POMIARÓW  
INSTYTUT MASZYN MATEMATYCZNYCH  
BRANŻOWY OŚRODEK INFORMACJI NAUKOWEJ TECHNICZNEJ I EKONOMICZNEJ

KOMITET REDAKCYJNY

dr inż. Stanisława BONKOWICZ-SITTAUER, mgr inż. Roman CZAJKOWSKI,  
mgr Hanna DROZDOWSKA (sekretarz redakcji), dr inż. Marek HOŁYŃSKI,  
doc. dr hab. inż. Roman KULESZA (redaktor naczelny) mgr inż. Jerzy  
MYSIOR, mgr inż. Józef SZMYD, mgr Robert ZAJĄC

Redaktor techniczny: Maria KOZŁOWSKA

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa  
tel. 28-37-29 lub 21-84-41 w. 431



P. 3057/77

# Biuletyn Informacyjny OBIEKTOWE SYSTEMY KOMPUTEROWE

Rok XVI

Nr 5-6

1977

## Komputeryzacja projektowania inżynierskiego

### Spis treści

### Contents

Roman KULESZA - Automatyzoacja projektowania maszyn . . . . s. 5

Stanisława BONKOWICZ-SITTAUER, Janina OLECH - Niektóre problemy małych banków danych dla komputerowo wspomaganego projektowania . . . . . s. 29

Adam LUTOBORSKI - Systemy metody elementu skończonego - podstawy matematyczne . . . . s. 41

Adam LUTOBORSKI, Ryszard PAWLIK - Systemy metody elementu skończonego - analiza systemów istniejących i koncepcja systemu wzorcowego . . . . . s. 51

Jerzy MOCALA - Zagadnienie modelu procesu projektowania w komputeryzacji projektowania technicznego . . . . . s. 61

Ryszard PAWLIK - Podstawowe wymagania stawiane programom konwersacyjnym . . . . . s. 75

Ryszard PAWLIK, Urszula ORŁOWSKA - Metoda projektowania dużych programów na przykładzie programu do projektowania pewnego mechanizmu płaskiego . . . . s. 83

Wojciech MOKRZYCKI - Algorytm i ocena dokładności układowych generacji krzywych 2-stopnia na siatkach kwadratowych . . . s. 89

Danuta KRYŃSKA - Analiza wybranych metod numerycznego całkowania układów równań różniczkowych zwyczajnych I rzędu, stosowanych w obliczeniach inżynierskich i naukowych . . . . s. 111

Izabella BOBIŃSKA - Analiza pewnych szeregów nieskończonych stosowanych w obliczeniach inżynierskich . . . . . s. 123

Z cyklu: Programowanie obrabiarek

Lidia JESIONOWSKA - Interaktywne systemy wspomagające wytwarzanie programów dla obrabiarek sterowanych numerycznie (OSN) . . . . . s. 129

Роман КУЛЕША - Автоматизация проектирования машин . . . . c. 5

Станислава БОНКОВИЧ-СИТТАУЭР, Янина ОЛЕХ - Некоторые проблемы малых банков данных машинного проектирования . . . . c. 29

Адам ЛУТОБОРСКИ - Системы метода конечного элемента - математические основы . . . . c. 41

Адам ЛУТОБОРСКИ, Рyszard ПАВЛИК - Системы метода конечного элемента. Анализ существующих систем и концепция эталонной системы . . . . . c. 51

Ежи МОЦАЛА - Проблема модели процесса проектирования в компьютеризации технического проектирования . . . . . c. 61

Рyszard ПАВЛИК - Основные требования, предъявляемые к диалоговым программам . . . . . c. 75

Рyszard ПАВЛИК, Уршуля ОРЛОВСКА - Метод проектирования больших программ на примере проектирования некоторого плоского механизма . . . . . c. 83

Войцех МОКРZYCKИ - Алгоритмы и оценка точности схемных генераций кривых 2-степени на квадратных решетках . . . . c. 89

Данута КРЬНЬСКА - Анализ избранных методов числового интегрирования систем обычных дифференциальных уравнений 1 степени применяемых при инженерских и научных вычислениях . . . c. 111

Изабелла БОБИЊСКА - Анализ некоторых бесконечных рядов, применяемых при инженерских вычислениях . . . . . c. 123

Из цикла: Программирование обрабатывающих станков

Roman KULESZA - Automation of machine design . . . . . p. 5

Stanisława BONKOWICZ-SITTAUER, Janina OLECH - Some small data banks problems of computer-aided design . . . . . p. 29

Adam LUTOBORSKI - Systems of a finite element method - mathematical bases . . . . . p. 41

Adam LUTOBORSKI, Ryszard PAWLIK - Systems of a finite element method. Analysis of existing systems and a concept of the principal system . . . . p. 51

Jerzy MOCALA - Problem of design process model in technical design computerization . . p. 61

Ryszard PAWLIK - Fundamental requirements for conversational programs . . . . . p. 75

Ryszard PAWLIK, Urszula ORŁOWSKA - A method of large programs design with the example of a flat mechanism . . . . . p. 83

Wojciech MOKRZYCKI - Algorithms and the accuracy evaluation of the arway generation of the 2nd order curves on the square nets . . . . . p. 89

Danuta KRYŃSKA - The selected methods analysis of numerical integration of the 1st order differential equations used for engineering and scientific calculations . . . . . p. 111

Izabella BOBIŃSKA - Analysis of some infinite series used for engineering calculations p. 123

From the cycle: Machine tool design

Lidia JESIONOWSKA - Interactive systems supporting program producing for computer controlled machine tools . . . . . p. 129

Лидия ЕСИОНОВСКА - Интерактивные системы, применяемые при разработке программ для обрабатывающих станков с числовым программным управлением . . . . c. 129



Od Redakcji

Prezentowany czytelnikom kolejny numer "Biuletynu Informacyjnego OSK" (5/6 1977) - jest w zasadzie numerem monotematycznym, poświęconym problemom komputerowo wspomaganego projektowania (KWP); czyli temu co w literaturze zagranicznej jest rozumiane pod pojęciem CAD (computer aided design). Kompleks zagadnień związanych z automatyzacją, czy też komputeryzacją projektowania jest niezmiernie szeroki, nawet jeżeli ograniczymy go do tzw. projektowania inżynierskiego.

Biorąc pod uwagę profil "Biuletynu Informacyjnego OSK" wydawanego przez Instytut Maszyn Matematycznych, oczywiste jest, że na problematykę KWP będziemy patrzeć od strony środków sprzętowo-programowych umożliwiających i jednocześnie warunkujących realizację podstawowych założeń koncepcji projektowania wspomaganego komputerem.

Bez wątpienia zawarte w tym numerze BIOSK materiały nie wyczerpują nawet tak wąsko nakreślonych ram tematycznych. Większość autorów artykułów wchodzących w skład niniejszego numeru koncentruje swoją uwagę na problemach oprogramowania specjalistycznego-użytkowego. Wobec powyższego Redakcja uważa, iż numerem tym jedynie rozpoczyna cykl pod wspólnym tytułem: "KOMPUTERYZACJA PROJEKTOWANIA INŻYNIERSKIEGO" i przewiduje, że w przyszłości artykuły dotyczące problemów KWP, będą publikowane pod tym właśnie wspólnym tytułem. Redakcja serdecznie zaprasza do zgłaszania artykułów związanych z tematyką KWP, które w miarę możliwości będą publikowane.



doc.dr hab.inż. Roman KULESZA  
Instytut Maszyn Matematycznych

## Automatyzacja projektowania maszyn

### 1. Wprowadzenie

Wynikiem imponującego rozwoju teorii i praktyki budowy maszyn jest dźwignia prosta, silnik elektryczny, wiertarka, maszyna matematyczna, programowany robot przemysłowy, a w ostatnich latach również komputerowo sterowane linie technologiczno-produkcyjne, czy też tak zwane centra produkcyjne.

Wszystkie te maszyny powstały jako wynik twórczej pracy, myśli konstruktorskiej wybiegającej naprzód i wyprzedzającej nowe potrzeby człowieka, a korzystającej z istniejących możliwości realizacyjnych wynikających z osiągnięć nauki i techniki.

Praca konstruktora czy też zespołu projektantów i budowniczych maszyny była zawsze jednakowo trudna i pasjonująca, pełna przeciwności i satysfakcji. Byłoby brakiem obiektywizmu twierdzenie, że konstruowanie najnowocześniejszych maszyn stawia przed ich budowniczymi wyjątkowo trudne zadanie. Owszem, realizacja takiego zadania wymaga odpowiedniego, dostosowanego do danego problemu zmieniania warunków i metod pracy konstruktorów, traktowania więc między innymi procesu konstruowania i budowania maszyny jako bardziej niż dotychczas złożonego przedsięwzięcia intelektualnego i techniczno-organizacyjnego, lecz za to wspomaganego dostępnymi dopiero dziś metodami i środkami wynikającymi z osiągnięć nauki i techniki. To oczywiste - taka jest dialektyka rozwoju nauki i techniki. Byłoby też nadmiarem optymizmu twierdzić, że użycie najnowszych środków projektowania zmniejsza rolę konstruktora.

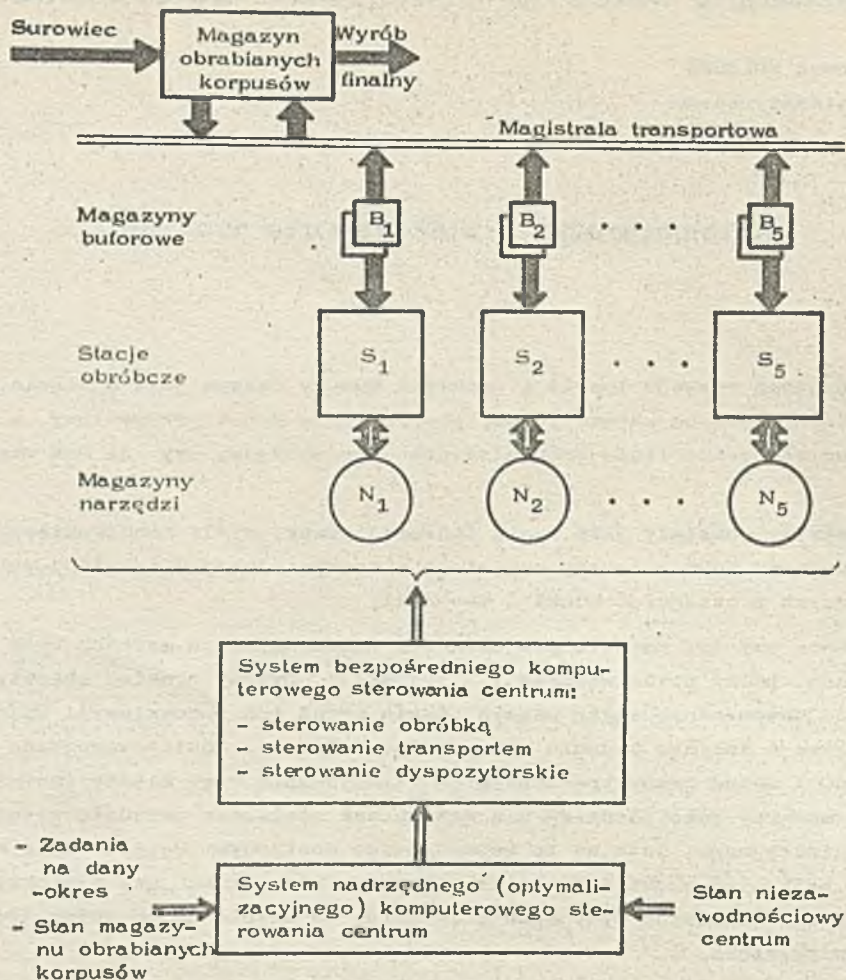
Wysiłki w celu pozyskania nowych metod i środków wspomagających proces budowania maszyn dotyczą oczywiście wszystkich faz tego procesu, w którym automatyzacja projektowania (na obecnym etapie rozwoju przy użyciu maszyn matematycznych i stąd zwana komputerowo wspomaganym projektowaniem) ma szczególnie istotne znaczenie.

Istotę automatyzacji projektowania, potrzebę, charakterystykę jej rozwoju i zastosowań oraz wpływ na rozwój innych dyscyplin łatwiej będzie naświetlić na tle problemów występujących w procesie projektowania i budowania nowoczesnych maszyn.

Rozpatrzmy centrum produkcyjne KOR-1, zaprojektowane i zbudowane przez Centrum Badawczo-Konstrukcyjne Obrabiarek w Pruszkowie w współpracy z Instytutem Maszyn Matematycznych Zjednoczenia NERA.

Centrum produkcyjne KOR-1 jest zautomatyzowaną, komputerowo sterowaną linią technologiczno-produkcyjną, przeznaczoną do jednoczesnej obróbki skrawaniem wielu, nie koniecznie identycznych przedmiotów korpusowych, wymagających z kolei wielu - ściślej setek a nawet tysięcy, również nie koniecznie jednorodnych, operacji technologicznych. Uproszczony schemat ideowo-funkcjonalny centrum przedstawiony jest na rys. 1. Centrum składa się z pięciu stacji obróbczych ( $S_1, S_2, \dots, S_5$ ) połączonych magistralą transportową umożliwiającą przemieszczanie palet z obrabianymi korpusami, w dowolnej kolejności między poszczególnymi stacjami oraz załadowywanie i rozładowywanie palet z i do magazynu obrabianych korpusów. Każda stacja obróbcza ma możliwość automatycznego wymienia palet z obrabianymi korpusami między wózkiem magistrali transportowej, dwumiejscowym magazynem buforowym  $B_i$  a imadłem stacji obróbczej  $S_i$  ( $i=1,2,\dots,5$ ) oraz automatycznego wymienia narzędzia na jedno z trzydziestu różnych narzędzi znajdujących się w magazynie  $N_i$  danej stacji.

Powyższe wskazuje z pewnością na fakt, iż jest to maszyna, której projektowanie wymaga potraktowania jej jako systemu. Koncepcja takiej maszyny wymaga zweryfikowania, i to zarówno z punktu widzenia funkcjonalnej struktury jak i struktury niezawodnościowej, systemu sterowania, metod profilaktyki, diagnostyki i tym podobnych punktów widzenia. Jest rzeczą pewną, iż pomimo wysoko roz-

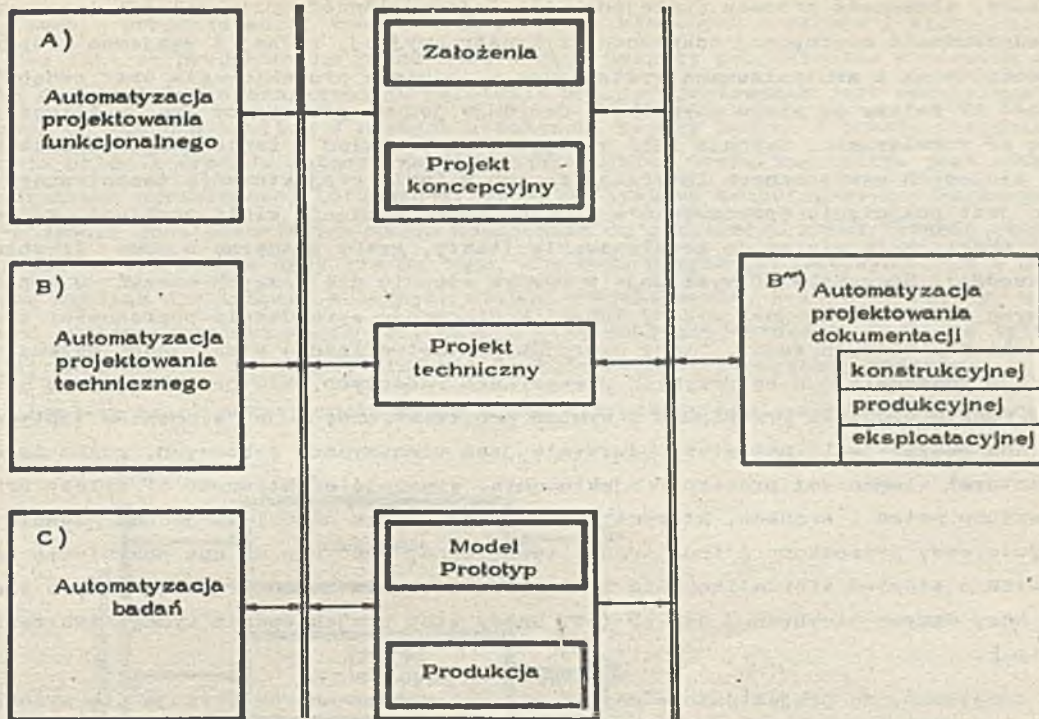


Rys. 1. Uproszczony schemat ideowo-funkcjonalny centrum produkcyjnego KOR-1

winiętych metod i narzędzi teorii systemów i analizy matematycznej oraz teorii budowy maszyn nie będzie można zbudować adekwatnych modeli matematycznych opisujących zjawiska nas interesujące przy badaniu koncepcji tej maszyny. Musimy korzystać w tej sytuacji z modelowania maszyny jako systemu za pomocą maszyny matematycznej. Podobnie projektowanie koncepcyjne i odpowiednio konstrukcyjne (techniczne) elementów tej maszyny lub różnych jej podsystemów (transportowego, sterowania komputerowego itp.) będzie również wymagało modelowania i w określonym sensie optymalnego rozwiązania. W tym zadaniu (ściślej wielu zadaniach) system komputerowo wspomaganego projektowania powinien pomóc w obliczeniach, powinien "podpowiedzieć" jakie są znane rozwiązania podobne do proponowanych, które z rozwiązań konstrukcyjnych są dopuszczalne, a które nie - ze względu na obowiązujące normy, standardy, technologie itp. W projektowaniu takiej maszyny będą brali udział specjaliści z różnych, nawet bardzo odległych dziedzin wiedzy. Konstrukcja będzie wynikiem ich kolektywnego projektowania. Czy komputer może pomóc i jak, aby wynik tego działania był możliwie optymalny? Nie ma potrzeby rozwijania dalej problemów, które powstają w różnych fazach projektowania takiej maszyny. Zauważmy tylko, że zaistnieje również konieczność opracowania systemu operacyjnego minikomputerowego systemu sterowania, a więc jako element konstrukcyjny maszyny pojawi się także "software".

Rola automatyzacji (komputeryzacji) projektowania maszyn jest różna na różnych etapach projektowania. Ilustruje to w dużym skrócie rys. 2. Mówi się o automatyzacji projektowania (AP) funkcjonalnego (konceptualnego) mając na myśli głównie wspomaganie prac projektowych związanych z analizowaniem, opracowywaniem i weryfikowaniem założeń na maszynę i opracowywaniem projektu koncepcyjnego. W tej fazie automatyzacja projektowania (AP) ma jeszcze ograniczone zastosowanie praktycznie. Wynika to z braku formalnych podstaw do automatyzowania procesu syntezy. Rozwój metod i





Rys. 2. Rola automatyzacji projektowania i badań w procesie projektowania maszyn

narzędzi, a szczególnie metod analizy, środków do modelowania procesów przebiegających w maszynie i coraz większe powiązanie informacyjne i metodyczne tej fazy komputerowo-wspomaganej AP (KWAP) z fazą opracowywania projektu technicznego przyczynia się do praktycznego wzrostu rangi KWAP w tej fazie. Wyróżniający się pod tym względem przykład stanowi KWAP maszyn matematycznych. Warto zauważyć, że wiele metod opanowanych w tym zakresie może być adaptowanych do projektowania conceptualnego innych maszyn. Uwaga ta dotyczy zresztą i innych faz projektowania. Myśl tę rozwinie my nieco dalej. Na etapie opracowywania projektu technicznego maszyny komputer ma za zadanie zarówno wspomagać prace projektowe (co z reguły jest realizowane za pomocą pakietów programów problemowo zorientowanych) jak i przejąć na siebie znaczną część zrutynizowanych prac związanych z wykonaniem dokumentacji (konstrukcyjnej, produkcyjnej i eksploatacyjnej). Istnieje już wiele praktycznie działających systemów AP, realizujących te zadania pomyślnie. Systemy takie dysponują z reguły obszernymi bazami danych o charakterze konstrukcyjno-technologicznym i są budowane przy uwzględnieniu poziomu oraz sposobu zautomatyzowania procesów produkcyjno-technologicznych w zakładach, w których projektowana maszyna będzie produkowana lub/i stosowana. Oddzielną poniekąd grupę tematyczną stanowi automatyzacja badań modeli, prototypów czy też maszyn pochodzących z produkcji seryjnej. Problemów tych nie będziemy w niniejszej pracy bliżej rozpatrywać. Zauważmy tylko, że komputery mają w tym zakresie już ugruntowaną pozycję i że rozwój metod i środków AP maszyn wywiera korzystny wpływ na zadania, będące do zrealizowania na etapie badań modeli bądź prototypów (istotne skrócenie czasu takich badań i zmniejszenie poniesionych kosztów). Z drugiej strony rozwój techniki komputerowej, a szczególnie minikomputerowych systemów kontrolno-pomiarowych stworzył nową "jakość" dla automatyzowania badań maszyn i to zarówno w zakresie estymowania parametrów konstrukcyjnych i stosowania dynamicznej statystycznej kontroli jakości, jak i w zakresie możliwości nowego podejścia do eksploataowania maszyn (diagnostyka techniczna, optymalizacja wymian profilaktycznych wywołanych określoną wymianą poawaryjną itp.).

Oczywiście rola, znaczenie, efektywność i tym podobne cechy AP maszyn są różne dla różnych maszyn i różnych faz ich projektowania. Rozpatrzmy to w dużym skrócie. Podatność na automatyzację projektowania conceptualnego zależy oczywiście od stopnia możliwości sformalizowania procesu syntezy projektowanej maszyny. Natomiast podatność na automatyzację projektowania konstrukcyjno-technicznego zależy od takich czynników jak: jednorodność bazy elementowo-konstrukcyjnej i technologicznej, stopień standaryzacji i unifikacji elemen-

tów tej bazy, złożoność procesu funkcjonowania i jednorodność opisu relacji tego procesu, objętość i jednorodność niezbędnej dokumentacji konstrukcyjnej, rodzaj i wzajemna proporcja między pracą conceptualną i zrutynizowaną występującą w procesie projektowania oraz podobnych czynników. Niezbędność AP zależy od wielu czynników, decyduje jednak praktyczny brak możliwości innego niż za pomocą AP rozwiązania. Czynnikiem takim występuje na przykład w fazie projektowania koncepcyjnego maszyn o złożonych wewnętrznych interakcjach lub w fazie projektowania technicznego wówczas, gdy konieczno jest pominięcie opracowywania modelu. Można wymienić wiele problemów projektowych, które bez AP praktycznie nie są do zrealizowania (testy, grafy planarne o dużej liczbie wierzchołków i krawędzi). Specyfika AP występuje w różnym stopniu dla różnych maszyn: w wypadku maszyn matematycznych tą specyfiką jest między innymi konieczność sprawdzenia poprawności realizowania bardzo dużej liczby różnych zadań. Stany awaryjne (katastroficzne) w maszynach matematycznych nie mają takiego znaczenia jak na przykład w maszynach roboczych. Nie projektuje się maszyny matematycznej z punktu widzenia przeglądów i wymian profilaktycznych jej elementów (optymalną strategią jest wymiana powaryjna), natomiast odwrotnie jest w maszynach roboczych, gdzie tego typu problemy są istotnymi elementami procesu projektowania. Wreszcie efektywność AP zależy przede wszystkim od poziomu metod i środków, którymi się posługuje. Nie chodzi tu jednak głównie o silne systemy komputerowe, grafoskopy i inne środki techniczne, choć nie są one oczywiście bez znaczenia, lecz również o stopień sformalizowania i zalgorytmizowania zadań projektowych, o stopień zintegrowania bazy danych niezbędnej dla AP (czy każdy etap projektowania wymaga tworzenia nowej bazy danych itp.).

Warto odnotować, że projektowanie maszyn matematycznych charakteryzuje się wysoką podatnością na AP, niezbędną i efektywną. Ma również swoją specyfikę, która nie wyklucza jednak możliwości przenoszenia metod i środków opracowanych dla potrzeb AP maszyn matematycznych na AP innego rodzaju maszyn. Można nawet wskazać na liczne analogie problemów występujących przy AP EMC i AP innych maszyn. Podstawowym jednak znaczeniem w rozwoju metod i środków AP EMC dla rozwoju AP innych maszyn ma zdobycie doświadczenia w organizowaniu i wdrażaniu systemu AP.

Automatyzacja projektowania maszyn jest dziedziną młodą i ma przed sobą dużą przyszłość. Powszeczenie uważa się, że jest ona konsekwencją automatyzacji procesów wytwarzania oraz zarządzania i jest niezbędnym ich uzupełnieniem. Automatyzacja projektowania maszyn musi powodować i powoduje wiele zmian (zjawisk) o charakterze gospodarczym, profesjonalnym, socjalnym i oświatowym. Mają one szersze i głębsze znaczenie niż tylko zwiększenie techniczno-ekonomicznej efektywności procesów projektowania nowych maszyn i wytwarzania za pomocą tych maszyn nowych dóbr materialnych. Automatyzacja projektowania przybliża projektanta do rzeczywistości, pozwala na lepsze gromadzenie doświadczeń, ułatwia korzystanie z dorobku innych konstruktorów. Tkwi w niej też nadzieja na zmniejszenie dystansu lub przynajmniej na zapobieżenie jego powiększaniu się między potrzebami rewolucji naukowo-technicznej a indywidualnymi zdolnościami człowieka.

## 2. Automatyzacja projektowania maszyn na etapie makroprojektowania

Pod pojęciem makroprojektowania maszyny pojmujemy pewną procedurę, której elementami są:

- założenia,
- projekt koncepcyjny maszyny,
- dekompozycja maszyny na elementy,
- opracowanie założeń na elementy,
- opracowanie koncepcji rozwiązania elementów,
- modelowanie własności funkcjonalnych i eksploatacyjnych elementów,
- agregacja wyników modelowania elementów oraz
- modelowanie maszyny jako całości i to w różnych aspektach badania jej właściwości.

Procedura ta ma oczywiście różne pętle zmieniające się wraz z kolejnymi cyklami iteracyjnymi i powinna być zbieżna do optymalnego projektu koncepcyjnego.

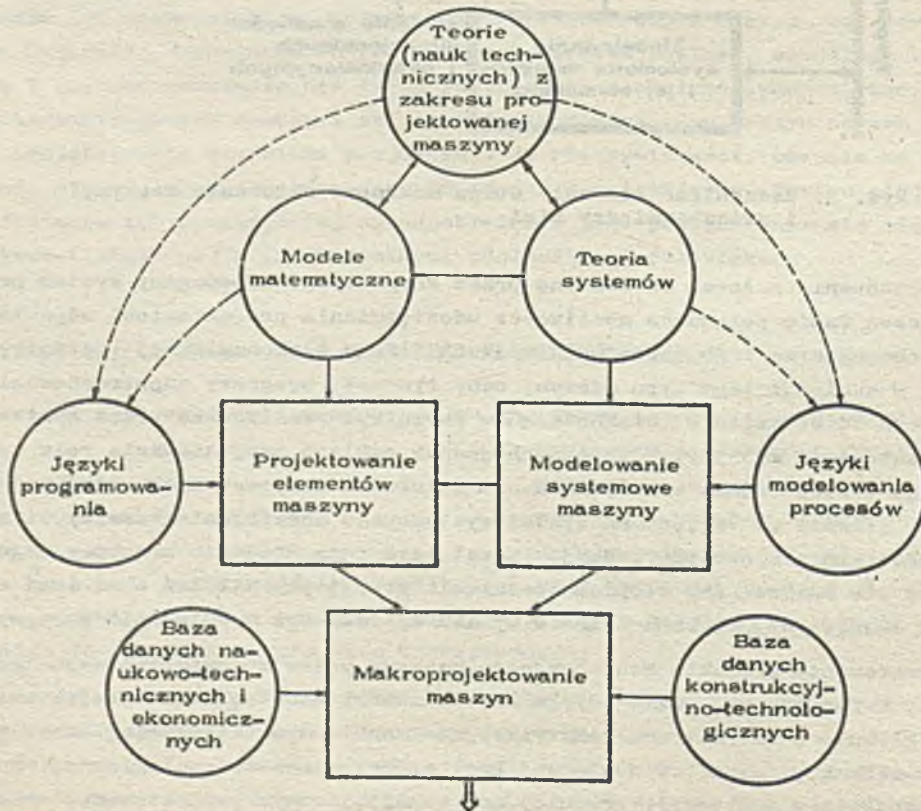
Etap makroprojektowania jest najbardziej conceptualnym etapem całości projektowania maszyny. Z punktu widzenia udziału w kosztach pełnego cyklu projektowania maszyny, etap ten nie jest specjalnie drogi, lecz z punktu widzenia strat, które w pełnym cyklu projektowania i wdrożenia maszyny do produkcji, mogą wynikać z zaniedbania lub zbyt dużego uproszczenia, istotnie decyduje o sukcesie projektowanej maszyny. Jest to szczególnie słuszne w odniesieniu do nowoczesnych maszyn, wy-



projektowania - bazy danych. Jednym z głównych, a zarazem bardzo trudnych do zalgorytmizowania zadań projektu koncepcyjnego jest synteza maszyny. Znacznie upraszczając można powiedzieć, że warunkiem koniecznym do zalgorytmizowania zadania syntezy jest możliwość formalnego wyrażenia związków między zadaniami funkcjonalnymi maszyny a różnymi możliwymi do zrealizowania wariantami procesów zdarzeń przyczynowo-skutkowych, opisanych za pomocą określonego zbioru operatorów. Celem syntezy jest wybranie - w określonym sensie - najkorzystniejszego procesu z możliwych do zrealizowania. W wielu wypadkach zadanie syntezy można opisać formalnie w postaci modelu matematycznego i rozwiązywać w sposób komputerowo wspomagany. Przykładem tego są chociażby liczne problemy dające się dobrze sformułować w kategoriach programowania matematycznego lub dualne do niego. Tak czy inaczej, zadanie to jest z reguły realizowane na podstawie osobistych koncepcji projektantów i dekompozycji. Tym większa staje się rola modelowania systemowego maszyny jako weryfikatora tych koncepcji i poprawności "rozdekomponowania" założeń na poszczególne elementy maszyny. Projekty koncepcyjne elementów maszyny mogą oczywiście mieć analogicznie do powyższych procedury postępowania wraz z ich dalszym dekomponowaniem. Zaletą przedstawionego postępowania jest między innymi możliwość posłużenia się przy opracowywaniu projektów koncepcyjnych i konstrukcyjnych elementów maszyny, wyspecjalizowanymi czy też problemowo zorientowanymi pakietami programów. Jednak z powyższych chociażby względów, większość projektów koncepcyjnych poszczególnych elementów wymaga zweryfikowania przez modelowanie za pomocą EMC.

Tak więc modelowanie systemowe maszyny i jej elementów za pomocą EMC jest jednym z najistotniejszych środków etapu makroprojektowania maszyny. Oczywiście modelowanie systemowe maszyny powinno być wykonane z różnych, zależnie od potrzeb, punktów widzenia, takich jak badanie struktury funkcjonalnej, struktury niezawodnościowej, wpływu procesu eksploatacji maszyny itp. Zauważamy, że o modelowaniu systemowym mówimy w tej pracy głównie w sensie modelowania za pomocą cyfrowych maszyn matematycznych, a nie na przykład za pomocą analogowych czy też hybrydowych maszyn matematycznych. Nie bierzemy też zapewne w dostatecznym stopniu pod uwagę innych technik modelowania systemowego, a szczególnie makietowania.

Na rys. 4 przedstawiono w dużym uproszczeniu elementy "zasobu metodyczno-informacyjnego", stanowiącego bazę dla makroprojektowania maszyn i zasadnicze związki między nimi.



Rys. 4. Elementy "Zasobu metodyczno-informacyjnego" stanowiącego bazę dla makroprojektowania maszyn i zasadnicze związki między nimi

Warto może zauważyć, że o ile przy projektowaniu elementów maszyn korzysta się w zasadzie z modeli matematycznych dobrze opisujących się w kategoriach programowania matematycznego, to modelowanie systemowe wymaga z reguły ujęcia w kategoriach teorii systemów, takich na przykład jak: teoria automatów skończonych, teoria obsługi masowej, teoria badań operacyjnych, teoria grafów itp. W pierwszym wypadku problemy są rozwiązywane za pomocą programów biblioteki standardowych programów numerycznych bądź wyspecjalizowanych pakietów programów, bądź też są pisane w językach algorytmicznych zorientowanych na opisywanie już "dobrze" sformułowanego w sensie matematycznym problemu. Takimi językami są przeważnie algorytmiczne języki algebraiczne, przeznaczone głównie do zapisywania algorytmów numerycznego przetwarzania informacji, jak na przykład ALGOL, FORTRAN, PL/1. Maszyna matematyczna w tym wypadku realizuje program głównie w sensie obliczeniowym. W innej sytuacji zadaniem maszyny matematycznej jest również (w różnym stopniu) "imitowanie" bądź "symulowanie" maszyny badanej jako system i wówczas język modelowania spełnia dodatkowe funkcje formalnego modelowania problemu. Dlatego też języki modelowania są budowane na bazie języków algorytmicznych, stosowanych do obliczeń naukowo-technicznych, lecz z reguły zawierają własne formalne środki niezbędne do opisu badanego systemu.

Ze względu na różnorodność zjawisk występujących w modelowaniu systemów, języki modelowania stanowią liczną grupę. Zasadniczy podział obejmuje języki do modelowania procesów dyskretnych, ciągłych i ciągle-dyskretnych.

Języki do modelowania procesów ciągłych powstały właściwie jako pochodne od technik programowania analogowych maszyn matematycznych, czy tak zwanych analizatorów równań różniczkowych. Na przykład takie języki jak PACTOLUS czy CSMP mają środki programowe wręcz analogiczne do stosowanych w technice maszyn analogowych, a więc wzmacniacze operacyjne, bloki całkowania, sumatory, invertory i tym podobne. Program modelowania systemu opisuje w tych kategoriach poszczególne elementy systemu i sieć współdziałania między nimi. Języki takie jak DSL-90, System/360 CSMP opisują modelowany system wprost w kategoriach układów zwyczajnych równań różniczkowych. Wadą tego typu języków modelowania procesów ciągłych jest słaby stopień odzwierciedlenia rzeczywistego przebiegu procesu w systemie. Do znanych języków modelowania procesów ciągłych należą również języki MIDAS i CSSL. Tak czy inaczej, istotą wszystkich języków tej klasy jest numeryczne całkowanie z określonym krokiem całkowania, układów zwyczajnych równań różniczkowych opisujących system. Języki te nie są językami proceduralnymi. Zakres zastosowań maszyn matematycznych do modelowania procesów ciągłych w projektowaniu maszyn jest dość duży i odgrywa istotną rolę. Obejmuje on szczególnie modelowanie układów automatycznej regulacji i układów napędowych. Oczywiście mamy na myśli układy ciągle. Zwrócić też należy uwagę, że udział tych układów z biegiem czasu systematycznie maleje. Należy także wspomnieć o możliwości modelowania procesów ciągłych za pomocą maszyn hybrydowych oraz realizowaniu za pomocą modelowania quasi-naturalnego eksperymentu.

Pod pojęciem procesu dyskretnego pojmujemy proces, w którym zarówno przebieg zdarzeń jak i ich przyszłe skutki, można adekwatnie opisać przez podanie stanów w określonych (nie koniecznie ustalonych) chwilach.

Zauważmy, że z pewnym przybliżeniem każdy proces ciągły może być traktowany jako dyskretny (w sensie skończoności przedziałów, na których określone są odpowiednio funkcje). Warto dodać, że odwrócenie tej metody - jakkolwiek bardzo rozpowszechnione - jest ryzykowne. Uwaga ta byłaby banalna, gdyby nie uzasadniona wieloma względami dążność do przybliżonego modelowania procesów ciągłych za pomocą aparatu modelowania procesów dyskretnych lub ciągle-dyskretnych. Można podać wiele takich przykładów, wymagających modelowania projektowanej maszyny jako systemu, w których zachodzące zjawiska interakcyjne mają charakter procesu dyskretnego bądź też dyskretność wynika z charakteru zmiennych decyzyjnych. W centrum obróbki korpusów, o którym poprzednio była już mowa, w pewnych chwilach pojawiają się nowe obiekty (na przykład odlewy korpusów wchodzące do obróbki) oraz w pewnych chwilach znikają z centrum obiekty (korpusy, których obróbkę zakończono lub przerwano z jakichś względów). Podobnie mogą z centrum "ubywać" lub do centrum "przybywać" niektóre elementy robocze, takie jak poszczególne maszyny, narzędzia itp. Chwile tych zdarzeń muszą być z reguły traktowane jako zmienne losowe (losowy czas obróbki korpusu na określonym stanowisku roboczym, losowy czas naprawiania uszkodzonej maszyny i in.). Każdy zabieg technologiczny jest zdarzeniem, w wyniku którego obiekt poddany temu zabiegowi przyswoił sobie nowe cechy (atry-

buty). Zdarzenia z reguły powodują możliwość zrealizowania się innych zdarzeń w przyszłości. Zbiór takich możliwych zdarzeń (wygenerowanych przez określone zdarzenie) może powiększać się lub pomniejszać w zależności od innych zdarzeń, które nastąpiły po tym zdarzeniu. Na przykład, jeśli z pownych względów jakiegoś zabiegu technologicznego nie wykonano na określonym stanowisku roboczym, to może on być dokończony na innych, które dysponują niezbędnymi do zrealizowania tego zabiegu narzędziami w swoich zasobnikach. Jeśli jednak w czasie transportowania tego obiektu, z zasobników narzędziowych niektórych stanowisk roboczych usunięto te narzędzia, to zbiór możliwych do kontynuowania zabiegu stanowisk odpowiednio się pomniejszył.

Dla każdego określonego zabiegu technologicznego opracowany jest i przechowywany w pamięci zewnętrznej systemu minikomputerowego sterowania centrum - odpowiedni program. Program ten jest identyczny dla różnych stanowisk roboczych, a "przyswojenie" tego programu przez określone stanowisko, w określonej chwili jest skutkiem określonych zdarzeń. Z systemu minikomputerowego sterowania wysyłane są różnego typu instrukcje do różnych odbiorców (stanowisk obróbczych i linii transportowej) oraz odwrotnie - przez ten system są przyjmowane od tych "odbiorców" określone informacje. Łączność może w określonej chwili odbywać się tylko z jednym "abonentem". Jeśli "linia" jest zajęta inni "abonenci" muszą czekać. Są jednak zdarzenia, których zrealizowanie się przerywa zajętość linii, co umożliwia przesłanie do systemu sterowania informacji o tym zdarzeniu.

Powyższe przykłady ilustrują charakter środków programowych, jakimi powinien dysponować język przy modelowaniu procesów dyskretnych. W aparacie matematycznym, właściwym takim teoriom jak teoria automatów skończonych, teoria grafów, teoria obsługi masowej, teoria procesów stochastycznych i in., istnieje wprawdzie wiele środków formalnych zdolnych do opisywania zjawisk właściwych modelowaniu procesów dyskretnych, jednak środki te nie są wystarczające. Dlatego więc języki modelowania procesów dyskretnych budują się zasadniczo na bazie uniwersalnych algorytmicznych języków algebraicznych, takich jak ALGOL, FORTRAN i PL/1, wprowadzając wiele nowych środków programistycznych (lub odpowiednio modyfikując istniejące) takich jak np. operatory planowania zdarzeń, operatory generowania obiektów, kalendarze (zdarzeń), zmodyfikowane operatory przyswajania, różnego rodzaju operatory warunkowe, środki do "aktywowania" i "pasywacji" procesu i tym podobne. Powyższe środki sprawiają, że języki modelowania procesów dyskretnych są tak zwanymi językami proceduralnymi, to jest takimi, które pozwalają na "rzeczywiste" odwzorowanie procesu. Są też przez to z reguły "mocniejsze" niż języki algorytmiczne, na których bazują i mają szersze możliwości operowania rodzajami danych (stosowanie listowych struktur danych) oraz bogatsze własności realizowania procesu przetwarzania informacji (współbieżność lub quasi-współbieżność realizacji programu).

Do znanych uniwersalnych języków modelowania procesów dyskretnych należą: GPSS, SOL, SIMSCRIPT, SLANG, SIMULA oraz SIMULA-67. Zbiór znanych, specjalizowanych języków modelowania procesów dyskretnych lub ciągle-dyskretnych liczy około kilkudziesięciu języków problemowo zorientowanych, wśród których wyróżniają się szczególnie języki do modelowania struktur cyfrowych (układów i sieci liczących, sieci teletransmisji cyfrowej, maszyn komunikacyjnych, cyfrowych sieci sterowania i in.). Początkowo języki modelowania procesów dyskretnych, takie jak np. GPSS czy też SOL były semantycznie bardzo złożone. Rozwój tych języków wykazał jednak, że istnieje zbiór elementarnych środków programowych wystarczający do dostatecznie dokładnego odwzorowania przebiegu procesu w modelowanym systemie przy jednoczesnym pełnym zachowaniu konceptualnej specyfiki tych języków. Główna zasada modelowania procesów dyskretnych (za pomocą języków modelowania dyskretnego), w dużym uproszczeniu, jest następująca: każde z możliwych zdarzeń opisuje się odpowiednim algorytmem (przy użyciu między innymi operatora przyswajania). Skutkiem określonego zdarzenia po upływie czasu  $\Delta t$  jest wystąpienie zdarzenia klasy E. Para  $E, \Delta t$  ma określoną miarę i jest przechowywana w zbiorze K zwanym "kalendarzem" (aktualna zawartość zbioru K generowana jest przez operatory zdarzeń). Każdorazowo ze zbioru K wybiera się parę  $(E_i, t_i)$  o minimalnej wartości  $t_i$  i wykonuje się algorytm zdarzenia  $E_i$ , przypisując mu wartość chwilową  $T + t_i$  (T - czas modelowany). Oczywiście może zdarzyć się, że kilka elementów zbioru K ma tę samą wartość uniwersalną i wówczas proces zrealizowany jest współbieżnie bądź kolejno dla tych elementów z "zatrzymaniem" czasu T. Modelowany proces jest w pełni określony, jeśli zbiór K jest zbiorem pustym dla  $T = 0$  oraz dano jest zdarzenie inicjujące. Oczywiście pełne określenie

modelowanego procesu napotyka w praktyce na liczne trudności. Nie wdając się w szczegóły wynikających stąd problemów przypomnijmy tylko, że z reguły modelowane systemy muszą być traktowane jako inercyjno procesy wzajemnie stochastycznie zależne, a celem modelowania jest między innymi wyestymowanie wartości pewnych parametrów procesów modelujących system, do czego niezbędna jest odpowiednia statystyka. Języki modelowania procesów dyskretnych dysponują z reguły środkami ułatwiającymi generowanie odpowiednich zmiennych losowych, gromadzenie statystyk, uzyskiwanie histogramów procesu i tym podobnymi środkami ułatwiającymi zorganizowanie eksperymentu statystycznego. Stąd też, są w różnym stopniu językami symulacyjnymi, umożliwiającymi badanie modelu metodami symulacyjnymi (metodami Monte Carlo).

Najbardziej uniwersalnym językiem modelowania dyskretnego, zorientowanym jednocześnie na modelowanie symulacyjno jest język SIMULA-67 stanowiący rozszerzenie języka ALGOL. W Polsce kompilator tego języka jest dostępny na maszynach CDC-6000 (Świerk, Kraków) oraz IRIS-80 (MERA-IMM, Warszawa). Doświadczenia uzyskane z symulacji na maszynie IRIS-80 są pozytywne i pozwalają stwierdzić, że kompilator języka SIMULA na tę maszynę nie jest gorszy od kompilatora na maszynę CDC-6000. Jest on opracowany nawet nieco staranniej. W Instytucie Maszyn Matematycznych (MERA-IMM) za pomocą języka SIMULA modelowano centrum obróbki korpusów oraz koncepcję automatycznej linii gwintowników dla Fabryki Wyrobów Precyzyjnych im. Gen. K. Świerczewskiego.

SIMULA jest językiem programowania strukturalnego. Ma wbudowany mechanizm pseudo-równoległości oraz mechanizm zegara (projektant nie musi tego kontrolować). SIMULA ma możliwości rozszerzania się (przez wprowadzenie koncepcji klasy), czyniąc z siebie język zorientowany na specjalistyczne aplikacje. Programy w języku SIMULA są czytelne i łatwe do dokumentowania; programy są modularne. Można je rozbudowywać przez wkładanie oddzielnych modułów. SIMULA ma rozbudowane procedury przetwarzania tekstów i operacji wejścia/wyjścia, ma także systemowe generatory liczb losowych, wreszcie w języku SIMULA można od razu budować model symulacyjny. Nie zdefiniowano natomiast w tym języku wydruków raportów (wyników analiz statystycznych), co występuje na przykład w GPSS. Jednak przez pojęcie klasy można ten brak łatwo wyeliminować.

Metody i środki modelowania dyskretnego i symulacyjnego nie są w Polsce jeszcze dostatecznie popularne. Należy spodziewać się, że ich zastosowanie nie ograniczy się tylko do modelowania tak zwanych "wielkich" systemów (systemów transportowych, telekomunikacyjnych itp.) czy też eksperymentów naukowych (fizyka jądrowa, neurochirurgia, kardiologia), co znalazło odpowiednie potwierdzenie w skali światowej, lecz również rozpowszechni się w projektowaniu nowoczesnych maszyn.

Wśród zalet tych metod i środków należy wymienić następujące:

- umożliwiając pogłębienie wiedzy (przez analizę i eksperyment) o właściwościach modelowanego obiektu wynikających ze skomplikowanych interakcji wewnętrznych;
- spostrzeżenia poczynione przy modelowaniu obiektu mogą być czasem bardziej wartościowe niż wynik eksperymentów zrealizowanych za pomocą modelowania;
- umożliwiają szybkie rozpoznanie dominant badanych zjawisk.

Użyteczność języków do modelowania procesów dyskretnych jest bardzo znaczna dzięki ich jednoczesnej przydatności do badania wielu modeli analitycznych metodami symulacyjnymi (metodami Monte Carlo). Wiadomo bowiem, że wiele problemów wynikających przy projektowaniu maszyn daje się wprawdzie dobrze sformułować w postaci modelu analitycznego, lecz zbadanie tego modelu napotyka na duże trudności o charakterze obliczeniowym, które mogą być w wielu sytuacjach przezwyciężone przez badanie modelu symulacyjnego będącego "analogonem" (w sensie własności estymatorów modelu symulacyjnego) tego modelu analitycznego. Jako przykład może tu służyć nawet rozwiązanie układu algebraicznych równań liniowych (gdy wyznacznik macierzy tego układu ma wartość bliską zeru) lub rozwiązanie układu zwyczajnych równań różniczkowych (gdy liczba zmiennych jest znaczna). Zauważmy, że sytuacje powyższe często występują w praktyce obliczeniowej jako wynik przybliżonego badania własności bardziej złożonego modelu matematycznego (np. typu zagadnień brzegowych, badań funkcjonalów i tym podobnie). Na przykład badanie problemów statyki, dynamiki, termodynamiki, ogólne wytrzymałości konstrukcji metodą elementu skończonego sprowadza się do rozwiązania właśnie układu liniowych równań algebraicznych.

Sytuacja opisana w drugim przykładzie zaczyna w naszym kraju nabierać praktycznego znaczenia i prowadzone są prace badawcze z tego zakresu. Są również pomysły rozwiązania i praktyczne zastosowania odpowiednich systemów komputerowej automatyzacji projektowania, wprawdzie nie w zakresie maszyn lecz mające duże analogie. W obu tych wypadkach próbuje się stosować metody i narzędzia komputerowo wspomaganego wytwarzania odpowiedniego oprogramowania.

Wreszcie trzeci przypadek występuje w Polsce jak dotychczas sporadycznie. Z matematycznego punktu widzenia problemy takie należą do klasy złożonych systemów obsługi masowej i mogą być efektywnie rozwiązywane tylko za pomocą metod symulacyjnych.

Rozpatrzmy z kolei problemy trzeciej grupy, a więc te, które wynikają przy projektowaniu konkretnych zabiegów produkcyjno-technologicznych. Język, w którym opisywane są te problemy, ma przeważnie złożoną strukturę syntaktyczną i semantyczną. Wynika to z istoty opisywanych charakterystyk technicznych oprzyrządowania, maszyn, procesów technologicznych, reguł projektowania i innych parametrów wejściowych dla procesu projektowania. Tradycyjnie były one wyrażane w dużym stopniu językiem naturalnym i w określonej konwencji rysunków i symboli. Rozwój automatyzacji projektowania maszyn stopniowo poprawia sytuację, niemniej przezwyciężenie tej bariery jest stale aktualne, zwłaszcza że wciąż nowe zadania projektowania zabiegów produkcyjno-technologicznych stają się przedmiotem automatyzowania (choćby dlatego, że wciąż nowe stanowiska produkcyjno-technologiczne są automatyzowane i komputerowo sterowane). Nie wdając się w tym miejscu w zbędne szczegóły należy zauważyć, że problem budowania języków sformalizowanych do opisu omawianych problemów wygląda nieco inaczej wówczas, gdy celem automatyzacji projektowania zabiegu produkcyjno-technologicznego jest uzyskanie programu sterującego maszyną, przy czym wygląda nieco inaczej, gdy celem jest uzyskanie programu sterującego dla konkretnej maszyny, a inaczej gdy celem jest uzyskanie takiego programu dla dowolnej numerycznie sterowanej maszyny. Jeszcze inaczej, gdy celem są inne korzyści.

Ostatni z powyższych wypadków występuje w zasadzie wtedy, gdy nie stosuje się numerycznie sterowanych maszyn, a zamiarem jest na przykład zautomatyzowanie wytwarzania określonych fragmentów dokumentacji lub wykonanie określonych obliczeń.

Dwa następne przykłady odnoszą się odpowiednio do tak zwanych języków jedno- i dwuprocesorowych. Oczywiście korzystniejszym rozwiązaniem jest język ostatniego typu, lecz nie zawsze udaje się go zrealizować ze względu na złożoność problemu, do opisu którego służy i parametrów komputera, w którym go się używa. Budowanie języka jest związane z budowaniem modelu opisującego procesy informacyjne w danym projektowaniu na bazie niezbędnego słownika elementarnego. Oczywiście zarówno procesy informacyjne, jak i niezbędne słowniki, są z reguły różne dla różnych rodzajów zabiegów produkcyjno-technologicznych, np. słownik pierwotnych elementów strukturalnych języka do projektowania obróbki skrawaniem powinien zawierać zasady kształtowania powierzchni oparte na zbiorze linii tworzących i sposobach ich przesunięć w przestrzeni. Natomiast język do projektowania mikrostruktur cyfrowych powinien zawierać reguły odwzorowania odpowiednich oddziaływań na strukturę półprzewodnikową własności funkcyjnych logicznych itp. To jest oczywiste.

Powyzsze przyczyny powodują, że języki do opisu procesów produkcyjno-technologicznych są silnie wyspecjalizowane (problemowo zorientowane), chociaż na różnych poziomach szczegółowości opisu (właściwej różnym poziomom swojej hierarchicznej struktury) mają różny stopień wzajemnego podobieństwa. Potwierdza to chociażby fakt istnienia dużej liczby języków do programowania numerycznie sterowanych obrabiarek i odmian tych języków zorientowanych na określone rodzaje obróbki skrawaniem (wiercenie, toczenie, frezowanie, wytaczanie i różne ich kombinacje) wykonywanej z różnym stopniem swobody. Do bardziej znanych języków tej grupy (i odpowiednio ich odmian) należą: APT (AD APT, MINI APT, IF APT), NEL (2PL, 2C, 2CL), AD APT - AUTOSPOT, AUTOPOL, EXAPT (BASIC-EXAPT, EXAPT 1,2), SPS, AUTOPIIT. W Polsce opracowano język APO w trzech wersjach, dostępne są także języki NEL, APT i EXAPT. Języki te są stale udoskonalane, rozwijane (w kierunku np. automatyzowania wyboru narzędzia, zamocowania itp.) bądź wzajemnie łączone (tak powstał AD APT AUTOSPOT oraz EXAPT 1/2 CL). Autorzy języka EXAPT zapowiadają rozbudowę wersji EXAPT-2 i opracowanie języka EXAPT-3. W ZSRR rozwija się wersje języka SPS, w NRD język SYMAP, a w CSRS język AUTOPROG.

Z powyższego wynika jakiego rodzaju problemy powstają w automatyzowaniu procesów produkcyjno-technologicznych, chociażby dla techniki komputerowej i kształcenia konstruktorów i technologów.



Zadania projektowania zabiegów produkcyjno-technologicznych polegają na znalezieniu, w określonym sensie optymalnych, sposobów wykorzystania środków produkcyjno-technologicznych, będących do dyspozycji projektanta (maszyn, narzędzi, urządzeń kontrolno-pomiarowych itp.), zapewniających, z odpowiednim prawdopodobieństwem, uzyskanie zamierzonych (w wyniku zabiegu) rezultatów i in. wyników oraz przy nienaruszaniu różnego rodzaju ograniczeń wynikających z wymagań dotyczących wyrobu, własności przebiegu procesu, parametrów maszyn i narzędzi. Ponadto zadaniem projektanta jest również uzyskanie takiej postaci dokumentacji produkcyjno-technologicznej (opisów, rysunków, programów), która byłaby wysoce użyteczna w procesie produkcji, a więc zrozumiała, ograniczająca do minimum prawdopodobieństwo pomyłek w interpretowaniu, adaptowalna do zmian zachodzących w procesie produkcyjno-technologicznym i umożliwiająca pogłębienie stopnia wnikania w jej szczegóły, w zależności od potrzeby. Wśród licznych metod ukształtowanych wieloletnimi doświadczeniami, stosowanych w projektowaniu procesów produkcyjno-technologicznych występują często metody analityczne, hourystyczne i relacyjne. Automatyzowanie projektowania w procesów wymaga wnikliwego analizowania roli tych metod i umiejętnego ich wykorzystania w ramach znanych środków projektowania. W różnych klasach zabiegów produkcyjno-technologicznych można wydzielić pewne zbiory operacji elementarnych, powtarzających się według pewnych prawideł w różnych, bardziej złożonych zabiegach. Często pozornie różne zabiegi produkcyjno-technologiczne mają atrybuty, rozpoznawanie których daje klucz do znalezienia analogicznych procedur ich projektowania. Pozwala to na uzyskanie wysokiego stopnia zautomatyzowania wielu czynności procesu projektowania.

Narzędziami automatyzacji projektowania procesów produkcyjno-technologicznych w nowoczesnym przemyśle są: kalkulator, minikomputer, terminale zdalnej komunikacji z dużymi maszynami matematycznymi, różnego rodzaju narzędzia i środki modelowania procesów, urządzenia kodowania informacji i wykonywania dokumentacji. Postęp w automatyzowaniu procesów produkcyjno-technologicznych, wyrażający się zastosowaniem

- układów numerycznego sterowania maszynami,
- komputerowych systemów sterowania maszynami typu CNC (computer numerical control) lub DNC (direct numerical control),
- cyfrowo sterowanych automatów montażowych,
- robotów,
- zautomatyzowanych linii transportowych,
- komputerowych systemów kontrolno-pomiarowych i diagnostycznych,

spowodował bezwzględną konieczność zautomatyzowania wielu procesów projektowania zabiegów produkcyjno-technologicznych. Zakres metod i środków, którymi muszą dysponować służby projektujące zabiegi produkcyjno-technologiczne stale się rozszerza. W pracach produkcyjno-technologicznych stosuje się już maszyny, które mają własności adaptacyjne, jak na przykład maszyny samoczynnie dostosowujące szybkość posuwu narzędzia skrawającego do układu sił występujących przy skrawaniu. Nowoczesne roboty wykonują dosyć złożone i precyzyjne operacje na obiektach, które przesuwają się wraz z linią transportową i to z uwzględnieniem przypadkowych zmian prędkości przesuwania się linii itp. Zastosowanie lokalnych układów sterowania (distributed control) nie tylko w maszynach lecz i w ich elementach, możliwe technicznie między innymi dzięki mikroprocesorom i pamięciom półprzewodnikowym bardzo dużej skali integracji, pozwala na tworzenie nowych koncepcji projektowania procesów produkcyjno-technologicznych i poszczególnych zabiegów. Stawia to też przed metodami i środkami automatyzowania tych procesów nowe problemy. W omawianych wyżej przykładach konieczne jest opracowywanie odpowiednich mikroprogramów i technicznych sposobów wpisywania ich do pamięci tych maszyn lub ich elementów. Rozwój metod i środków automatyzacji projektowania zabiegów produkcyjno-technologicznych staje się coraz bardziej zintegrowany z rozwojem automatyzacji projektowania maszyn. Sprzyja temu konieczność korzystania w dużym stopniu ze wspólnych baz danych oraz możliwość uzyskania w wielu wypadkach rozwiązań problemów produkcyjno-technologicznych jako konsekwencji rozwiązania poprzednich etapów projektowania maszyn. Sprzyja temu obserwowana w skali światowej dążność do integrowania się komputerowo wspomaganego wytwarzania (computer aided manufacturing - CAM) z komputerowo wspomaganym projektowaniem (computer aided design - CAD).

Metody symulacyjne są szczególnie przydatne przy rozwiązywaniu równań różniczkowych cząstkowych, równań całkowych oraz zadań optymalizacyjnych sformułowanych w kategoriach programowania matematycznego (szczególnie nieliniowego). Np. problemy projektowania takich elementów maszyn, jak wały, łożyska, różnego rodzaju przekładnie są przeważnie formułowane w kategoriach nieliniowego programowania matematycznego. Ponieważ z reguły, wiedza o "lokalizacji" rozwiązań optymalnych w przestrzeni rozwiązań dopuszczalnych w takich zadaniach jest uboga oraz często funkcje celu nie są różniczkowalne, praktycznie pozostają do dyspozycji tylko metody Monte Carlo (metoda błędzenia przypadkowego, metoda losowego gradientu).

Wreszcie na zakończenie powiemy kilka zdań o modelowaniu procesów ciąгло-dyskretnych. Proces taki jest pewną kompozycją procesów ciągłych i dyskretnych, konieczną wówczas, gdy nie można pominąć w modelowaniu faktu, iż proces dyskretny zachowuje się między kolejnymi zdarzeniami jak proces ciągły. Istotą języków do modelowania takich procesów jest konieczność rozszerzenia pojęcia atrybutów na zmienne zależne (dziedziny zależne) lub funkcje. Tak więc w trakcie działania operatora planowania zdarzeń, a ściślej w trakcie realizowania określonego algorytmu operator przyswajania nie może odwzorować bezpośrednio tego rodzaju atrybutu (choć wartość funkcji w chwili działania tego operatora jest znana), gdyż o odwzorowaniu decyduje wartość pewnego funkcjonału określonego na realizacji tej funkcji w nieznanym a priori przedziale czasu. Oczywiście nie ma potrzeby wykazywania jak ważnym narzędziem w projektowaniu maszyn mogą być języki modelowania procesów ciąгло-dyskretnych. Ograniczymy się do stwierdzenia, że ich konstruowanie napotyka na duże trudności, również techniczne: próby konstruowania takich języków przez rozszerzanie języka SIMULA rodzą problem eliminowania z baz danych, danych zbędnych (oczyszczania baz danych). Docho- dzi nawet do tego, że przy takim modelowaniu mówi się, iż przestrzeń istnienia zmiennych przekracza przestrzeń istnienia obiektu. Sytuacje takie dopuszczalne są w języku SIMSCRIPT. Tak czy inaczej ograniczeniami przy takim modelowaniu są: pojemność pamięci oraz odpowiedzialność programisty za korzystanie z odwołań do "histogramów".

Na zakończenie podkreślimy również przydatność języków modelowania przy organizowaniu eksperymentów quasi-naturalnych (przebiegających prawie w czasie rzeczywistym).

Widzimy więc, że na etapie makroprojektowania maszyn, metody modelowania maszyny, jako systemu, za pomocą cyfrowej maszyny matematycznej i odpowiednich języków modelowania procesów ciągłych, dyskretnych i ciąгло-dyskretnych, obok symulacyjnych metod badania modeli matematycznych maszyny lub jej elementów są podstawowymi metodami i narzędziami projektanta.

Metody te i narzędzia uzupełniają się wzajemnie z komputerowymi systemami informacyjnymi w zakresie informacji naukowo-technicznej, ekonomicznej i produkcyjno-technologicznej oraz pakietami programów problemowo zorientowanych na projektowanie różnego rodzaju elementów (części) maszyny.

### 3. Automatyzacja projektowania elementów maszyn

Oczywiście pojęcie "elementy maszyny" jest pojęciem względnym i sugeruje jedynie, że w procesie projektowania maszyna została potraktowana jako system. Takie podejście okazuje się niezbędnym przy projektowaniu nowoczesnych maszyn z wielu powodów, między innymi dlatego, że pakiety programów komputerowo wspomaganego projektowania budowano są jako pakiety problemowo zorientowane, to znaczy dostosowane do wspomaganego projektowania określonych klas urządzeń, np. takich jak:

- układy automatycznej regulacji,
- napędy elektryczne,
- układy hydrauliczne i pneumatyczne,
- przekładnie zębate,
- mikroelektroniczne układy kontrolno-pomiarowe itp. urządzenia.

Stąd też jeśli w procesie projektowania maszyny chcemy skorzystać z metod i narzędzi komputerowo wspomaganego projektowania, to powinniśmy zdekomponować maszynę (traktowaną jako system) tak, aby możliwie najefektywniej móc posłużyć się dostępnymi pakietami programów problemowo zorientowanych. Należy wymienić przynajmniej trzy zasadnicze czynniki kształtujące problemowo zorientowany pakiet programów komputerowo wspomaganego projektowania.

● Po pierwsze - część programów pakietu, szczególnie tych, które przeznaczone są do realizowania zadań syntezy i poszukiwania rozwiązań optymalnych, łączy wspólna i specyficzna dla nich cecha "wyspecjalizowania" do opisu i badania klasy modeli właściwych problemom projektowym danej klasy urządzeń. Oczywiście jest, że określony program biblioteki standardowych programów numerycznych, np. z zakresu programowania nieliniowego, może posłużyć do rozwiązywania różnych problemów optymalizacyjnych występujących, zarówno przy projektowaniu układów elektronicznych, jak i kół zębatych. Analogiczny program w odpowiednim pakiecie programów problemowo zorientowanych będzie natomiast wyspecjalizowany w tym sensie, że uwzględni własności funkcji celu i więzów określających przestrzeń rozwiązań dopuszczalnych, a więc własności odnoszące się do określonych typów projektowanych układów. Tak więc za cenę uniwersalności programów w pakiecie problemowo zorientowanym uzyskuje się ich większą efektywność obliczeniową, co praktycznie oznacza uzyskanie szybszej zbieżności do rozwiązań quasi-optymalnych. Warto przy okazji podkreślić, że w powyższym sensie określone "specjalizowanie się" odpowiednich programów analizy numerycznej w pakietach problemowo zorientowanych, jest wynikiem samouczenia się autora programu oraz wprowadzanych zmian i postępuje wraz z nagromadzonym doświadczeniem, zwłaszcza że analityczne badanie własności przestrzeni, w których poszukuje się rozwiązań z reguły nie daje wystarczających podstaw do optymalnego budowania programów i procedur obliczeniowych.

● Po wtóre - projektowanie elementów maszyn przy użyciu pakietów programów problemowo zorientowanych obejmuje zarówno problemy projektowania funkcjonalnego jak i technicznego. Tak więc pakiet programów powinien tworzyć spójny system numeryczno-logicznego przetwarzania danych opierając się zarówno na wynikach syntezy i obliczeń optymalizacyjnych, jak i na zbiorze danych o obowiązujących w danej konstrukcji normach i standardach. Oczywiście zbiór danych o normach i standardach, a szczególnie zbiór preferowanych do zastosowania czy też wręcz obowiązujących w rozwiązaniu typodetali, podzespołów i elementów konstrukcyjnych nie tylko nadaje pakietowi jeszcze większą problemową orientację na projektowanie określonego typu urządzenia lecz nawet dostosowuje go do technologii stosowanej przez określonego producenta. Warto przy okazji zwrócić uwagę, że z jednej strony cechy zbioru norm, standardów, typodetali i tym podobnych elementów wpływają istotnie na strukturę danych w systemie przetwarzania informacji, orientując już tym samym system na określony typ urządzeń, zaś z drugiej strony dążność do optymalizowania struktury danych, uzależniona od charakteru odwołań do zbiorów danych w procesie projektowania, sprzyja również orientowaniu bazy danych na określony typ projektowanych urządzeń. Należy też podkreślić, że efektywność tego typu systemów komputerowo wspomaganego projektowania w istotny sposób zależy od cech przyjętej bazy danych oraz systemu zarządzania tą bazą i to tak dalece, że optymalizowanie bazy danych staje się często głównym czynnikiem poprawiającym efektywność systemu, dystansującym swym wpływem takie czynniki jak: wzbogacanie konfiguracji systemu, zwiększanie wydajności procesora, pojemności pamięci operacyjnej i in. Należy również nadmienić, że specyfika baz danych w problematyce automatyzacji projektowania zrodziła wiele kierunków poszukiwań nowych rozwiązań w tej dziedzinie. Przykładowo mogą tu być badania w zakresie relacyjnych baz danych.

● Po trzecie - pakiety programów problemowo zorientowanych zwykle składają się z kilkudziesięciu, a nawet kilkuset programów wzajemnie współdziałających w procesie projektowania. Sposób współdziałania programów może być w różnym stopniu, w różnych rozwiązaniach modyfikowany przez ingerencję projektanta lub nawet automatycznie, dzięki istnieniu w tym zbiorze programów parametryzowanych i sterujących. Tak czy inaczej, dążność do optymalizowania programów, w sensie ich efektywności obliczeniowej, i do zapewnienia możliwie znacznej elastyczności sposobów współdziałania powoduje, że znaczna część programów jest pisana nie w uniwersalnych językach algorytmicznych, a więc, że pakiet programów problemowo zorientowanych jest w jakimś stopniu jednocześnie pakietem programów maszynowo zorientowanych. Należy nadmienić, że sposób zorganizowania zbiorów danych również wpływa na maszynowe zorientowanie pakietu.

Tak więc praktycznie projektant maszyny może znaleźć się w sytuacji, w której ma dostęp do pakietu programów komputerowo wspomaganego projektowania wielostopniowej przekładni zębatej za pomocą maszyny matematycznej Mińsk-32 i pakietu programów komputerowo wspomaganego projektowania układów cyfrowego sterowania tą maszyną za pomocą maszyny matematycznej IRIS-80 i to w technologii właściwej określonemu producentowi tych układów. W powyższym przykładzie prawdopodobnie można o-

bydwa zadania projektowe rozwiązać efektywnie w sposób niezależny, często jednak niestety tak nie jest. Powstają więc ważne problemy związane z opracowaniem efektywnych narzędzi automatyzowania (za pomocą komputerów) "przenoszenia" odpowiedniego oprogramowania i baz danych między maszynami matematycznymi różnych typów lub między różnymi komputerowymi systemami automatyzacji projektowania.

#### 4. Automatyzacja projektowania procesów produkcyjno-technologicznych

Powyżej użyto sformułowania "procesy produkcyjno-technologiczne", gdyż zamiarem autora jest rozpatrzenie zarówno takich procesów technologicznych, jak wszelkiego rodzaju obróbki mechanicznej, chemicznej itp. jak też procesów montażowych, transportowych, kontrolno-pomiarowych i in.

Rozpatrzenie automatyzacji projektowania procesów produkcyjno-technologicznych w aspekcie automatyzacji projektowania maszyn uzasadnione jest między innymi następującymi względami:

- automatyzacja projektowania maszyn powinna uwzględniać stopień automatyzacji procesów produkcyjno-technologicznych w zakładzie, w którym projektowana maszyna będzie wytwarzana;
- ostatnio coraz częściej zdarza się, że zdolność (podatność) projektowanej maszyny na automatyzowanie projektowania procesów produkcyjno-technologicznych, do wykonywania których jest przeznaczona lub może być użyta, staje się istotnym parametrem w procesie projektowania; wyraźnie występuje to w przypadku takich maszyn jak wielostanowiskowe, zautomatyzowane contra produkcyjnie;
- urządzenia wspomagające (automatyzujące) projektowanie procesów produkcyjno-technologicznych są również bardzo często maszynami. Stąd między innymi rodzą się nowe wymagania na nowe maszyny.

Zauważmy, że zdecydowana większość wszystkich procesów produkcyjno-technologicznych występujących w przemyśle maszynowym ma charakter złożonych i silnie uwarunkowanych procesów dyskretnych, co oczywiście rodzi określonej klasy problemy w automatyzowaniu projektowania tych procesów. Powoduje to określone konsekwencje w niezbędnym formalizowaniu poszczególnych elementów tych procesów, budowaniu modeli matematycznych i poszukiwaniu rozwiązań. Często wyrażany jest nawet pogląd, że powyższa właściwość procesów produkcyjno-technologicznych, występujących w przemyśle maszynowym, czyni problem automatyzowania projektowania tych procesów wyjątkowo trudnym, a nawet trudniejszym od automatyzowania projektowania analogicznych procesów w innych przemysłach. Prawdopodobnie jest w tym trochę racji chociażby dlatego, że jak wiadomo - procesy ciągle łatwiej się modeluje i bada. Niemniej jednym z czynników istotnie ograniczających postęp w automatyzowaniu projektowania tych procesów są stosowane współcześnie w przemyśle technologicznym, nie dające z reguły formalnych podstaw do rozwiązywania problemów, które można by rozwiązać za pomocą współczesnej techniki obliczeniowej. Dodatkowo - różnorodność metod i środków technicznych, maszyn roboczych, wyposażenia technicznego i aparatury, występująca w poszczególnych przedsiębiorstwach doprowadza do tego, że jednakowe nawet produkty wytwarza się według różnych procesów produkcyjno-technologicznych. Tak więc, poza różnymi trudnościami na jakie napotyka automatyzacja projektowania procesów produkcyjno-technologicznych w przemyśle maszynowym, występuje również konieczność dostosowania się do powyższych warunków oraz rozwijania metod i środków tak, aby było możliwe projektowanie procesów odpowiadających warunkom konkretnego zakładu przemysłowego.

Słusznie chyba będzie omówienie problemów automatyzacji projektowania procesów produkcyjno-technologicznych w podziale na trzy następujące grupy:

- problemy wynikające przy opracowywaniu projektu koncepcyjnego procesów produkcyjno-technologicznych w nowych warunkach działania zakładu, spowodowanych np. uruchamianiem produkcji nowych wyrobów, modernizacją zakładu, zmianami w sieci kooperacyjnej zakładu i podobnymi czynnikami;
- problemy wynikające przy projektowaniu konkretnej realizacji procesu produkcyjno-technologicznego, a polegające głównie na rozdzieleniu zadań dla poszczególnych elementów realizujących ten proces oraz

• problemy wynikające przy projektowaniu konkretnych zabiegów produkcyjno-technologicznych, polegające na sporządzaniu roboczej dokumentacji produkcyjno-technologicznej dla poszczególnych stanowisk i operacji oraz programów sterujących, o ile te operacje są wykonywane przez zautomatyzowane maszyny (urządzenia).

Problemy pierwszej grupy nie są właściwie problemami automatyzacji projektowania maszyn lecz informatyki, organizacji przemysłu, badań operacyjnych i innych pokrewnych dyscyplin. Są też przedmiotem badań, którymi zajmują się te dyscypliny naukowe. Znaczenie użytkowe rozwiązań tych problemów jest bardzo duże, zarówno z racji konieczności podążania za znaczną dynamiką zmian warunków działania zakładów, jak i rezerw tkwiących w optymalizowaniu procesów produkcyjno-technologicznych oraz ze względu na możliwości wskazania efektywnych kierunków modernizacji bazy produkcyjno-technologicznej. Brak ewidentnych rezultatów użytkowych w tej dziedzinie badań spowodowany jest w większym stopniu brakiem dostatecznie wiernych i wystarczającej bazy danych niż brakiem odpowiednich metod (modeli matematycznych) i narzędzi, którymi w tej sytuacji może być tylko dostatecznie bogata i silna konfiguracja systemu komputerowego dysponującego oprogramowaniem symulacyjnym. Rozwój badań z zakresu automatyzacji projektowania maszyn, a szczególnie z zakresu pozostałych spośród wymienionych wyżej zagadnień automatyzacji projektowania procesów produkcyjno-technologicznych, będzie miał z pewnością korzystny wpływ na osiągnięcia w tej grupie problemów.

Problemy drugiej grupy, a więc wynikające przy projektowaniu konkretnej realizacji procesu produkcyjno-technologicznego oraz polegające głównie na rozdzielaniu zadań dla poszczególnych elementów realizujących ten proces, są w różnym stopniu zarówno problemami automatyzacji technicznego przygotowania produkcji, jak i automatyzacji projektowania maszyn, przy czym stopień wzajemnego działania tych dziedzin jest zależny od poziomu zautomatyzowania procesów produkcyjno-technologicznych oraz od charakteru produkcji.

Na przykład w zakładzie o niskim stopniu zautomatyzowania procesów produkcyjno-technologicznych, realizującym seryjną, jednorodną produkcję, omawiane problemy są realizowane wyłącznie przez komputerowy system zarządzania przedsiębiorstwem za pomocą typowych pakietów programów technicznego przygotowania i planowania produkcji, zawierających między innymi programy: harmonogramowania prac, rozwinięć materiałowych, zaopatrzenia w surowce, podzespoły i narzędzia, kosztorysowania i tym podobno programy. Jeśli jednak w tym samym zakładzie produkuje się seryjnie wyroby niejednorodne w tym sensie, że każdy może być odpowiednią (dopuszczalną w określonym znaczeniu) kompilacją zbioru elementów, będącego podzbiorem określonego dla danej klasy wyrobów zbioru elementów typowych, to proces produkcyjno-technologiczny jest w dużym stopniu wynikiem zaprojektowania odpowiednich wyrobów.

Powyższa sytuacja występuje na przykład w zakładach produkujących maszyny sposobem "na życzenie klienta". Zauważmy tylko, że ten sposób produkowania maszyn staje się coraz częstszy w nowoczesnym przemyśle i zasługuje na bardzo wnikliwe rozpatrzenie, gdyż rodzi wiele ciekawych wniosków, między innymi dotyczących kierunków rozwoju metod i środków automatyzacji projektowania maszyn. Rozpatrzmy wreszcie przykład zakładu charakteryzującego się wysokim lub nawet bardzo wysokim stopniem zautomatyzowania procesów produkcyjno-technologicznych. W jakimś stopniu ilustracją tego przypadku jest opisane już poprzednio centrum obróbki korpusów. Jedną z cech konstruowania maszyn klasy takiego centrum jest dążność do uzyskania cech samoadaptowalności maszyny do określonego typu zadań i stanu wewnętrznego maszyny. Uzyskuje się to między innymi przez stosowanie różnego rodzaju nadmiarów w systemie, którym jest taka maszyna oraz przez zastosowanie lokalnego (własnego) systemu operacyjnego (systemu sterującego). Wówczas konkretny przebieg procesu produkcyjno-technologicznego może być w dużym stopniu (lub całkowicie) zaprojektowany w sposób automatyczny jako wynik odpowiedniego zaprojektowania maszyny.

Tak więc w drugiej grupie problemów automatyzacji projektowania procesów produkcyjno-technologicznych rozpatrzyliśmy trzy charakterystyczne przykłady ilustrujące relacje między poziomem zautomatyzowania procesów produkcyjno-technologicznych i charakterem produkcji a automatyzacją projektowania maszyn. Pierwszy przykład znajduje potwierdzenie w Polsce dość powszechnie. Problemy automatyzacji projektowania procesów produkcyjno-technologicznych, w tym wypadku prowadzącej się do wdrażania z reguły dość powszechnych w świecie i w naszym kraju wspomnianych pakietów programów z zakresu technicznego przygotowania produkcji.

## 5. Problemy komunikowania się projektanta z komputerowo wspomaganym systemem projektowania maszyn

Problemy komunikowania się projektanta z systemem komputerowo wspomaganego projektowania można podzielić na trzy następująco podzbiory: problemy wprowadzania danych, prezentacji wyników oraz konwersacji projektanta z systemem liczącym. W stosunku do każdego z tych trzech podzbiorów systemy komputerowo wspomaganego projektowania stawiają wymagania o wysokim współczynniku trudności. Wynika to z jednej strony z takich cech procesu projektowania jak:

- iteracyjność,
- interakcyjność,
- interdyscyplinarność,
- kompleksowość,
- kolektywność,
- wariantowość itp.

a z drugiej strony

- z architektury i struktury oprogramowania systemów komputerowych,

w których duża moc obliczeniowa i obszerna zasoby pamięci operacyjnych i zewnętrznych muszą iść w parze z szybką i elastyczną obsługą procesów wprowadzania i wyprowadzania danych. Nie wdając się w tym miejscu w szczegóły analizujące wpływ rozwoju automatyzacji projektowania maszyn na architekturę systemów komputerowych oraz analizujące efektywność użycia uniwersalnych maszyn matematycznych do automatyzowania procesów projektowania maszyn należy zauważyć, że w zasadzie uniwersalne maszyny matematyczne, a tym bardziej konfiguracje komputerowe, zorientowane na automatyzowanie procesów zarządzania (komputerowe systemy informatyczne), charakteryzują się średnią lub nawet słabą przydatnością i efektywnością w rozwiązywaniu złożonych problemów projektowania maszyn. Oczywiście nie oznacza to, że uniwersalnych komputerów nie stosuje się efektywnie w automatyzowaniu projektowania elementów maszyn, dokumentacji produkcyjno-technologicznej, w technicznym przygotowaniu produkcji lub do rozwiązywania niektórych problemów syntezy, optymalizacji, modelowania itp. Faktem natomiast jest, że komputerowe systemy zorientowane na automatyzowanie projektowania złożonych maszyn powstały w mniejszym lub większym stopniu jako systemy unikalne i multi-procesorowe (zawierające specjalizowane procesory macierzowe, graficzne i inne). Wzrost znaczenia automatyzacji projektowania maszyn, opierając się na uniwersalnych systemach komputerowych, spowodował między innymi dążność do tworzenia systemów bardziej przystosowanych do rozwiązywania takich zadań. Szczególnie systemy minikomputerowe tworzy się jako problemowo zorientowane, przystosowane do realizowania określonego typu zadań projektowych i w odniesieniu do określonego typu projektowania maszyn.

Sytuacja jest tym trudniejsza, że wymagania dotyczące środków i metod komunikacji projektanta z systemem komputerowo wspomaganego projektowania dynamicznie zmieniają się wraz z automatyzowaniem procesów produkcyjno-technologicznych w przemyśle produkującym projektowane maszyny. Tradycyjno miary jakości metod i środków komunikowania się z komputerem nie przystają należycie do metod i środków wymagających w systemach zastosowania komputerowo wspomaganego projektowania maszyn (KWPM). Jedną z nielicznych tradycyjnych miar, przydatną i w ocenie systemów KWPM jest czas reakcji systemu na żądanie projektanta. Większość nowych kryteriów oceny jakości komunikowania się projektanta z systemem KWPM, poza problemami technicznymi musi uwzględniać psychologiczne asocjacje projektanta wynikające z tego, czy system nadaje czy też nie za jego pracą, "rozumie" czy też "nie rozumie" jego poleceń itp. W szczególności chodzi tu o wyeliminowanie takich negatywnych w procesie projektowania stanów psychicznych projektanta jak:

- znudzenie, jako konsekwencja niewłaściwie dobranych do różnych klas zadań projektanta - czasów reakcji systemu;
- niepokój, jako konsekwencja niespodziewanie długich opóźnień w oczekiwaniu na odpowiedź komputera;
- frustracje, jako wynik nieoczekiwanej odpowiedzi na zadane pytanie;
- zakłopotanie, będące skutkiem na przykład zbyt ubogiego zbioru poleceń, które można wydawać systemowi, a tym samym niemożności przekazania swoich życzeń lub też zbyt bogatego zbioru możliwych poleceń.

Języki komunikowania się projektanta z systemem powinny w swojej strukturze być jak najbliższe naturalnych języków, którymi posługują się projektanci przy opisywaniu określonego problemu oraz powinny być tak skonstruowane, aby projektant mógł koncentrować się wyłącznie na semantyce tego, co chce przekazać systemowi, a nie stać się zaabsorbowanym obawą przed naciśnięciem niewłaściwego guziczka czy też koniecznością wpisania kilku różnych nawiasów, myślników lub innych znaków. Poza tym projektant nie powinien tracić zbyt wiele czasu na nauczanie się sposobów korzystania z systemu komputerowego. Zadania powyższe coraz lepiej realizują projektanci systemów komputerowo wspomaganego projektowania np. wzrostowi liczby różnorodnych języków zorientowanych na opisywanie określonej klasy problemów przeciwstawia się komputerowo wspomaganą systemy nauczania tych języków.

Projektowanie jest z natury rzeczy procesem iteracyjnym o różnej strukturze na różnych poziomach szczegółowości traktowania problemu i różnej fazie przybliżenia jego realizacji do zamierzonego celu. Reguły dekompozycji i agregacji problemu wnoszą jeszcze dodatkowe cechy do procesu projektowania. Złożona struktura procedur występujących w projektowaniu utrudnia zrealizowanie programów zapewniających zautomatyzowany przebieg procesu projektowania. Są wprawdzie sytuacje, gdy w pełni uzyskuje się taki rezultat, lecz nie są one reprezentatywne dla większości problemów praktycznie występujących w automatyzacji projektowania maszyn. Prace badawcze zmierzają oczywiście w kierunku rozszerzenia zakresu automatyzowania przebiegu procesu projektowania, przy czym jednym z kierunków poszukiwań jest dążność do wykorzystania wiedzy wynikającej z rezultatów, które uzyskano stosując określone procedury przy rozwiązywaniu określonych zadań projektowych. Wiąże się to z poszukiwaniem relacyjnych metod i relacyjnych baz danych oraz z odpowiednim budowaniem i użytkowaniem systemów KWPM. Tak czy inaczej, automatyzacja projektowania maszyn wymaga od systemu KWPM możliwości interakcyjnego trybu pracy. Niezbędność ta występuje oczywiście w różnym stopniu przy badaniu matematycznego modelu maszyny i przy projektowaniu dokumentacji konstrukcyjno-technologicznej. Wprawdzie przebieg procesu projektowania (polegającego na przykład na badaniu metodami symulacyjnymi niektórych własności maszyny lub na poszukiwaniu optymalnych rozwiązań) tak jak w ogóle każdy dosyć dobrze sformułowany w sensie matematycznym problem, daje się dość łatwo rozwiązać, to jednak i w tych wypadkach dysponowanie interakcyjnym trybem pracy jest pożądane i efektywne. Pozwala to skrócić czas obliczeń, analizować warianty, modyfikować funkcje celu i wiązy ograniczające oraz wykonywać inne podobne zabiegi będące czasami wręcz jedynymi środkami, które, przy będącym do dyspozycji potencjale systemu KWPM, umożliwiają wykonanie zadania projektowego.

Kompleksowość problemów projektowania maszyn wiąże się z koniecznością tworzenia i przechowywania obszernych zbiorów informacyjnych, a także zarządzania nimi. Aczkolwiek zasadniczo jest to zadanie z zakresu organizacji banków danych, to jednak i w stosunku do środków i metod komunikacji projektanta z komputerem pojawia się wiele specyficznych wymagań.

- Pierwsze z nich dotyczy konieczności prezentowania informacji w różnej formie, z wykorzystaniem różnych środków technicznych.
- Drugie odnosi się do stworzenia odpowiedniego podsystemu gromadzenia danych i wprowadzania ich do banku danych.
- Trzecie wreszcie odnosi się bezpośrednio do sposobu organizacji dostępu do banku danych, do brania pod uwagę konieczności transformacji zawartych w nim informacji, z uwzględnieniem specyficznej struktury strumienia informacji wyjściowej (odnosi się to szczególnie do wyprowadzenia informacji na monitory graficznie).

Należy przy tym pamiętać, że miernikiem jakości jest tu czas reakcji systemu na żądanie projektanta dostarczenia odpowiednich informacji oraz zmniejszanie się strumienia wprowadzanej informacji w trakcie przechodzenia do kolejnych etapów projektowania i kolejnych modułów na każdym etapie. W nawiązaniu do omawianej tu cechy systemów KWPM rodzi się spostrzeżenie, że prezentacja wyników przestaje być końcowym etapem pracy z komputerem, ponieważ po każdej fazie projektowania jej wyniki muszą być dodatkowo przetworzone do postaci akceptowanej, jako dane wejściowe dla innej fazy. Byłoby to spostrzeżenie trywialne, gdyby nie jego następstwa dla sposobu oprogramowania systemu i zasad tworzenia baz danych.

Interdyscyplinarność systemów KWP przejawia się w konieczności zapewnienia środków i metod współdziałania przy realizacji projektu specjalistom o różnorodnych zawodach. System KWPM staje się z tego punktu widzenia przekazywaczem wiadomości z różnych dziedzin wiedzy. Dla metod i środków komunikacji z systemem KWP oznacza to konieczność opracowania różnorodnych języków konwersacji i różnorodnych form prezentacji danych, a także metod i środków ich wzajemnej przenaszalności (kompatybilności).

Jak już zostało wcześniej powiedziane, kolejne fazy projektowania stawiają zróżnicowane wymagania w stosunku do metod i środków komunikowania się projektanta z systemem KWPM. Na etapie formułowania założeń na maszynę istotne jest uzyskanie możliwości odwołania się do "wiedzy" systemu KWPM, to jest do informacji o patentach, standardach, cenach, technologii produkcji, parametrach innych obiektów tej samej klasy i podobnych danych. Uzyskane informacje należy przedstawić projektantowi w formie dla niego czytelnej, to znaczy przyjętej w jego kręgu zawodowym. Techniki prezentacji tu stosowane to: wykresy, histogramy, tabele, a więc częściowo wykorzystujące tradycyjną metodę wydruku, ale najczęściej wymagają one stosowania bardziej nowoczesnych urządzeń w postaci monitorów alfanumerycznych, autokreślarek i innych podobnych urządzeń. Gromadzenie danych dla tej fazy projektowania opiera się na dość już rozpowszechnionych technikach przygotowania i kodowania danych na taśmach i kasetach magnetycznych. Język konwersacji z systemem - to najczęściej wybór "z możliwości". Metoda ta dla wszystkich hierarchicznie zorganizowanych zbiorów informacji jest najlepsza. Wkładem decydenta jest również wprowadzenie informacji uzupełniającej, której nie mógłby udzielić wcześniej, przed zaznajomieniem się ze stanem aktualnym procesu.

Na etapie opracowywania projektu koncepcyjnego istotne są metody i narzędzia komunikacji projektanta z systemem KWPM, ułatwiające rozwiązywanie problemów syntezy i analizy projektowanej maszyny bądź jej fragmentów. Wymaga to przede wszystkim języków do opisu tych problemów. Często problemy te dadzą się dobrze opisać analitycznie w postaci operatorów różniczkowych czy układów liniowych równań algebraicznych, albo wyrazić w kategoriach programowania matematycznego, zadań rachunku wariacyjnego, korelacyjnej teorii procesów stochastycznych i podobnych i wówczas bezpośrednio lub przy zastosowaniu przybliżonych metod rozwiązywania tych problemów można je opisywać językami algorytmicznymi, takimi jak ALGOL, FORTRAN, PL/1 i inne. Czasami wręcz można posłużyć się w tym celu standardową biblioteką programów numerycznych lub pakietem programów zorientowanych na dany rodzaj zadań analitycznych. Niemniej i tym razem pożądane są środki interakcyjnego działania projektanta, graficzne urządzenia odwzorowania wyników projektowania oraz metody i środki ułatwiające dekomponowanie problemu i agregowanie uzyskanych wyników częściowych. Wynikać to może stąd, że nie są dobrze znane a priori własności funkcji celu i że projektant wpływa na bieg procesu obserwując zachowanie się tego procesu i podobnych zjawisk. W wielu typach zadań syntezy i analizy korzysta się z opracowanych metod dekomponowania problemu, z tym że wykonywanie tego zabiegu bez automatycznego wspomaganie przez komputer eliminuje praktycznie te metody z użycia. Niestety wiele problemów tego etapu projektowania musi być opisanych za pomocą specjalizowanych języków, a wiele zagadnień może być rozwiązanych tylko na drodze modelowania i technik symulacyjnych. Interpretacja wyników, szczególnie w tych ostatnich wypadkach, wymaga z reguły urządzeń zobrazowania graficznego.

Etap projektowania dokumentacji produkcyjno-technologicznej wymaga zastosowania najszerszej gamy środków technicznych i metod komunikowania się projektanta z systemem KWPM. W stosunku do poprzednich faz projektowania wzrastają wymagania dotyczące dokładności urządzeń graficznego wprowadzania i wyprowadzania danych. Równocześnie bardzo często korzysta się z tradycyjnych metod i środków, na przykład z taśm perforowanych sterujących następnie autokreślarkami, numerycznie sterowanymi obrabiarkami itp.

Istotną cechą tej fazy projektowania jest często sięganie do środków symulujących przebieg zaprojektowanego procesu produkcyjno-technologicznego w celu zweryfikowania rozwiązania przez specjalistów różnych dziedzin. Ten etap projektowania charakteryzuje się również wykorzystywaniem wyspecjalizowanych języków do opisu problemu i istotnym wykorzystywaniem baz danych.



## 6. Automatyzacja projektowania maszyn w kontekście automatyzowania procesów technologicznych w przemyśle maszynowym i automatyzowania procesów technicznego przygotowania produkcji

Oczywiste jest, że sposób zorganizowania automatyzacji projektowania maszyny powinien uwzględniać zarówno stopień zautomatyzowania procesów technologicznych jak i stopień automatyzacji technicznego przygotowania produkcji w zakładzie, w którym dana maszyna ma być wytworzona bądź seryjnie produkowana. Naturalnie w różnym stopniu obowiązuje to w odniesieniu do poszczególnych elementów maszyny lub różnych aspektów jej projektowania. Z reguły jednak systemy automatyzacji projektowania, nie uwzględniające powyższego, nie mają praktycznie dużej wartości. Wynika to stąd, że efektywność automatyzacji projektowania maszyn, poza nielicznymi wyjątkami, polega (przynajmniej w obecnej fazie rewolucji naukowo-technicznej) na korzyściach wypływających z takiego połączenia etapów projektowania funkcjonalnego (konceptualnego) i technicznego (w dużym stopniu zrutynizowanego), że znaczna część dokumentacji produkcyjnej (technologicznej) jest wykonywana możliwie w sposób zautomatyzowany jako konsekwencja wykonanego projektowania konceptualnego. Praktycznie już przy obecnym poziomie zautomatyzowania procesów technologicznych oznacza to, że wraz z dokumentacją konstrukcyjną w postaci opisowej i/lub graficznej otrzymuje się z systemu komputerowo wspomaganego projektowania programy sterujące określonymi automatami technologicznymi lub dane ułatwiające sporządzenie takich programów. W powyższym rozumieniu zyski polegają głównie na istotnym skróceniu okresu wdrażania nowej konstrukcji do produkcji i istotnym zmniejszeniu liczby błędów w dokumentacji, a nie na optymalnym rozwiązaniu konstrukcji. Pogląd ten wymaga oczywiście bardziej wnikliwego uzasadnienia i to opartego na analizie przynajmniej następujących trzech problemów:

- w jakim stopniu automatyzowanie projektowania, i to głównie z punktu widzenia konceptualnego, jest obecnie praktycznie jedynym sposobem znalezienia rozwiązania i to nie koniecznie quasi-optymalnego lecz w ogóle realizującego dane założenia funkcjonalne;
- jaki jest rozmiar prac projektowo-technicznych o cechach zrutynizowanych, niezbędnych do sporządzenia dokumentacji produkcyjno-technologicznej i eksploatacyjnej;
- jak liczny jest wspólny zbiór dziedzin, na którym zbudowane są odpowiednie relacje niezbędne do projektowania konceptualnego jak i zrutynizowanego.

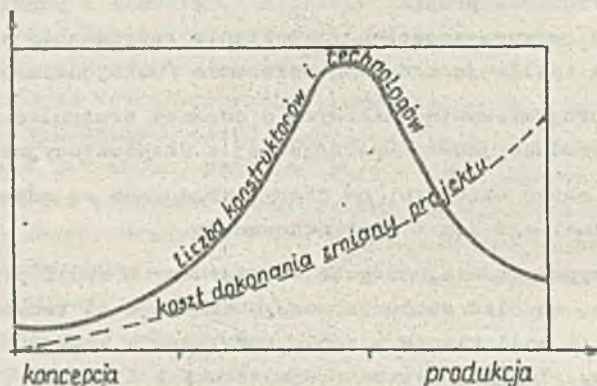
Istnieją podstawy do przypuszczenia, że problem pierwszy w skali przemysłowej występuje jeszcze stosunkowo sporadycznie, chociaż ewolucja maszyn wskazuje na tendencje wzrostu znaczenia automatyzacji projektowania dla znalezienia w ogóle rozwiązania konstrukcyjnego spełniającego określone założenia funkcjonalne. Znaczna liczba projektowanych i produkowanych maszyn charakteryzuje się tym, że rozmiar prac projektowo-technicznych o cechach zrutynizowanych jest znacznie większy od rozmiaru prac konceptualnych z tym, że ewolucja maszyn i sposobów ich wytwarzania wskazuje na tendencje do zmniejszania tej różnicy. Wspólny obszar danych niezbędny zarówno do projektowania konceptualnego jak i zrutynizowanego był i jest znaczny, a dążność do unifikacji bazy podzespołowo-technologicznej i modularności rozwiązań konstrukcji wskazuje na tendencje zwiększania się tego obszaru.

Podbudujmy i uzupełnijmy wypowiedziane wyżej poglądy niektórymi przykładami lub rozważaniami.

Z danych zebranych w przemyśle amerykańskim i omawianych na I Seminarium na temat automatyzacji projektowania i automatyzacji wytwarzania, które odbyło się w 1972 r. w Atlancie, (Georgia, USA) wynika (rys. 5), że liczba konstruktorów i technologów zatrudnionych przy projektowaniu maszyny osiąga maksimum w fazie opracowania dokumentacji produkcyjno-technologicznej. To oczywiste i znajduje potwierdzenie nie tylko w tym źródle. Stąd obniżenia kosztów projektowania maszyny należałoby szukać głównie w zautomatyzowaniu projektowania dokumentacji produkcyjno-technologicznej. Natomiast koszt dokonania zmian w projekcie wzrasta wykładniczo w funkcji czasu projektowania maszyny. Prawdopodobieństwo zdarzenia polegającego na konieczności dokonania takich zmian jest z reguły wyższe gdy nie stosuje się metod i środków automatyzacji projektowania maszyny już na etapach założeń i projektu koncepcyjnego, i to tym bardziej wyższe im projektowana maszyna jest bardziej złożona. Warto by głębiej zbadać te zależności uwzględniając, że część kosztów ogólnych, stanowiąca koszty dokonywania zmian w projektowanej konstrukcji jest funkcją określoną na

realizacjach procesu losowego, którymi są poszczególne zmiany i to określonym w przedziale czasowym. Na długość tego przedziału czasowego ma również istotny wpływ stosowanie lub nie automatyzacji projektowania i to w różnych fazach projektowania. Z literatury wiadomo jest, że zastosowanie automatyzacji projektowania złożonych maszyn w takich firmach amerykańskich jak General Motors, IBM, Boeing czy Lockheed, skróciło czas pełnego cyklu projektowania dwu a nawet trzykrotnie. Prawdopodobnie fakt ten wpływa również na atrakcyjność wyrobów.

Weźmy pod uwagę inny przykład. Często w literaturze fachowej z zakresu automatyzacji projektowania ocenia się efektywność różnych metod poszukiwania optymalnych rozwiązań. Na przykład optymalizacja (quasi-optymalizacja) projektowania wielostopniowej przekładni zębatej, w sensie zminimalizowania objętości, daje efekt rzędu 10 - 15% w stosunku do rozwiązania uzyskanego metodami tradycyjnymi, przy czym wyższość jednej metody nad inną wyraża się tym, że czas liczenia (z punktu widzenia wyłącznie czasu pracy procesora) wynosił 3 minuty, a nie 5 minut. Czas tradycyjnego projektowania w tym wypadku jest porównywalny z czasem przygotowania danych dla systemu komputerowo-wspomaganej projektowania i jest kilkakrotnie krótszy wtedy, gdy dysponujemy komputerowym systemem już wyspecjalizowanym w projektowaniu takich przekładni. Powyższe nie dyskwalifikuje oczywiście sensowności stosowania metod optymalizacyjnych. Można wymienić wiele przykładów, gdy uzyskanie nawet minimalnie lepszych rozwiązań niż dają to metody tradycyjne, ma bardzo istotne techniczne, poznawcze bądź ekonomiczne znaczenie. Podajmy jeszcze inny przykład. Poszukiwanie testów kontrolno-diagnostycznych, a więc rozstrzygających na podstawie obserwacji reakcji badanego obiektu na określono wymuszenia czy znajduje się on w stanie zdatności, czy też nie i ewentualnie w jakim stanie niezdatności się znajduje, jest jednym z ważnych problemów badawczych nauki i techniki.



Rys. 5. Pracochłonność opracowania maszyny w pełnym cyklu, a koszt zmian projektowych

Na etapie projektowania koncepcyjnego maszyny dąży się do zapewnienia istnienia takich testów, a na etapie projektowania produkcyjno-technologicznego wyznacza się takie testy jawnie dla poszczególnych zabiegów kontroli międzyoperacyjnej. Techniczne i ekonomiczne znaczenie pomyslnego rozwiązania tego problemu jest oczywiste. Pomimo rozwoju teorii testów diagnostycznych i wielu metod wyspecjalizowanych dla określonych klas badanych obiektów, w wypadku obiektów dosyć złożonych, praktycznie efektywnych testów nie udaje się uzyskać innymi metodami niż zautomatyzowanie ich projektowania za pomocą wydajnych maszyn matematycznych.

Można przypuszczać, że rola automatyzacji projektowania maszyn będzie wzrastała wraz ze złożonością tych maszyn i tym samym większym uzależnieniem znajdowania rozwiązań od metod i narzędzi automatyzacji projektowania, to znaczy wraz ze wzrastaniem kosztu szeroko pojętego "software" maszyny w stosunku do szeroko pojętego "hardware" tej maszyny. Przy całkowitej powściągliwości do zbyt pochopnego wyciągania wniosków warto jednak zwrócić uwagę na następujące fakty. Jeśli pakiet cyfrowy zawierający 12 - 16 mikroukładów elektronicznych dużej skali integracji, 50 średniej skali integracji i około 100 innych elementów kosztuje dwa razy tyle co koszt elementów, z których jest zmontowany, to cena mikrokomputera zawierającego 6 takich pakietów, odpowiednio ze sobą po-

łączonych i znających zasady współpracy ze sobą, jest ośmiokrotnie wyższa od kosztu elementów, z których jest zmontowany. Za co więc się płaci nabywając minikomputer?

Coraz bardziej powszechny staje się pogląd, że automatyzacja projektowania maszyn jest konsekwentnym, trzecim z kolei etapem obecnej fazy rewolucji naukowo-technicznej, której poprzednimi etapami były odpowiednio: mechanizacja i automatyzacja procesów produkcyjno-technologicznych oraz automatyzacja procesów sterowania i zarządzania, utożsamiona też czasami, między innymi i w Polsce, z zastosowaniem komputerowych systemów informatycznych.

Przeanalizujmy chociażby pobieżnie zasadnicze związki między etapami w kontekście poszczególnych faz działalności przemysłowej, takich jak przedprodukcyjna, produkcyjna i poprodukcyjna oraz szczebli organizacji przemysłowych, takich jak zakład produkcyjny, organizacje przemysłowe i centralne jednostki przemysłu maszynowego. Z dużym uproszczeniem można powiedzieć, że obszar działalności automatyzacji projektowania (rys. 6) pokrywa fazę działalności przedprodukcyjnej i część

	Fazy działalności przemysłowej		
	przed- produkcyjna	produkcyjna	po- produkcyjna
Centralne jednostki przemysłu maszynowego	Automatyzacja projektowania		
Organizacje przemysłowe		informatyka	
Zakłady produkcyjne		Automatyzacja procesów produkcyjno- technol.	

Rys. 6. Charakter automatyzacji w działalności przemysłowej

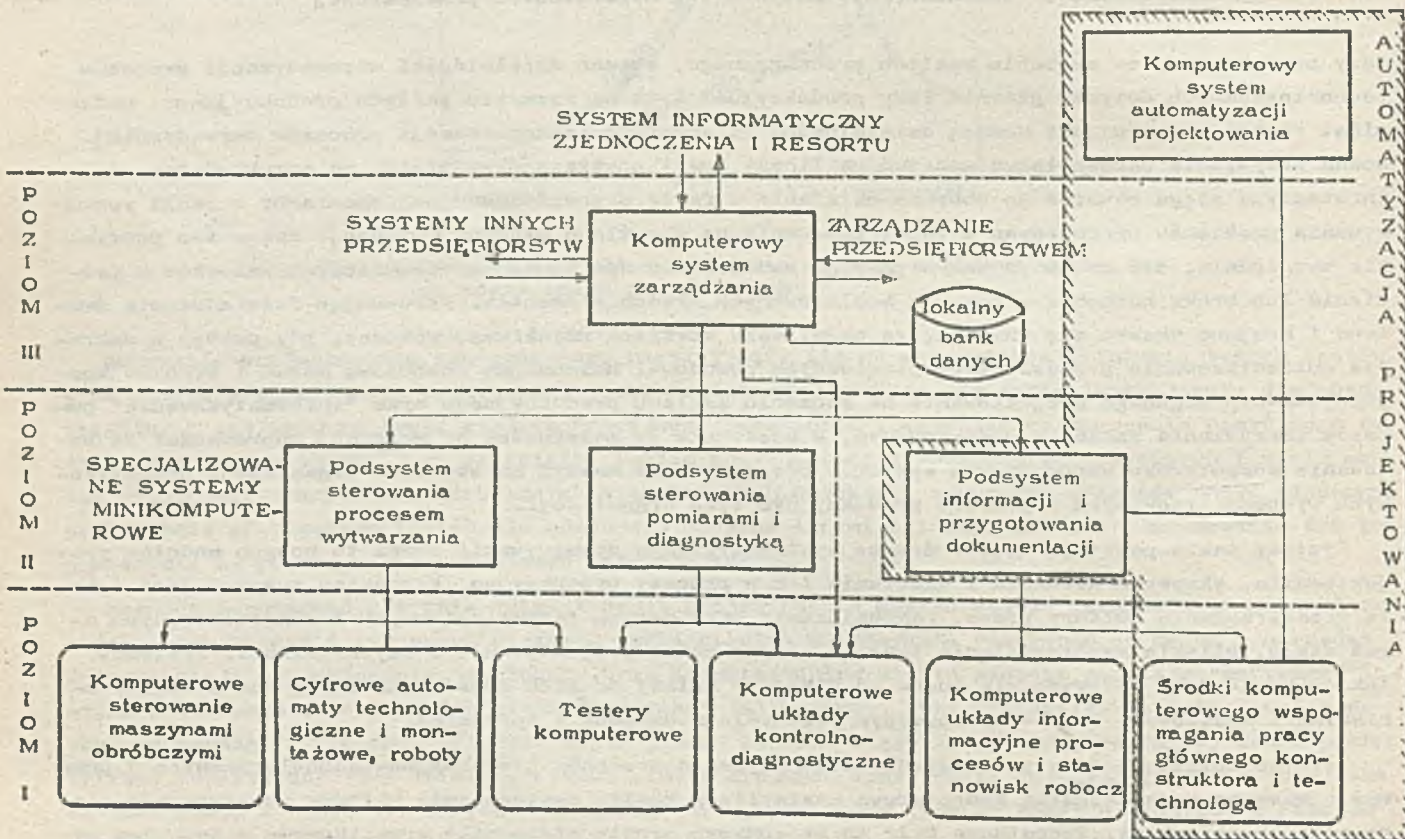
fazy produkcyjnej na szczeblu zakładu produkcyjnego, obszar działalności automatyzacji procesów technologicznych dotyczy głównie fazy produkcyjnej i to na szczeblu zakładu produkcyjnego, natomiast reszta obszaru jest domeną działalności informatyki (automatyzacji procesów zarządzania). Można oczywiście bardzo łatwo wskazać na liczne braki powyższego podziału, na przykład ten, że informatyka sięga również do obszaru działania w fazie przedprodukcyjnej, chociażby z racji rozwiązywania problemów prognozowania zapotrzebowania na określone maszyny (wyroby), czego ten podział nie uwzględnia. Nie ma to jednak większego znaczenia w wyciągnięciu zasadniczych wniosków z istnienia lub braku harmonii w rozwoju analizowanych trzech elementów. Porównując doświadczenia śląskie i krajowe nasuwa się wniosek, że największe korzyści uzyskiwano wówczas, gdy postęp w zakresie automatyzowania procesów technologicznych powodował harmonijną rozbudowę metod i środków komputerowo-wspomagane projektowania na szczeblu zakładu produkcyjnego oraz "informatyzowanie" procesów zarządzania zakładem produkcyjnym, a następnie (w zależności od potrzeb) doprowadzał do budowania komputerowo wspomaganego systemu projektowania maszyn na szczeblu organizacji przemysłowych wyspecjalizowanych w profilu produkcyjnym tych organizacji.

Systemy takie powstają więc w drodze systematycznego opanowywania coraz to nowych modułów projektowania, eksperymentowania i włączania ich w procesy produkcyjne. Warunkiem sukcesu jest ściśle przestrzeganie kultury pracy, uaktualnianie bazy danych technologicznych i konstrukcyjnych oraz stale, bliskie współdziałanie twórców nowych metod projektowania z użytkownikami. Systemów takich kupić jako gotowych nie można od nikogo - należy do nich dojść samemu. Są one z reguły unikalne i kosztowne, lecz w nowoczesnym przemyśle niezbędne i opłacalne.

Istotne zbliżenie tematyczne między automatyzacją procesów produkcyjno-technologicznych w przemyśle maszynowym a techniką komputerową nastąpiło z chwilą zastosowania układów numerycznego sterowania obrabiarkami. Początkowo były to stosunkowo proste sterowniki zrealizowane w technice cyfrowej, do których program sterujący (krok po kroku) był wprowadzany z czytnika taśmy papierowej.

Następnie wprowadzono układy komputerowego sterowania computer numerical control - CNC i układy komputerowego sterowania zespołami obrabiarek (direct numerical control - DNC), które poza rozdziałem zadań między poszczególne maszyny wykonywały również inne funkcje (sterowanie transportem, rejestracja wyników pracy i in.). Układy typu CNC i DNC zaczęły przejmować nie tylko funkcje sterowania tokarkami, frezarkami, wiertarkami i podobnymi maszynami do obróbki skrawaniem lecz również automatami technologicznymi i montażowymi. Technika komputerowa weszła kolejno w fazę produkcyjną w postaci komputerowych systemów kontrolno-pomiarowych i diagnostycznych. Dalszy postęp w konstruowaniu maszyn w technice komputerowej, mikroelektronice i automatyce wprowadził do procesów produkcyjno-technologicznych roboty przemysłowe i układy ograniczonego, a następnie optymalizacyjnego sterowania, stanowiące integralne części poszczególnych maszyn. Wraz z rozwojem wyższego integrowania się techniki komputerowej z procesami produkcyjno-technologicznymi rozwijało się wyposażenie komputerowe działów głównego konstruktora i technologa. Wyposażenie to coraz bardziej, poza funkcjami informacyjno-kontrolnymi, ukierunkowywało się na automatyzowanie projektowania procesów produkcyjno-technologicznych wykonywanych za pomocą komputerowo sterowanych maszyn. Kierunek tych przemian w metodach i środkach realizowania procesów produkcyjno-technologicznych jest w Polsce nazywany "komputerową automatyzacją procesów produkcyjnych" (odpowiednio w przemyśle maszynowym i innych), a w literaturze anglosaskiej computer aided manufacturing (CAM).

Wyżej wymienione środki wytwarzania wspomaganego komputerem są w hali produkcyjnej przemysłu maszynowego uzupełniano środkami informatyki, która przeważnie na szczeblu zakładu przeznaczona jest do automatyzowania procesów zarządzania i znajduje się w gestii służb ekonomiczno-administracyjnych. Tradycyjną formą wspomagania procesów produkcyjno-technologicznych ze strony systemów informatycznych jest chociażby techniczne przygotowanie produkcji, gospodarka magazynowa, narzędziowa itp. Łączne potraktowanie metod i środków komputerowej automatyzacji procesów produkcyjnych oraz metod i środków elektronicznej techniki obliczeniowej (informatyki), świadczącej usługi na rzecz procesów produkcyjno-technologicznych, prowadzi do hierarchicznego systemu, który w dużym uproszczeniu przedstawiony jest na rysunku 7. Prace badawcze nad urządzeniami automatyki, techniki pomiarowej i komputerowej, stanowiącymi elementy takiego systemu, oraz nad problemami



Rys. 7. Miejsce automatyzacji projektowania w hierarchicznym systemie komputerowej automatyzacji przedsiębiorstwa

systemowej ich współpracy i wdrożenia w przemyśle prowadzone są w Polsce od 1976 r. w ramach problemu węzłowego 06.1 (obiektywne systemy komputerowej automatyzacji - OSKA) koordynowanego przez Zjednoczenie MEERA Ministerstwa Przemysłu Maszynowego.

W skali światowej podobno prace rozpoczęto w latach 1965 - 1970, np. w USA w 1972 r. w 50 zakładach przemysłowych pracowały centra produkcyjne komputerowo sterowane. Jedną z trudności realizacyjnych takich systemów jest znaczna odmienność technik produkcyjno-technologicznych i techniki komputerowej. Ciekawe jest, że znani w świecie producenci maszyn matematycznych przez długi okres stronili od aktywnego włączenia się do realizowania takich systemów. Przyczyną mogła być obawa przed nowością i dobra koniunktura na sprzęt komputerowy zorientowany na automatyzowanie procesów zarządzania (sprzęt informatyczny). Z obserwacji przemysłu amerykańskiego wynika na przykład, że takie firmy jak Mc Donnell-Douglas, General Motors Company, United Aircraft i General Electric osiągnęły wysoki poziom komputerowo wspomaganego wytwarzania i komputerowo wspomaganego projektowania głównie dzięki własnym pracom rozwijającym układy CNC, DNC, roboty, minikomputery, technikę sterowania, metody i środki grafiki komputerowej, metody symulacyjne i podobne metody i środki. Systemy tego typu rozwijają się bardzo dynamicznie w przemyśle japońskim. W Japonii od 1967 r. szczególnie dynamicznie rozwijają się zastosowania robotów przemysłowych w połączeniu z innymi środkami automatyzowania procesów produkcyjno-technologicznych. Specjalna komisja powołana w Japonii w 1968 r., opierając się na analizie rynku pracy i jego specyfice w warunkach japońskich opracowała plan automatyzowania procesów produkcyjno-technologicznych przy maksymalnym eliminowaniu zrutynizowanych prac fizyczno-manipulacyjnych.

Znaczne osiągnięcia w tej dziedzinie obserwujemy w Anglii, Szwecji i w Republice Federalnej Niemiec.

Realizowanie obiektowych systemów komputerowej automatyzacji w różnych przemysłach jest oczywiście nie tylko problemem naukowo-badawczym i technicznym lecz problemem ekonomicznym i co najważniejsze społecznym. Od osiągnięć w tej dziedzinie zależy produktywność społeczna i kultura pracy, a więc jakość życia.

Dynamicznie rozwijają się prace nad takimi systemami w krajach RWPG. W Polsce prace te były na szerszą skalę realizowane od 1970 r. i dały liczne, praktyczne rezultaty wdrożone w zakresie numerycznie sterowanych obrabiarek, różnego rodzaju cyfrowych automatów technologicznych, sprzętu komputerowego, automatyki i aparatury kontrolno-pomiarowej.

Opracowano wiele metod i systemów, wśród nich wymienione poprzednio centrum obróbki korpusów. Integracja prac badawczych i wdrożeniowych z tego zakresu we wspomnianym problemie węzłowym 06.1 sprzyja dalszemu rozwojowi prac.

Biuro Polityczne PZPR oceniając w 1977 r. rozwój informatyki w naszym kraju, zaleciło zintensyfikowanie prac badawczych i konstrukcyjnych nad zastosowaniem techniki komputerowej w automatyzowaniu procesów produkcyjnych. Sprawą nauki i techniki jest zrealizowanie tego zalecenia.



dr inż. Stanisława DONKOWICZ-SITTAUER

mgr Janina OLECH

Instytut Maszyn Matematycznych

## Niektóre problemy małych banków danych dla komputerowo wspomaganego projektowania

### 1. Uwagi wstępne

W praktyce projektowania inżynierskiego, przy tradycyjnym sposobie rozwiązywania problemów, projektant zużywa 20-40% czasu na zbieranie informacji [4] np. na odszukiwanie i przeglądanie niezbędnych właściwych norm, katalogów, przepisów, wyszukiwanie i wybór metod, sprawdzanie czystości patentowej itp.

Nie więc dziwnego, że równoległe z pierwszymi próbami wykorzystywania komputerów do pewnych fragmentów obliczeń projektowych zaczęto zastanawiać się nad zautomatyzowaniem wyszukiwania potrzebnej projektantowi informacji [11], [9]. Prace te poszły jednak na ogół w kierunku tworzenia klasycznych systemów wyszukiwawczych, analogicznych np. do systemów obsługi bibliotek i podobnych banków danych.

Przyjęcie takiego kierunku poszukiwań nasuwało się przez to, że w swoich koncepcjach twórcy systemów wyszukiwawczych dla biur projektowych nie uwzględnili specyfiki danej dziedziny inżynierskiej. Oczywiście jest, iż próby stworzenia banku danych dla całej problematyki danego biura projektowego wymagały uwzględnienia bardzo rozległego zakresu informacji. Informacje tego typu są różnorodne, a ponadto forma większości z nich jest całkowicie niedostosowana do przetwarzania na zapis komputerowy. Z tych więc przyczyn prace nad komputerowym zapisem norm i katalogów praktycznie nie wyszły poza fazę prób.

Równocześnie, twórcy programów obliczeniowych nie mogąc czekać na wspomniane systemy wyszukiwawcze, rozwiązywali problemy po swojemu, stosując trzy sposoby:

- pewne wielkości po konsultacjach z projektantami i ewentualnie innymi specjalistami danej dziedziny przyjmowano w programie jako stałe; nawet jeżeli w istocie normy czy odpowiednie przepisy dopuszczały pewne różnice, wybierano np. wartość najbezpieczniejszą lub najbardziej prawdopodobną, lub też dobierano je według jakiegoś innego kryterium;
- wbudowywano w program pewną określoną liczbę różnych wartości danego parametru wraz z odpowiednim mechanizmem doboru właściwej wartości; w tej sytuacji użytkownik programu bądź musiał explicitnie podać wartość kryterialną, bądź też kryterium doboru było związane jednoznacznie z całym zestawem danych aktualnych; ten drugi sposób, wprawdzie nieco wygodniejszy dla użytkownika, często okazywał się tylko pozornie efektywniejszy, np. gdy wydłużał czas pracy programu;
- najprostszym (dla twórcy programu obliczeniowego) sposobem, było praktyczne ominięcie całego problemu i traktowanie wszystkich informacji jako danych aktualnych, a więc żądanie, aby użytkownik podawał je wszystkie, ale wówczas stosowanie komputerów w niczym nie zmniejszało kłopotów projektanta z wyszukiwaniem potrzebnych informacji, co gorsza, nieraz je zwiększało, gdyż precyzja określania takich czy innych wartości na użytek programu obliczeniowego często musi być większa niż przy tradycyjnym wykonywaniu tych samych obliczeń, tj. gdy projektant np. wybiera wartości z tablic, arbitralnie je interpolując lub zakręglając.

Zdanie sobie sprawy z powyższych tendencji stało się punktem wyjścia dla omawianych w artykule koncepcji MBD (mały bank danych).

Istotną przesłanką dla niniejszych rozważań było również przeświadczenie, iż w najbliższej (i zapewne nieco dalszej również) przyszłości najwłaściwszymi formami komputerowego wspomaganego projektowania będą systemy minikomputerowe problemowo lub obiektowo zorientowane. Stopień tej orientacji powinien być dość wysoki, co inaczej można sformułować, że powinny to być systemy o stosunkowo wąskiej specjalizacji tematycznej.

W świetle powyższych uwag wydaje się zupełnie uzasadnione, aby w skład takich systemów weszły nie tylko odpowiednie moduły licząco, nie tylko moduły czytania, kontroli danych i wydawania wyników, ale również własne, stosunkowo wysoko wyspecjalizowane zbiory danych stałych [6], [8]. Owa wysoka specjalizacja determinuje tak małą liczbę zawartych w tychże zbiorach informacji, że rzutuje to na formy organizacji tych zbiorów i ich przeszukiwanie. Fakt takiej specjalizacji jest pierwszym podstawowym założeniem niniejszej pracy.

Analizując zagadnienia organizacji zbiorów informacji normowo-katalogowych dla projektowania wspomaganego komputerem zaobserwowano, że informacje te mają zasadniczo charakter względnie stały, tzn. stosunkowo rzadko ulegają zmianom oraz że istotne znaczenie posiada aktualnie obowiązująca wartość informacji.

Ta cecha - względnej stałości danych - powoduje, że konieczność wprowadzenia jakiegokolwiek zmiany wymaga często równoczesnej reorganizacji zbioru. Wprowadzono zmiany - zarówno co do treści, jak i ogólnej ilości poszczególnych informacji - nie wymagają zapamiętywania stanów poprzednich bazy. Rejestr zmian bazy, prowadzony w klasycznych organizacjach banków danych, w tej organizacji nie jest więc konieczny. Głównym bowiem założeniem MBD jest aktualność zbioru bez konieczności zapamiętywania kolejnych jego zmian. Opisana tu właściwość stanowi drugie zasadnicze założenie niniejszej pracy.

Prowadzono w ramach prac studialnych w Zakładzie Programów dla Konstrukcji Technologii i Eksploatacji Maszyn INM dotychczasowe badania nad metodami tworzenia, organizacją i eksploatacją katalogowo-normowych zbiorów dla projektowania inżynierskiego wspomaganego komputerem doprowadziły do sformułowania koncepcji małego banku danych [8].

## 2. Omówienie pojęcia banku danych

Dotychczasowe rozwiązania związane z bankami danych i ich budową wywodzą się z założeń potrzeby oporowania dużymi (o dużej liczebności) zbiorami informacji. Przyczynę tego należy upatrywać w fakcie, że koncepcje banków powstały i nadal wiążą się głównie z dziedzinami, w których powstające wielkie ilości informacji pierwotnych muszą być wprowadzone do banku, wstępnie przetworzone i stale przechowywane. Duża jest również częstotliwość zmian i aktualizacji danych, co powoduje, że formy i metody aktualizacji stają się nieraz naczelnym problemem oprogramowania banku, a nie rzadko rzutują na samą organizację i strukturę przechowywania danych. Wreszcie - przeszukiwanie zasobów banku odbywa się na ogół pod kątem wyboru wielu informacji i ostatecznego ich przetwarzania w celu wydania żądanych zestawień.

Potrzeba tworzenia tak pojętych banków jest oczywistą konsekwencją szybkiego wzrostu ilości informacji w bardzo różnych dziedzinach oraz konieczności zapewnienia wielu użytkownikom, wielokrotnego korzystania z tej samej zasadniczo - aczkolwiek w szczegółach stale aktualizowanej - informacji. Należy wreszcie podkreślić, że koncepcja klasycznych banków danych zrodziła się dla dużych liczebnie zbiorów danych, których nie można było pomieścić ani w całości, ani nawet w odpowiednio dużych partiach w pamięci operacyjnej. Główny więc wysiłek twórców banków danych był i jest nadal skierowany na opracowanie takich organizacji zbiorów w pamięciach pomocniczych, aby zapewnić bezpośredni i szybki dostęp do poszczególnych informacji. Specyfika organizacji pamięci pomocniczych powoduje, iż stosuje się tu bardzo różnorakie rozwiązania typu: stosy, kolejki, listy, z różnorodnymi sposobami adresowania konkretnych informacji. Pojawiają się całe rozbudowane systemy indeksowania, katalogowania tych indeksów itp.

Drugą istotną cechą warunkującą rozwiązanie w klasycznym banku danych, jest konieczność pamiętania poprzednich stanów wartości poszczególnych informacji po wprowadzeniu odpowiednich aktualizacji. To znowu prowadzi do organizowania specjalnych zbiorów-katalogów wprowadzonych zmian wraz z pełnym rejestrem stanów poprzednich. Umożliwia to również odtworzenie aktualnej treści banku (bazy) w wypadku jej zniszczenia. Zostało to podyktowane faktem, iż na ogół treści takich banków są jedynym pełnym zbiorem informacji z danej dziedziny.

Na marginesie tych rozważań warto zaznaczyć, że sytuacja odpowiadająca klasycznym koncepcjom



banków danych, też może pojawić się w systemach komputerowo wspomaganego projektowania, a mianowicie w takich dużych, zazwyczaj obiektowo zorientowanych systemach, w których nad projektem jednego i tego samego obiektu pracują równocześnie różni specjaliści lub grupy specjalistów. Można wtedy tworzyć taką wspólną bazę danych całkowicie i jednoznacznie opisującą aktualny stan projektu, automatycznie rejestrującą postęp prac nad projektem, wszelkie zmiany i modyfikacje. Tego typu problemy nie są jednak przedmiotem niniejszego opracowania.

Opisane cechy charakterystyczne dla pojęcia klasycznego banku danych, nie występują, jak wynika z przytoczonych we wstępie rozważań, w odniesieniu do względnie stałych zbiorów katalogowo-normowych potrzebnych projektantom-konstruktorom. Problem ilości informacji, czy raczej liczebności poszczególnych zbiorów może być dokładniej omówiony dopiero przy analizowaniu organizacji bazy, co nie wchodzi w zakres niniejszego artykułu. Natomiast z drugą sprawą - pamiętanie historii zmian - wiążą się następujące fakty:

- częstotliwość i liczba koniecznych zmian w treści bazy jest stosunkowo mała;
- treść zbiorów danych stałych potrzebna jest tylko w pracach projektowych, a więc baza zawierająca te informacje musi nie tylko być zawsze aktualna, ale dostęp do obowiązującej aktualnie informacji nie może być utrudniony z powodu wprowadzanych uaktualnień;
- dane stanowiące treść bazy są w sposób trwały gromadzone w postaci odpowiednich wydawnictw i to jest ich stały obowiązujący pierwowzór, a forma zapamiętana w komputerze jest tylko formą pomocniczą, roboczą. Tak więc w sytuacji zniszczenia komputerowej bazy danych - odtworzyć ją będzie można na podstawie formy źródłowej.

Wracając do rozważań nad pojęciem klasycznych banków danych trzeba zauważyć, że problemy banku i bazy danych są terenem ciągłych twórczych penetracji zarówno teoretyków, jak i praktyków, a w literaturze [1], [2], [5], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30] pojawiło się wiele różnych definicji banku danych. Dowodzi to, że problem ten ciągle jest w sferze poszukiwań i eksperymentów i że jeszcze wiele problemów wymaga zbadania i uściślenia. Szczególnie charakterystyczny jest tu brak zgodności co do samej treści pojęciowej, tak podstawowych terminów, jak bank, czy baza danych. Propozycje definiowania tych pojęć wychodząco z różnych punktów widzenia są merytorycznie bardzo niejednolite, a co zatem idzie ciągle dyskusyjne. Ten stan ośmielił autorów do sformułowania własnych propozycji metodologicznych.

Nie podejmując niepotrzebnej dla niniejszego opracowania dyskusji terminologicznej można jednak wskazać następujące cechy charakteryzujące większość znanych koncepcji banku danych, a przydatnych dla niniejszych propozycji:

- istnieje baza danych - rozumiana jako zbiór lub zbiory ulokowane w pamięciach pomocniczych, bez względu na formę ich organizacji;
- istnieją specjalne metody (wyrażone w formie programów lub procedur programowych) zakładania tej bazy;
- istnieją różnorodno metody (a więc i programy) obsługi bazy, pozwalające na wykonywanie wszystkich potrzebnych operacji w bazie, a mianowicie:
  - dostęp do bazy,
  - przeszukiwanie, wybieranie i przetwarzanie danych wraz z formami wydawniczymi,
  - aktualizacja poszczególnych informacji lub ich partii, aż do modyfikacji całych zbiorów,
  - przetwarzanie informacji pierwotnej dla utworzenia nowej wraz z rozszerzeniem bazy o tę nową informację;
- musi być zapewniona dostępność informacji zawartych w bazie dla wielu różnych użytkowników.

Przyjmując ww cechy, jako niezmienniki pojęcia bank danych uznano, że stwierdzone we wstępie właściwości stałych informacji projektowych uzasadniają przyjęcie dla ich komputerowej organizacji pojęcia bank danych.

Natomiast omówiono już we wstępie oraz w niniejszym rozdziale cechy i właściwości zbiorów informacji katalogowo-normowych powodują konieczność wprowadzenia pewnych zmian jakościowych do klasycznego pojęcia bank danych, co znalazło swoje odbicie w pojęciu mały bank danych (MDD). Tak

więc w niniejszym opracowaniu przyjęto, że mały bank danych jest to specyficzna organizacja zbiorów jednorodnych tematycznie, zwanych dalej bazą danych i zestawu programów (bądź procedur) gospodarowania i wykorzystywania bazy zapewniających:

- założenie bazy o najlepszej, dla uwzględnionych informacji, strukturze,
- szybki i wygodny dostęp użytkownika do potrzebnej mu informacji,
- dokonywanie wszelkich operacji wyszukiwania i modyfikacji na zbiorach w chwili umieszczenia tych zbiorów w pamięci operacyjnej, a w pamięci pomocniczej wyłącznie przechowywanie aktualnej postaci tych zbiorów,
- nieprowadzenia żadnego rejestru zmian i historii treści bazy.

Zestaw takich założeń, spełniających wprawdzie wymienione uprzednio cechy charakteryzujące banki danych, w istotny jakościowo sposób odróżnia się od założeń niezbędnych dla organizacji klasycznych banków danych.

Jeszcze raz należy podkreślić, że na problem banku danych spojrzano tu pod kątem zapewnienia użytkownikowi takiej bazy danych, która w wąsko wyspecjalizowanym zagadnieniu dałaby mu kompletne informacje, a nie wymagała korzystania z bardzo rozbudowanego aparatu klasycznych banków danych.

### 3. Charakterystyka MBD

MBD, jak wynika z przyjętych na wstępie założeń, jest zbiorem informacji jednorodnych tematycznie, uwzględniających specyfikę problemów inżynierskich, których rozwiązaniom ma służyć. Zatem zakres zawartej w bazie informacji zależy będzie od stopnia specjalizacji tematycznej. Ustalenie tego zakresu wymaga indywidualnego podejścia do zagadnienia tworzenia MDD. Cechą wspólną w organizacji pracy nad MBD może być jedynie metodologia pracy, której zasady sformułowane będą po pewnym okresie doświadczeń.

#### O r g a n i z a c j a M B D

Założono również, jak zwykle w systemach użytkowych, że organizacja MBD musi być otwarta, tzn. iż we wszystkich jego częściach mogą być wprowadzane, w zależności od potrzeb i zmian sytuacji - zmiany i uzupełnienia.

Oprócz omówionych w punkcie 2 założeń dotyczących zasad tworzenia MBD, przyjęto jeszcze dwa odmiennej wprawdzie natury, lecz również mające istotne znaczenie dla dalszych prac nad koncepcją MDD. Przyjęto mianowicie, że MBD musi być przystosowany zarówno do pracy - eksploatacji w trybie konwersacyjnym, jak i wsadowym oraz że użytkownikiem może być bądź bezpośrednio człowiek, bądź inny program inżyniersko-obliczeniowy. W tej ostatniej sytuacji będzie się mówiło o wyszukiwaniu automatycznym.

W konsekwencji wszystkich tych założeń podstawowych przyjęto, że MBD będzie się składał z 2 części:

- z oprogramowania banku oraz
- z bazy danych.

Część pierwsza zawierać będzie 4 grupy programów. Pierwszą grupę stanowią będą programy informacyjne (jeden lub dwa), które dostarczą użytkownikowi informacji o zakresie tematycznym banku oraz o zestawie programów umożliwiających eksploatację bazy danych. Ten typ programów będzie służył przy konwersacyjnym korzystaniu z banku.

Następnymi grupami programów będą: zbiór programów przeszukiwania i wybierania danych z bazy wg różnych kryteriów, programy zakładania bazy, a wreszcie programy modyfikacji i reorganizacji bazy. Dwie ostatnie grupy programów będą wykorzystywane jedynie przez osoby do tego uprawnione (konserwator bazy).

Część druga MDD to baza danych, która może składać się conajwyżej z grup trzech typów informacji:

- ogólnych danych precyzujących treść bazy, do których odwołują się programy informacyjne części I,
- danych podstawowych stanowiących treść bazy,
- danych uzupełniających (rozszerzających) niektóre elementy danych podstawowych.

Taka organizacja bazy danych pomyślana jest, podobnie jak cała organizacja MBD, pod kątem konwersacyjnego korzystania z informacji. Uzyskiwana na wstępie informacja o treści bazy pozwoli użytkownikowi określić obszar jego poszukiwań.

### Treść bazy i zasady jej zakładania

Zgodnie z przyjętymi założeniami, treścią bazy danych MBD mogą być różnorodne stałe informacje wykorzystywane w taki lub inny sposób w konkretnym procesie projektowania. Wydaje się przy tym celowe wprowadzenie tu dwojakiego rodzaju klasyfikacji treści informacji projektowych: rzeczową i przedmiotową.

Klasyfikacja rzeczowa posłuży do podziału całego zbioru informacji związanych z danym procesem projektowym; i tak będziemy tu rozróżniać:

- a) informacje materiałowe: parametry charakteryzujące materiał, katalogi produkowanych materiałów, elementów, części, prefabrykatów itp.,
- b) informacje metodyczne obejmujące metody obliczeń czy doboru - obowiązujące, czyli regulowane normami lub przepisami, albo zalecane (podręcznikami, instrukcjami) - katalogi rozwiązań typowych lub przykładowych itp.,
- c) informacje prawno-patentowe, dotyczące danej dziedziny lub klasy problemów.

Charakter informacji zawartej w każdej z tych trzech grup jest zdecydowanie odmienny i łączenie ich w jednym banku nie wydaje się ani proste, ani konieczne, zwłaszcza że rozważając potrzeby, możliwości i zasadność tworzenia banku danych w aspekcie metod komputerowo wspomaganego projektowania w konkretnej dziedzinie za najpilniejsze należy uznać tworzenie specjalistycznych MBD dla informacji materiałowych.

Nadal jednak zarówno ilość informacji, jak i jej różnorodność są zdecydowanie za duże, aby mówić o MBD. Posłużymy się więc klasyfikacją przedmiotową, stosując podział procesów projektowych wg obiektów (przedmiotów) projektowanych w odniesieniu do każdej z grup. Dla każdego takiego obiektu (przedmiotu) projektowanego pełny zbiór informacji stałych obejmowałby wszystkie trzy klasy rzeczowe. Jednak w dalszych rozważaniach będziemy ograniczać się tylko do informacji rzeczowej klasy a), oraz do podziału jej wg kryteriów przedmiotowych. Ostatecznie, zgodnie z uwagami wstępnymi, ograniczymy nasze rozważania do projektowania konstrukcji inżynierskich. Dla przybliżenia tematu określimy pewne rodzaje tych obiektów. Tak więc będziemy mieli tu na myśli zarówno projektowanie konstrukcji maszyn, rozumianych jako urządzenia do wytwarzania np. obrabiarek, bądź też pojazdów; przy czym na użytek niniejszej pracy terminem tym obejmiemy nie tylko samochody i różnego rodzaju maszyny robocze, rolnicze, budowlane, przemysłowe itp., lecz również okręty i samoloty, jak i projektowanie szeroko pojętych konstrukcji budowlanych - hale fabryczne i fundamenty pod maszyny, kominy i maszty, mosty i inne obiekty budownictwa komunikacyjnego. Możemy tu również uwzględnić projektowanie silników od spalinowych po elektryczne oraz różnego rodzaju inne urządzenia elektryczne od urządzeń oświetleniowych czy transformatorów, aż po konstrukcje linii przesyłowych.

Każda z tych dziedzin wymaga specjalistycznego zestawu danych (informacji) stałych. I tak z innych materiałów konstruuje się obrabiarkę, z innych stację transformatorową, z innych tamę wodną, czy wreszcie silnik elektryczny itd. Jeżeli nawet niektóre materiały będą podobne, np. zarówno korpus obrabiarki, jak i ramę samochodu wytwarza się z pewnych gatunków stali, to zazwyczaj zgodność będzie ograniczała się do poszczególnych elementów w całym zbiorze informacji. A napewno odmiennie będą kryteria doboru itp.

Możemy więc założyć, że różnice w treści odpowiednich zbiorów parametrów i charakterystyk materiałowych będą tak duże, iż warto będzie dla każdego z nich wypracować właściwą organizację zapisu komputerowego.

- Powinno się zapewnić zarówno formę tych modułów jako programów samodzielnych wywoływanych przez użytkownika-projektanta dla uzyskania potrzebnej informacji, jak i formę procedury (sub-routiny, podprogramu), która może być włączona w inny program np. w program liczący. W tym drugim wypadku będzie się mówiło o automatycznym użytkowaniu bazy.
- Moduły wyszukiwawcze zorganizowane jako samodzielne programy, dla pracy konwersacyjnej powinny mieć dobrze rozbudowaną część dialogową, a dla pracy wsadowej powinny być wyposażone w dobrą instrukcję użytkownika. W obydwu sytuacjach musi być dobrze zorganizowana i przemyślana graficzna forma wydawnicza, tzn. że odszukana informacja musi mieć możliwość wyprowadzania wraz z pełną informacją opisową, dotyczącą danego przebiegu wyszukiwawczego, przy czym w eksploatacji konwersacyjnej wydawnictwo wyników musi być realizowane nie tylko na monitorze - gdzie opis objaśniający może być zredukowany, ale i na drukarce, gdzie informacja może być podana na życzenie użytkownika wraz z pełnym opisem objaśniającym.
- Należy przewidzieć możliwość - czyli opracować zasady, jak z modułów wyszukiwawczych typu procedura, użytkownik może proponować samodzielne programy wyszukiwawcze.
- Należy również opracować taki opis dokumentacyjny treści bazy, aby użytkownik mógł zaproponować nowe moduły wyszukiwawcze.

Absolutną jednak zasadą powinno być niedopuszczanie użytkownika-projektanta do ingerencji w treść bazy. I dlatego proponuje się drugi typ eksploatacji bazy nazwany tu roboczo konserwacją bazy.

Jak już podkreślano w niniejszym opracowaniu, pomimo stałości danych (informacji) stanowiących treść bazy należy przewidzieć możliwość wprowadzania pewnych zmian. Konieczność taka może być związana z wprowadzaniem zmian do obowiązujących norm i przepisów, mogą się także pojawiać całkowicie nowe materiały lub elementy, wreszcie przy zakładaniu bazy jakiegoś błędy lub pomyłki mogły być wychwycone przez kontrolny aparat programu zakładającego bazę, ale jako błędy zwykle pozostać do usunięcia, bądź w ogóle mogły być nie wychwycone i dopiero zauważone w eksploatacji. Można jednak przyjąć, iż takie zmiany są niewielkie, jak i niewielka jest częstotliwość takich zmian.

Zmiany można następująco sklasyfikować:

- zmiana wartości konkretnych informacji,
- usunięcie pewnych informacji z bazy,
- dodanie całkowicie nowych informacji,
- może też zaistnieć potrzeba całkowitego przeorganizowania pojedynczego zbioru lub całej bazy; ta jednak sytuacja należy raczej do działalności konserwatora bazy, gdyż wymaga wprowadzenia odpowiednich zmian w oprogramowaniu (punkt 4).

Tak więc typowe czynności konserwacji bazy obejmują jedynie trzy pierwsze czynności z podanych wyżej.

Należy więc dla ich realizacji opracować odpowiednie moduły aktualizacji i reorganizacji dostosowane do klasy modyfikacji. Moduły te mogą być bądź samodzielnymi programami, bądź procedurami pozwalającymi na zorganizowanie konkretnych programów modyfikacyjnych w zależności od zaistniałych potrzeb. Oczywiście, zgodnie z przyjętymi założeniami, wszelkie operacje przeprowadzane przez moduły modyfikacyjne będą realizowane wyłącznie w pamięci operacyjnej.

Tak więc zasadą ich działania jest cykl: sprowadzenie określonej porcji informacji do PAO, wprowadzenie zmian z uporządkowaniem danej porcji informacji oraz przygotowanie miejsca w pamięci pomocniczej, tak aby aktualna informacja mogła być bezpiecznie tam umieszczana. Oczywiście jest rzeczą, że moduły modyfikacyjne muszą zapewniać wyprowadzenie dokumentacji wprowadzanych zmian. Pamiętana w systemie bazy data wprowadzonych ostatecznie zmian jest jedyną formą pamiętania historii bazy.

Wprawdzie przewiduje się, że dla konserwacji bazy będzie przeznaczony specjalny personel działający w ścisłej współpracy z użytkownikami-projektantami, ale mimo to należy zadbać o właściwą instrukcję konserwatora bazy. Instrukcja taka przy pracy w trybie wsadowym jest jedynym przewodnikiem konserwatora bazy. Natomiast przy pracy w trybie konwersacyjnym powinna być zdublowana przez odpowiednio zorganizowany dialog programu z konserwatorem.

Jeszcze większą różnorodnością będą się charakteryzować zbiory typu katalogów części lub elementów. Całkowicie inaczej można zorganizować zapis dla profili walcowanych, inaczej dla katalogu elementów nośnych elektroenergetycznych linii przesyłowych, a inaczej wreszcie dla podzespołów pewnych urządzeń elektrycznych itp.

Przyjmując za punkt wyjścia przytoczone rozważania co do możliwych treści MDD dla projektowania, oczywiście staje się przyjęte uprzednio założenie co do względnej stałości tych zbiorów i poszczególnych zawartych w nich informacji. Wraz z założeniem o braku potrzeby pamiętania historii niezbędnych zmian, jak również założeniem, że zapis komputerowy nie jest podstawowy, można sformułować główne zasady zakładania bazy dla konkretnego MDD.

Otóż należy przyjąć, że organizacja bazy musi zależeć od wyników analizy treści bazy oraz zasad korzystania z bazy. Zakładanie bazy jest aktem jednorazowym realizowanym przez wyspecjalizowany program. Zadaniem tego programu powinno być - oprócz wczytania danych z nośników zewnętrżnych (np. z kart) - odpowiednie ich przeorganizowanie. Wynika to z faktu, iż formy zapisu wstępnego powinny być, dla uniknięcia pomyłek, najbardziej zbliżone do zapisu tradycyjnego, który jak już wspomniano, na ogół nie jest dostosowany do zapisów komputerowych. Można by proces dostosowania np. normy pozostawić poza komputerem, ale nie ma rozsądnych powodów, aby tak postępować. Niemniej powierzenie tej roli programowi powoduje, że trudno jest myśleć o jakimś uniwersalnym programie zakładania bazy. Raczej należy przewidywać rozwiązania unikalne związane z konkretnym MDD.

Kolejnym niezmiernie ważnym zadaniem programu zakładającego bazę powinno być przeprowadzenie dokładnej i wszechstronnej kontroli poprawności wczytywanych danych. Konieczność tych kontroli wiąże się ściśle z założoną stałością bazy i przewidywaniem niezmiernie małej częstotliwości wprowadzania zmian i modyfikacji. Metody tej kontroli, od całkowicie formalnych po różne stopnie kontroli merytorycznej, np. granice wartości poszczególnych parametrów, również zależą od konkretnych zbiorów danych, wpływają więc także na indywidualizację poszczególnych programów zakładania bazy. Należy podkreślić, że wykrywane przez program zakładający bazę błędy mogą być bądź "fatalne", to znaczy wstrzymujące dalszą pracę programu i wymagające po usunięciu błędów wprowadzania wszystkich danych od początku, bądź zwykłe. W tej sytuacji program zakładania bazy kontynuuje pracę, a na podstawie informacji lokalizującej błąd, konserwator bazy wprowadza poprawkę odpowiednim programem modyfikującym.

Kończąc te rozważania o programie zakładającym bazę należy jeszcze podkreślić, że zgodnie z przyjętymi założeniami o realizacji wszelkich działań związanych z danymi tylko w pamięci operacyjnej, program ten wszystkie wymienione zadania realizuje w PAO, a do pamięci pomocniczej odsyła całkowicie gotowe partie danych. Ponadto wyprowadza na zewnątrz (w postaci listingów) pełną informację o zawartości bazy lub o błędach.

#### Z a s a d y   e k s p l o a t a c j i   b a z y

Pod pojęciem eksploatacji bazy będziemy rozumieć dwie klasy czynności całkowicie - co należy podkreślić - rozłączne: użytkowanie bazy i jej konserwacja.

Użytkowanie bazy to wyszukiwanie w jej zasobach potrzebnej informacji. Może to być zarówno informacja pojedyncza, np. wartość konkretnego parametru konkretnego materiału lub pewien zbiór danych np. dotale o określonych cechach itp. Oczywiście jest, że sposoby i kryteria wyszukiwania informacji w zasobach bazy danych zależą od konkretnego procesu projektowego, dla którego MDD jest opracowywany, a w konsekwencji poszczególne programy wyszukiwawcze mogą być opracowane wyłącznie na podstawie analizy tego procesu. Jednak kilka spraw metodycznych można sformułować na podstawie znajomości ogólnych cech procesu projektowania konstrukcji inżynierskich oraz na podstawie założeń MDD.

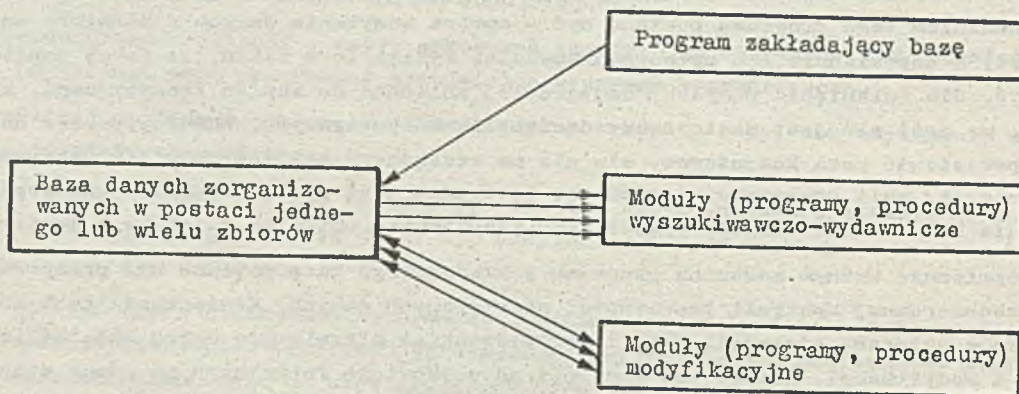
- Podstawową cechą modułów użytkowych jest ich wyłącznie wyszukiwawczy charakter. Dopiero odszukane elementy mogą podlegać dalszemu przetworzeniu, bądź przez moduł pomocniczy ściśle związany z danym modułem wyszukiwawczym, np. jego podprogram, bądź przez zupełnie inny luźno związany z wyszukiwawczym.
- Z zasady należy uwzględniać wiele różnych modułów wyszukiwawczych obsługujących poszczególne fazy procesu projektowego.

Na zakończenie warto nadmienić, że do zadań konserwatora bazy należy również zakładanie bazy, jak i pewna okresowa kontrola treści bazy, polegająca na pełnym wydruku informacji zawartych w bazie i porównywaniu ich z materiałami źródłowymi.

### Struktura MBD

Zgodnie z przyjętymi założeniami, MBD jest to zespół (system) składający się z bazy danych i z oprogramowania. Obie te części już scharakteryzowano, a tutaj będzie pokazane wzajemne ich powiązanie, zasady współdziałania oraz sposoby obu rodzajów eksploatacji banku.

Ogólną strukturę MBD pokazuje rys. 1.



Rys. 1. Struktura MBD

Kierunki strzałek na tym rysunku obrazują kierunek przepływu informacji. Z programu zakładającego bazę informacje idą wyłącznie do bazy. Z bazy idą do programów wyszukiwawczych, ale z nich nie do bazy nie wpływa. Natomiast moduły modyfikacyjne zarówno biorą informacje z bazy, jak i przesyłają do bazy. Szczegółowy tryb pracy poszczególnych klas programów ilustrują rysunki 2, 3, 4, 5.

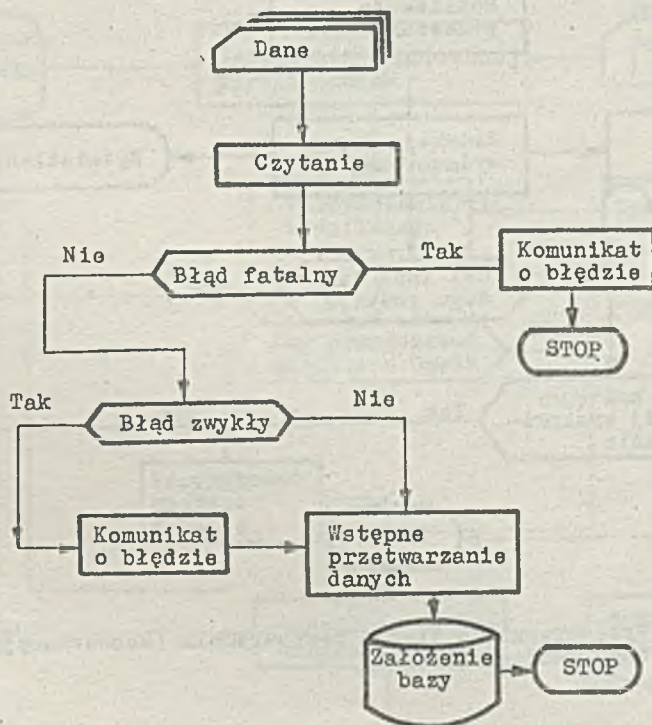
W odniesieniu do przedstawionych na rys. 3 i 5 prac wyszukiwania bezpośredniego lub konserwacji bazy, trzeba jeszcze dodać, że jeżeli prace te odbywają się w trybie konwersacyjnym, to przedstawione na tych rysunkach wprowadzanie wstępnych informacji z kart jest zastąpione wejściem z klawiatury monitora ekranowego. Pozostają natomiast nadal wydruki jako forma dokumentacji trwałej, zdublowane, rzecz jasna, wyświetlaniem informacji na ekranie monitora.

#### 4. Konserwacja MBD

Pokazane na rysunkach 3, 4, 5 zasady eksploatacji bazy oraz na rys. 2 zasady jej zakładania, świadczą o wielkiej roli, jaką pełnią poszczególne moduły oprogramowania banku. Dbanie o ich poprawność i niezawodność oraz jak najlepszą merytoryczną sprawność, leży w gestii konserwatora banku, w odróżnieniu od konserwatora bazy, który ma prawo ingerencji w treści przechowywanych w banku informacji, ale posługuje się w tym celu gotowymi programami (lub modułami) bez prawa ingerencji w treść oprogramowania; kompetencje konserwatora banku są dużo szersze [7]. Jego zdaniem jest:

- okresowe testowanie poprawności całego oprogramowania banku,
- wprowadzanie niezbędnych poprawek do oprogramowania,
- rozbudowa oprogramowania banku, zarówno poszczególnych modułów wyszukiwawczych lub modyfikacyjnych, jak i tworzenie nowych modułów,
- uaktualnianie dokumentacji banku,

- utrzymywanie kontaktu z projektantami oraz analiza komunikatów otrzymywanych z maszyny w celu poszerzenia programów o zgłaszane słuszne postulaty dotyczące zmian w banku,
- śledzenie zmian obowiązujących przepisów (w celu ich uwzględnienia) nie tylko dotyczących treści bazy, ale i treści poszczególnych programów. Ma to znaczenie zwłaszcza przy modułach wyszukiwawczych, gdzie zmiany kryteriów mogą być tak duże, iż wpływają nie tylko na zmianę organizacji tych modułów, ale nawet na zmianę w programach liczących, przy wyszukiwaniu automatycznym.



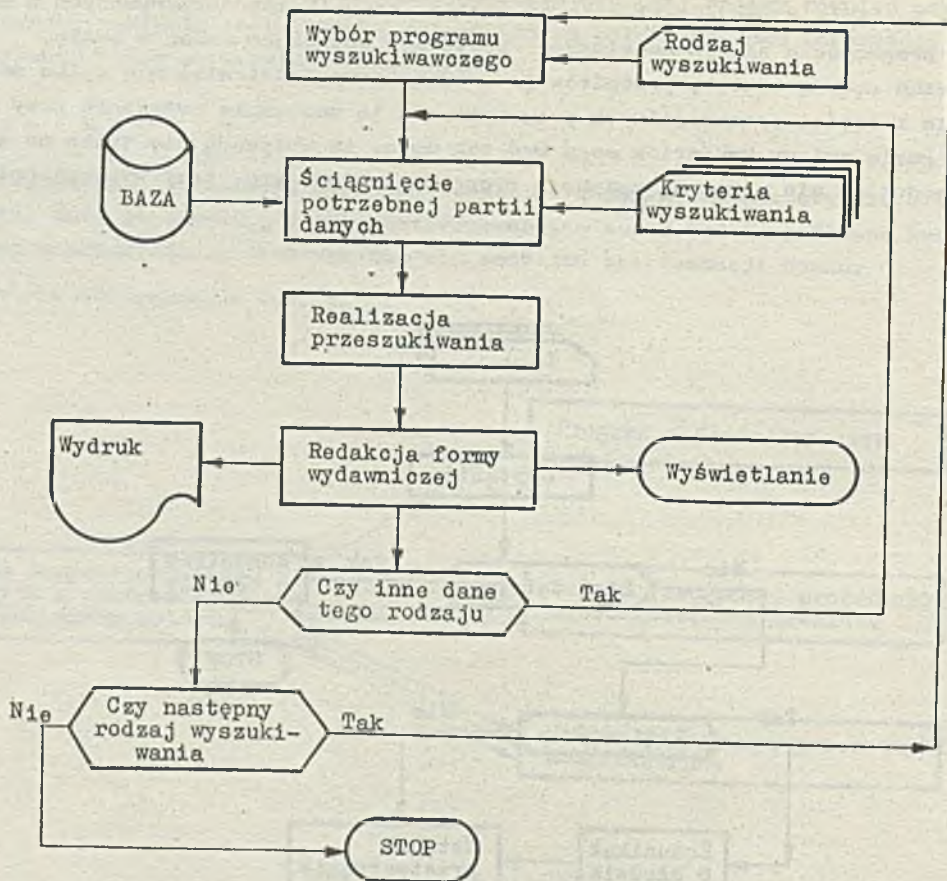
Rys. 2. Praca programu zakładania bazy

### 5. Wnioski końcowo

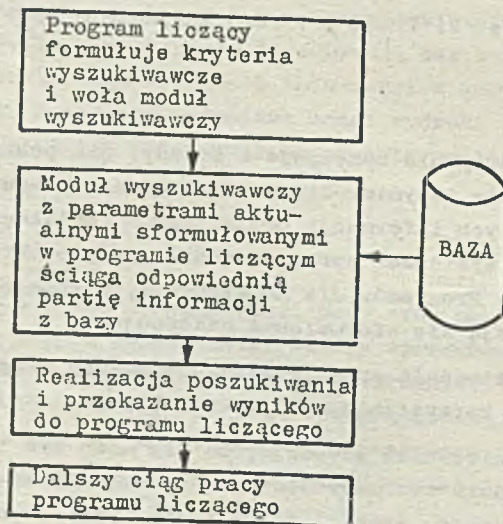
Przedstawiono w niniejszym artykule koncepcje i zasady, jak powiedziano na wstępie, wypływają z analizy problemów projektowania inżynierskiego wspomaganego komputerem, a ściślej z potrzeby takiej właśnie organizacji stałych informacji projektowych (katalogowo-normowych), stwierdzonej w trakcie prac nad różnorodnymi systemami oprogramowania dla projektowania wspomaganego komputerem, prowadzonych od lat w Zakładzie Programów dla Konstrukcji, Technologii i Eksploatacji Maszyn IMM. Wnioski jakie stąd wypływają dają się sformułować następująco:

- mało banki danych dla projektowania inżynierskiego wymagają uwzględnienia specyfiki tej dziedziny, stąd nie jest celowe korzystanie z klasycznych organizacji banków danych,
- różnorodność zagadnień inżynierskich powoduje, że nie może być jedna uniwersalna organizacja MBD, natomiast taki bank będzie tworzony dla każdego zagadnienia, a jego struktura wewnętrzna uzależniona od treści informacji. Stwarza to potrzebę wypracowania ogólnych metod podejścia do projektowania MBD oraz ogólnych zasad organizacji i eksploatacji MBD.

Niektóre z tych zagadnień omówiono w niniejszym artykule, pozostaje jednak niezmiernie więcej problemów nie tylko nie rozwiązanych, ale nawet nie zasygnalizowanych. Dlatego celowe wydaje się prowadzenie prac eksperymentalnych, w trakcie których wykrystalizują się poglądy i zbierze się materiał do dalszych dyskusji.

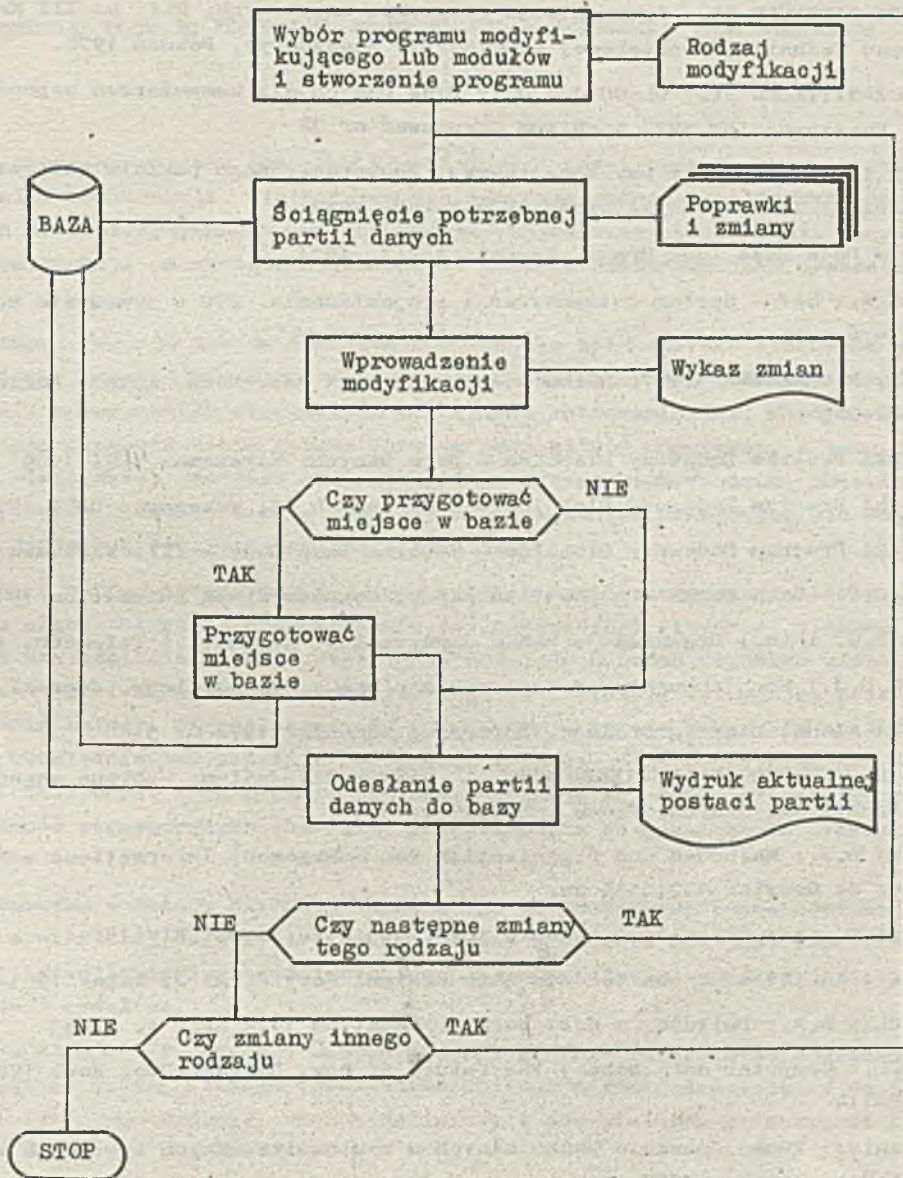


Rys. 3. Wyszukiwanie w trybie bezpośrednim (konwersacyjnym bądź wsadowym)



Rys. 4. Wyszukiwanie automatyczne





Rys. 5. Konserwacja bazy

Literatura

- [1] ABRIAL J.R.: Data Semantics, IFIP-TC-2 Working Conference on "Data Base Management Systems". Cargese Corsica 1974
- [2] AIRENS F.: Das Datenbankdiagramm - ein Mittel zur Darstellung von Datenstruktur und Speicherungsstruktur einer Datenbank. Computer Praxis 1973-nr 6 s. 166-170
- [3] BĄBIŃSKI Cz.: Elementy nauki o projektowaniu. Warszawa: WNT 1969
- [4] BĄBIŃSKI Cz.: Systemy projektujące. I ogólnopolska konferencja metodologii projektowania, 1971 Warszawa: PWN 1973

- [5] BIELAK A., SKULSKI K.: Systemy banków danych. Warszawa: OBRI 1975, Problemy informatyki
- [6] BONKOWICZ-SITTAUER St.: Projektowanie wspomaganie w małych ośrodkach projektowych. V konferencja Instytutu Cybernetyki Technicznej, Międzygorze, październik 1977. Prace naukowe ICT Politechniki Wrocławskiej 1977 nr 49
- [7] BONKOWICZ-SITTAUER St.: Dokumentacja programów użytkowych. Ref. na III Krajowy Przegląd Zastosowań Techniki Komputerowej w Przemysle Maszynowym, Poznań 1976.
- [8] BONKOWICZ-SITTAUER St., OLECH J.: Mały Bank Danych dla komputerowo wspomaganego projektowania. Warszawa: IMM 1977 Archiwum Opracowań nr 32
- [9] BUCKI J. i in. - Projekt użytkowej wersji konwersacyjnego pakietu obliczeń numerycznych. Sosnowiec: IMM - Oddział Śląsk 1976 oprac. wewn.
- [10] CODASYL - Data Base Task Group. Report, April, 1971
- [11] DROŻNIAK A.: SAP - System automatyzacji projektowania. ETO w przemyśle budowlanym 1972 nr 5 - 6.
- [12] ELMAN J., MICHALICEK V.: Technika a aplikacje bank dat. Mech. Autom. Administr. 1973 nr 9 s. 382-385
- [13] Europejski Program Badawczy Diebolda - Bank Danych. Warszawa: OBRI 1969
- [14] Europejski Program Badawczy Diebolda - Bank Danych II. Warszawa: OBRI 1971
- [15] Europejski Program Badawczy Diebolda - Wspólna Baza Danych III. Warszawa: OBRI 1972
- [16] FORSTER C.C.: Data banks - a position paper. Computers and Automation 1971 nr 3 s. 28-30
- [17] GLEJZER I.G. i in.: Organizacja banku danych ASV. Avtomat. i Telemekh. 1973 nr 7
- [18] IRONSDIDE D.J.: Some thoughts on uses of data banks. Canad. Lubr. Journal 1971 nr 1 s.30-35
- [19] KADERKOWA Alona: Bank Informacji. Inform. i Komunik. 1972 nr 3
- [20] KOŁACKA D., WIECZOREK A., WIERZBOWSKI J., WIŚNIEWSKI Julian: Wybrano zagadnienia związane ze strukturami danych. Prace IMM 1974 nr 1
- [21] KOREIMANN D.S.: Methoden und Organisation von Management Informations - Systemen. Berlin: Walter de Gruyter 1971 154 ss.
- [22] KOWALCZYK Edward: Człowiek w świecie informacji. Warszawa: KIW 1974
- [23] LYON J.K.: An introduction to data base design. Nowy Jork: J. Wiley 1971 81 ss.
- [24] Mc LAUGHLIN R.A.: Building a data base. Datamation 1972 nr 7 s. 51-55
- [25] NOLAN R.L.: Computer data bases: the future in now. Harvard Bus. Rev. 1973 t. 51 nr 5 s. 98-114
- [26] OLECH Janina: Funkcjonowanie banku danych w zautomatyzowanych systemach zarządzania. Wiadomości Statystyczne 1975 nr 5
- [27] POSSELT Jan: Uwagi o organizacji banku danych. Informacje i Komunikaty 1972 nr 3
- [28] PLESCH M., GRISE J.: Eigenschaften von Datenbanksystemen in Vergleich. Angewandte Informatik 1972 nr 11 s. 489 - 498
- [29] SCHUBERT R.F.: Basic concepts in data base management systems. Datamation 1972 nr 7 s. 42-47
- [30] SKRONN H.J.: Methoden der Strukturierung von Datenbanken. Angewandte Informatik 1973 nr 5 s. 204-210
- [31] System zarządzania Bazą Danych SAD. Warszawa: IMM 1976, oprac. wewn.

mgr Adam LUTONORSKI  
Instytut Maszyn Matematycznych

## Systemy metody elementu skończonego - podstawy matematyczne

### 1. Wstęp

Metoda elementu skończonego jest sposobem przybliżonego rozwiązywania skomplikowanych problemów statyki i dynamiki konstrukcji, dla których nie istnieją gotowe, analityczne rozwiązania, a wielka ilość danych oraz otrzymywanych wyników wymaga użycia maszyny cyfrowej. Pomysł metody polega na podziale konstrukcji na rozłączne elementy i badania continuum jako dyskretnego układu tych elementów.

W początkowym etapie rozwoju metoda elementu skończonego była szeroko stosowana jako efektywna metoda obliczeniowa, chociaż nie poparta głębszymi badaniami podstaw matematycznych oraz bez poważniejszych analiz numerycznych stosowanych algorytmów obliczeniowych. Wkrótce jednak stała się przedmiotem zainteresowania wielu matematyków, a szczególnie numeryków. W toku prowadzonych intensywnych badań stwierdzono, że jest ona szczególnym przypadkiem znanych, obecnie często nazywanych klasycznymi, metod Ritz-Galerkina rozwiązywania zagadnień brzegowych dla cząstkowych równań różniczkowych, których krótki opis podano w punkcie 3.

Początkowy sukces i bardzo intensywny rozwój metody elementu skończonego należy przypisać szczególnie prostej algorytmizacji, programowaniu i interpretacji fizycznej wyników. Zalety te wymagały stosownego uzasadnienia numerycznego; w toku dalszych badań uzyskano wiele ważnych wyników w teorii aproksymacji i metodach numerycznych algebry liniowej, otrzymując precyzyjne, matematyczne sformułowanie metody. Sformułowanie to wykorzystuje przede wszystkim analizę funkcjonalną, ze szczególnym uwzględnieniem teorii aproksymacji w funkcyjnych przestrzeniach Sobolewa oraz teorię równań różniczkowych. W tych właśnie terminach opisuje się cechy asymptotyczne metody charakteryzujące zbieżność otrzymywanych rozwiązań przybliżonych do rozwiązania prawdziwego oraz tzw. rząd zbieżności.

Rozpatrując zastosowanie metody elementu skończonego do analizy wytrzymałościowej konstrukcji sprężystych należy stwierdzić, że jest to zadanie trudne a ponadto nie jest odpowiednio poglądowym przykładem przy przedstawieniu zasadniczych cech metody, dlatego w dalszej części przedstawiamy prostsze zadanie modelowe.

Rozważane zadanie modelowe to znane i dobrze zbadane płaskie zagadnienie brzegowe dla operatora eliptycznego drugiego rzędu, którego rozwiązanie opisuje drgania zamocowanej na brzegu membrany. Ścisłe sformułowanie tego zadania, wraz z definicjami odpowiednich przestrzeni funkcyjnych podano w punkcie 2.

W praktycznych zastosowaniach, a u nas w modelowym zadaniu, wykorzystujemy wariacyjne sformułowanie zadania różniczkowego. Polega ono na minimalizacji tzw. funkcjonału energii w pewnej klasie funkcji. Uzyskiwane przez nas rozwiązanie przybliżone jest wynikiem minimalizacji funkcjonału energii w węższej klasie funkcji stanowiących skończenie elementowe aproksymacje funkcji z dziedziny funkcjonału energii. Jest to program postępowania analogiczny do metod Ritz-Galerkina. Punktem decydującym o jakościowych różnicach między metodami klasycznymi a metodą elementu skończonego jest określenie wspomnianych skończenie elementowych aproksymacji funkcji z dziedziny minimalizowanego funkcjonału energii. W tym celu obszar, w którym rozpatrujemy zagadnienie brzegowe dzielimy na pewną liczbę rozłącznych elementów, przy czym najczęściej są to trójkąty lub prostokąty. Przybliżonego rozwiązania naszego zadania w postaci wariacyjnej poszukujemy w postaci liniowej kombinacji  $\sum_{j=1}^n q_j \varphi_j$  bazowych funkcji  $\varphi_j$ , które wewnątrz każdego z elementów są wielomianami na ogół niskiego stopnia i są różne od zera w co najwyżej kilku przyległych elementach. W pewnych wypadkach, np. w teorii sprężystości, ze względu na ich interpretację fizyczną - funkcje nazywa się funkcjami kształtu. W ten sposób będzie określona owa węższa klasa skończenie elemen-

towych aproksymacji. Znalazienie minimum funkcjonału energii w zbiorze tych właśnie funkcji prowadzi się do zadania z algebry liniowej tzn. do rozwiązania wielkiego na ogół układu równań liniowych  $Kq = F$  gdzie  $Q = /q_1, \dots, q_n/^T$ . Macierz  $K$  zwana macierzą sztywności jest rozrzedzona i pasmowa, a wektor  $F$  zwany jest wektorem uogólnionych obciążeń. Sposoby rozwiązania tego zadania podano w punkcie 6. Współrzędne  $q_1$  rozwiązania  $q$  określają współczynniki w poszukiwanej przez nas kombinacji liniowej  $\sum_{j=1}^n q_1 \varphi_j$ , która jest przybliżonym rozwiązaniem naszego zadania różniczkowego generowanym przez metodę elementu skończonego.

Metoda elementu skończonego jest obecnie jedną z intensywnie stosowanych i badanych metod numerycznych. Istniejących obecnie programów skończenie elementowej analizy konstrukcji używa się z powodzeniem w budownictwie, w analizie wytrzymałościowej konstrukcji okrętowych i lotniczych. W przyszłości powinny one znaleźć zastosowanie w automatyzacji projektowania maszyn.

Zamieszczona bibliografia zawiera wiele dostępnych w literaturze pozycji na dany temat, przy czym bibliografie w pozycjach [5], [8], [9], [11] stanowią wyczerpującą listę publikacji dotyczących matematycznej analizy metody elementu skończonego. Odwołanie się do zawartych tam informacji wyjaśni szczegółowo problemy poruszone w niniejszym opracowaniu.

## 2. Zagadnienie brzegowe dla operatora eliptycznego i jego wariacyjne sformułowanie w przestrzeniach Sobolewa

Poszukujemy przybliżonego rozwiązania następującego, poprawnie postawionego zadania, które nazwiemy zagadnieniem brzegowym:

$$Au(x_1, x_2) = - \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} \left( a_{ij}(x_1, x_2) \frac{\partial u(x_1, x_2)}{\partial x_j} \right) + cu(x_1, x_2) = f(x_1, x_2) \quad \text{dla } (x_1, x_2) \in \Omega$$

$$u(x_1, x_2) = 0 \quad \text{dla } (x_1, x_2) \in \partial\Omega \quad (1)$$

gdzie o operatorze różniczkowym  $A$  zakładamy, że jest eliptyczny w  $\Omega$ , tzn.

$$a_{ij}(x_1, x_2) = a_{ji}(x_1, x_2), \quad \sum_{i,j=1}^2 a_{ij} \xi_i \xi_j \geq \mu \sum_{i=1}^2 \xi_i^2, \quad (\xi_1, \xi_2) \in \mathbb{R}^2$$

gdzie  $\mu = \text{const} > 0$ ,  $c(x_1, x_2) \geq 0$  oraz  $a_{ij}$ ,  $c$ ,  $f$  są funkcjami ciągłymi w  $\bar{\Omega}$ , wreszcie  $\Omega$  jest ograniczonym obszarem w płaszczyźnie z brzegiem  $\partial\Omega$ .

Wprowadzimy definicje następujących przestrzeni funkcyjnych  $W_2^k(\Omega)$ , zwanych przestrzeniami Sobolewa.  $W_2^k(\Omega)$  oznacza przestrzeń funkcji mających pochodne uogólnione, aż do rzędu  $k$  włącznie, które należą do przestrzeni  $L_2(\Omega)$  funkcji całkownych z kwadratem. Normę w przestrzeni  $W_2^k(\Omega)$  określamy następująco:

$$\|u\|_{W_2^k(\Omega)}^2 = \sum_{|i| \leq k} \|D^i u\|_{L_2(\Omega)}^2$$

gdzie  $i = (i_1, i_2)$   $|i| = i_1 + i_2$

$$D^i u = \frac{\partial^{|i|} u}{\partial x_1^{i_1} \partial x_2^{i_2}}$$

Zdefiniujemy dodatkowo podprzestrzeń  $W_2^k(\Omega)$  w przestrzeni  $W_2^k(\Omega)$  jako uzupełnienie w normie  $\|\cdot\|_{W_2^k(\Omega)}$  zbioru funkcji z  $C_0^k(\Omega)$  o nośniku zwartym leżącym w  $\Omega$  /patrz [1], [3]/. Okazuje się że rozwiązanie  $u \in W_2^k(\Omega)$  zagadnienia brzegowego (1) jest identyczne z funkcją, która minimalizuje funkcjonał

$$I(v) = \int_{\Omega} \left( \sum_{i,j=1}^2 a_{ij} \frac{\partial v(x)}{\partial x_i} \frac{\partial v(x)}{\partial x_j} + cv^2(x) - 2f(x)v(x) \right) dx \quad (2)$$

w klasie  $W_2^1(\Omega)$ ,  $x \in \Omega$ .

Zadanie znalezienia minimum funkcjonału  $I$  w klasie  $W_2^1$ , równoważne zadaniu (1), nazywamy wariancyjnym sformułowaniem zagadnienia brzegowego dla operatora eliptycznego  $A$  [3], [4], [5].

Tego typu sformułowania możemy otrzymać również dla innych zagadnień brzegowych, tzn. innych operatorów eliptycznych i innych /niż  $u(x_1, x_2) = 0$  dla  $(x_1, x_2) \in \partial\Omega$  / warunków brzegowych. Na przykład drgania membrany są opisane przez (1), gdy  $A$  jest operatorem Laplace'a, w sformułowaniu wariancyjnym minimalizują funkcjonał

$$I(v) = \int_{\Omega} \left( \left( \frac{\partial v(x)}{\partial x_1} \right)^2 + \left( \frac{\partial v(x)}{\partial x_2} \right)^2 \right) dx$$

### 3. Klasyczne metody przybliżonego rozwiązywania zagadnienia brzegowego

Załóżmy, że równanie (1) zachodzi dla funkcji  $u \in S$ , poprzednio rozważaliśmy przestrzeń rozwiązań  $S = W_2^1(\Omega)$ . Mnożąc obie strony (1) przez dowolną funkcję  $v$  należącą do pewnej przestrzeni funkcji testowych  $V$  i całkując po  $\Omega$  obie strony otrzymujemy

$$(Au, v) = (f, v) \quad (3)$$

gdzie  $(u, v) = \int_{\Omega} u(x)v(x)dx$  oznacza iloczyn skalarny w przestrzeni  $L_2(\Omega)$ .

Zadanie znalezienia funkcji  $u \in S$  dla każdego  $v \in V$  spełniającej (3) nazywamy słabym sformułowaniem zadania (1).

Jeżeli  $V$  zawiera wszystkie  $\delta$  funkcje wówczas mówimy, że  $u$  spełnia równanie (1) w sensie klasycznym. Zakładając, że  $V = C_0^\infty(\Omega)$  i całkując przez części lewą stronę (3) otrzymujemy

$$(u, A^*v) = (f, v)$$

gdzie  $A^*u(x) = - \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} (a_{ji}(x) \frac{\partial u(x)}{\partial x_j}) + cu(x) = Au(x)$  bo  $a_{ij} = a_{ji}$

i mówimy, że  $u$  spełnia równanie (1) w sensie dystrybucji. Rozwiązanie zadania (3) w wypadku pośrednim, gdy  $S = V = W_2^1(\Omega)$  pokrywa się z rozwiązaniem zadania (2).

Rozpatrzmy  $n$ -wymiarowe podprzestrzenie  $S^n \subset S$ ,  $V^n \subset V$  o bazach złożonych odpowiednio z funkcji  $\varphi_1, \dots, \varphi_n$  oraz  $\psi_1, \dots, \psi_n$ . Metoda Galerkin przybliżonego rozwiązania zadania (3) polega na znalezieniu

$$u_n = \sum_{j=1}^n q_j \varphi_j \in S^n$$

takiego, aby dla każdego  $v \in V^n$

$$(Au_n, v) = (f, v)$$

Powyższe równanie równoważne jest układowi  $n$  równań liniowych

$$\left( \sum_{j=1}^n q_j \varphi_j, \psi_k \right) = (f, \psi_k) \quad k = 1, \dots, n \quad (4)$$

Układ (4) w postaci macierzowej ma postać

$$Kq = F$$

gdzie  $K_{k,j} = (A\varphi_j, \psi_k)$  wektor  $F$  ma współrzędne  $F_k = (f, \psi_k)$ , a wektor  $q$  ma współrzędne  $q_k$ ,  $k=1, \dots, n$ .

W szczególnym wypadku, gdy  $S^n = V^n \subset W_2^1$  i  $\varphi_j = \psi_j$  metoda Galorkina znana jest jako metoda Ritza.  $K$  nazywa się wtedy macierzą sztywności.

Współczynniki  $q_i$   $i = 1, \dots, n$  stanowią rozwiązanie układu (4) i określają przybliżone rozwiązanie  $u_n = \sum_{i=1}^n q_i \varphi_i$  zadania (2).

Przy praktycznym zastosowaniu wymienionych metod podstawowym zadaniem jest określenie skończonego wymiarowego przestrzeni  $S^n$ . W klasycznych [6], [2] konstrukcjach tych przestrzeni macierze układów (3) i (4) zwane macierzami sztywności były pełne, czego konsekwencją były duże błędy w znajdowaniu współczynników  $q_i$   $i = 1, \dots, n$ .

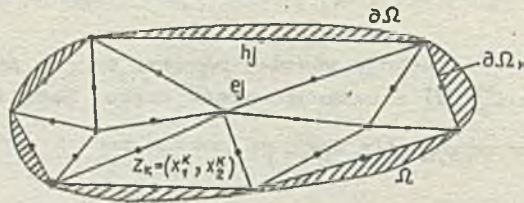
Metoda elementu skończonego stanowi szczególny przypadek metod typu Ritza-Galorkina. Jako funkcje bazowe  $\varphi_j$  wybieramy mianowicie funkcje znikające w całym obszarze  $\Omega$  z wyjątkiem ustalonej liczby elementów, na które obszar ten podzieliliśmy. Ta własność funkcji bazowych sprawia, że ich konstrukcja jest łatwiejsza w obszarach o skomplikowanym kształcie, a otrzymana macierz sztywności jest rozrzedzoną macierzą pasmową o niewielkim wskaźniku uwarunkowania.

Tak więc istotą metody elementu skończonego jest odpowiedni wybór funkcji bazowych [7] przestrzeni, a w zadaniach (2) i (4) aproksymującej przestrzeni  $W_2^1(\Omega)$ .

#### 4. Konstrukcja funkcji bazowych w metodzie elementu skończonego

##### Elementy trójkątne

Rozpoczynamy od podzielenia obszaru  $\Omega$  na pewną liczbę nie zachodzących na siebie części /o-  
elementów/  $e_j$ , na ogół trójkątów lub prostokątów leżących tak, że każde dwa przyległe elementy mają albo wspólny wierzchołek, albo wspólny bok.



Najdłuższy bok tego trójkąta  $j$  oznaczamy  $h_j$  i  $h = \max h_j$ . Można podać rozmaite algorytmy przeprowadzenia automatycznie takiej triangulacji, jak na rysunku.

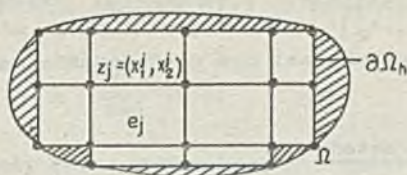
Skończenie wymiarową podprzestrzeń  $S^n \subset W_2^1(\Omega)$  określimy jako przestrzeń funkcji, które na każdym z trójkątów  $e_j$  są kwadratowym wielomianem  $w(x_1, x_2) = \alpha_1 + \alpha_2 x_1 + \alpha_3 x_2 + \alpha_4 x_1^2 + \alpha_5 x_1 x_2 + \alpha_6 x_2^2$  i są równe zeru na brzegu  $\partial\Omega$ . Wobec tego każda funkcja  $v \in S^n$  może być zatem jednoznacznie określona wewnątrz każdego z trójkątnych elementów przez podanie wartości  $v$  w wierzchołkach i środkach boków  $e_j$  punkty oznaczone na rysunku kropkami. Rzeczywiście  $S^n \subset W_2^1(\Omega)$  gdyż funkcje  $v$  mają początkowe pochodne pierwszego rzędu kawałkami ciągle oraz przyjmują te same wartości na wspólnych dla dwóch trójkątów bokach, co oznacza, że są to funkcje ciągle. Wymiar przestrzeni  $S^n$  równy jest liczbie tzw. węzłów swobodnych, tzn. nie leżących na brzegu  $\partial\Omega$ . Bazę przestrzeni  $S^n$  stanowią funkcje  $\varphi_i \in S$  takie, że

$$\varphi_i(z_k) = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \quad (5)$$

gdzie  $z_k$  oznaczają wszystkie węzły swobodne w  $\Omega$ .

##### Elementy prostokątne

Konstrukcja funkcji bazowych przebiega podobnie na siatce prostokątnej, tzn. wtedy, gdy obszar  $\Omega$  dzielimy na przylegające do siebie całymi bokami prostokąty  $e_j$ . Najdłuższy bok tego prostokąta  $e_j$  oznaczmy  $h_j$  i  $h = \max h_j$ .



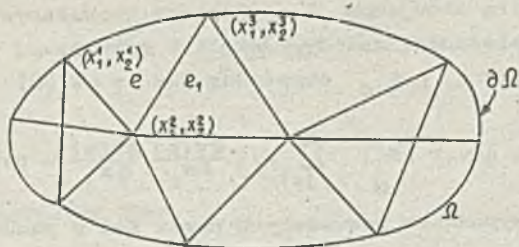
Określamy skończenie wymiarową podprzestrzeń  $S^n \subset \overset{0}{W}_2^1(\Omega)$  jako przestrzeń funkcji kawałkami bi-  
liniowych, tzn. takich, że na każdym z prostokątów  $e_j$  są wielomianem  $w(x_1, x_2) = \alpha_1 + \alpha_2 x_1 +$   
 $+ \alpha_3 x_2 + \alpha_4 x_1 x_2$  i są równe zero na brzegu obszaru  $\partial\Omega_h$ . Wobec tego każda funkcja  $v \in S^n$   
może być jednoznacznie określona wewnątrz każdego z elementów  $e_j$  przez podanie jej wartości w  
węzłach tzn. wierzchołkach  $z_j$  prostokątów. Podobnie, jak poprzednio można zauważyć, że istot-  
nie  $S^n \subset \overset{0}{W}_2^1(\Omega)$ . Baza  $S^n$  stanowią funkcje  $\varphi_j \in S^n$  takie, że

$$\varphi_j(z_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} \quad (6)$$

gdzie  $z_k$  oznaczają wszystkie węzły swobodne, tzn. węzły należące do wnętrza  $\Omega$ , a więc wymiar  
 $S^n$  jest równy liczbie swobodnych węzłów w  $\Omega$ .

### Elementy krzywoliniowe

W poprzednich przykładach konstrukcji funkcji bazowych na trójkątach i prostokątach brzeg  $\partial\Omega$   
obszaru  $\Omega$  aproksymowaliśmy łamaną, co niestety nie gwarantuje uzyskania dobrych przybliżeń po-  
chodnych rozwiązania w węzłach leżących blisko krzywoliniowego brzegu. Niech  $\Omega$  zostanie podzie-  
lony na trójkąty, z których niektóre mają krzywoliniowe boki leżące na  $\partial\Omega$ . Na trójkątach okre-  
ślimy wielomiany  $\varphi_j$  jak w punkcie "Elementy trójkątne".



Konstrukcja wielomianu na zwykłym trójkącie  $e_1$  sprowadza się do zbudowania odpowiedniego wielo-  
mianu  $r(\xi, \eta) = \alpha_1 + \alpha_2 \xi + \alpha_3 \eta + \alpha_4 \xi^2 + \alpha_5 \xi \eta + \alpha_6 \eta^2$  na trójkącie  $\bar{e}_1$  o wierzchołkach  $(0,0)$ ,  
 $(1,0)$ ,  $(0,1)$  w płaszczyźnie  $(\xi, \eta)$ . Szukany wielomian  $\varphi(x_1, x_2)$  otrzymujemy wówczas jako

$$\varphi(x_1, x_2) = r[\xi(x_1, x_2), \eta(x_1, x_2)] \quad (x_1, x_2) \in e_1$$

gdzie

$$\begin{aligned} x_1 &= x_{10}(\xi, \eta) = x_1^1 + (x_1^2 - x_1^1)\xi + (x_1^3 - x_1^1)\eta \\ x_2 &= x_{20}(\xi, \eta) = x_2^1 + (x_2^2 - x_2^1)\xi + (x_2^3 - x_2^1)\eta \end{aligned}$$

Dla trójkąta krzywoliniowego  $e$  powyższe odwzorowanie liniowe trzeba zastąpić odwzorowaniem nie-  
liniowym

$$\begin{aligned} x_1 &= x_{10}(\xi, \eta) + (1 - \xi - \eta) \phi(\eta) \\ x_2 &= x_{20}(\xi, \eta) + (1 - \xi - \eta) \psi(\eta) \end{aligned} \quad (7)$$

gdzie funkcje  $\phi(\eta)$ ,  $\psi(\eta)$  konstruujemy znając parametryczne równania  $x_1 = \phi(s)$ ,  $x_2 = \psi(s)$   
brzegu  $\partial\Omega$ . Zastępując  $\phi$  i  $\psi$  ich wielomianami interpolacyjnymi odpowiedniego stopnia i znaj-

dując odwzorowanie odwrotne  $\xi = \xi(x_1, x_2)$ ,  $\eta = \eta(x_1, x_2)$  definiujemy funkcje  $\varphi$  na trójkątach krzywoliniowych. Oczywiście  $\varphi(x_1, x_2)$  na ogół jest funkcją nieliniową. Funkcja  $\varphi$  na wspólnych bokach trójkątów krzywoliniowych jest dla dostatecznie małego  $h$  ciągła.

### 5. Zbieżność i oszacowanie błędu metody

W rozważanym przez nas wariacyjnym sformułowaniu (2) modelowego zadania różniczkowego (1) poszukiwaliśmy rozwiązania dokładnego jako elementu  $W_2^1(\Omega)$  realizującego

$$\min_{v \in W_2^1(\Omega)} I(v)$$

Po zastosowaniu aproksymacji elementem skończonym tzn. budowy skończonej wymiarowej przestrzeni  $S^n \subset W_2^1(\Omega)$  poszukiwaliśmy rozwiązania przybliżonego  $u_n$  jako elementu  $S^n$  realizującego

$$\min_{v \in S^n} I(v)$$

Przez zbieżność metody elementu skończonego rozumiemy, że

$$\lim_{n \rightarrow \infty} \|u_n - u\|_{W_2^1(\Omega)} = 0 \quad (8)$$

Podstawowe twierdzenie określające szybkość zbieżności metody elementu skończonego np. w wypadku rozpatrywanych elementów trójkątnych i prostokątnych gwarantują, że

$$\|u - u_n\|_{W_2^s(\Omega)} = O(h^{k-s} + h^{2(k-1)}) \quad (9)$$

gdzie  $k-1$  oznacza stopień wielomianu, którym jest funkcja  $u_n \in S^n$  na poszczególnych elementach, a  $s$  jest liczbą całkowitą nieujemną. W większości rozważanych zadań wykładnik  $k-s$  jest mniejszy i określa tzw. rząd zbieżności metody. Jeżeli w rozważanej przez nas przestrzeni  $W_2^1(\Omega)$  wprowadzimy tzw. normę energetyczną  $\|\cdot\|_E$  określoną dla  $v \in W_2^1(\Omega)$  wzorem

$$\|v\|_E^2 = a(v, v) \equiv \int_{\Omega} \left( \sum_{i,j=1}^2 a_{ij} \frac{\partial v(x)}{\partial x_i} \frac{\partial v(x)}{\partial x_j} + cv^2(x) \right) dx$$

Wówczas można udowodnić, że w wypadkach rozważanych przez nas w punktach "Elementy trójkątne" i "Elementy prostokątne" prawdziwe są nierówności następującego typu, zwane oszacowaniem błędu metody elementu skończonego w normie energetycznej

$$a(u - u_n, u - u_n) \leq C^2 h^{2(k-1)} \|u\|_{W_2^k(\Omega)}^2 \quad (10)$$

Dowody odpowiednich twierdzeń o zbieżności i wyczerpujące informacje o postaci oszacowań błędów metody elementu skończonego zawierają pozycje bibliografii od [8] do [22].

### 6. Rozwiązanie wynikowego układu algebraicznego równań liniowych

#### Macierz sztywności

Po skonstruowaniu skończonej wymiarowej podprzestrzeni  $S^n$  przestrzeni  $W_2^1(\Omega)$  znalezienie przybliżonego rozwiązania zagadnienia (2) sprowadza się do znalezienia

$$\min_{v \in S^n} I(v) = \min_{v \in S^n} \int_{\Omega} \left( \sum_{i,j=1}^2 a_{ij}(x) \frac{\partial v(x)}{\partial x_i} \frac{\partial v(x)}{\partial x_j} + cv^2(x) - 2f(x)v(x) \right) dx \quad (11)$$



Załóżmy, że funkcja  $u_n \in S^n$  realizująca minimum (11) ma przedstawienie

$$u_n = \sum_{i=1}^n q_i \varphi_i \tag{12}$$

Podstawiając (12) do (11) otrzymujemy formę kwadratową we współczynnikach  $q_i$

$$I(u_n) = q^T Kq + q^T Mq - 2F^T q \tag{13}$$

gdzie  $q$  - wektor szukany o współrzędnych  $q_1, \dots, q_n$ .

$K$  - tzw. macierz sztywności o elementach  $K_{lk} = \int_{\Omega} \sum_{j=1}^2 a_{1j} \frac{\partial \varphi_l}{\partial x_1} \frac{\partial \varphi_k}{\partial x_j} dx$

$M$  - tzw. macierz masową o elementach  $M_{lk} = \rho \int_{\Omega} \varphi_l \varphi_k dx$

$F$  - tzw. wektor obciążeń o elementach  $F_k = \int_{\Omega} f \varphi_k dx$

Wektor  $q$  minimalizujący (13) jest więc określony układem równań liniowych

$$(K + M)q = F \tag{14}$$

Praktycznie macierze  $K, M$  oraz wektor  $F$  budujemy składając je odpowiednio z macierzy sztywności, macierzy masowych i wektorów obciążeń poszczególnych elementów, tzn. obliczając najpierw

$$K_{lk}^e = \int_{\Omega_e} \left( \sum_{j=1}^2 a_{1j} \frac{\partial \varphi_l}{\partial x_1} \frac{\partial \varphi_k}{\partial x_j} \right) dx$$

gdzie  $l$  oraz  $k$  oznaczają tylko te numery funkcji bazowych, dla których  $\varphi_j$  są różne od zera na elemencie  $e$ . Następnie analogicznie

$$M_{lk}^e = \rho \int_{\Omega_e} \varphi_l \varphi_k dx$$

oraz

$$F_k^e = \int_{\Omega_e} f \varphi_k dx$$

W efekcie  $(K + M)q = F$  otrzymujemy jako

$$\sum_e (K^e + M^e)q^e = \sum_e F^e \tag{15}$$

gdzie  $q^e$  jest wektorem zawierającym parametry  $q_j$  o numerach odpowiadających numerom węzłów elementu  $e$ , a sumuje się po wszystkich elementach, na które podzielony jest obszar  $\Omega$ . Macierz  $K + M$  przy założeniach o eliptyczności operatora różniczkowego  $A$  w zadaniu (1) jest dodatnio określona i symetryczna, a więc odwracalna, czyli (14) ma jednoznaczne rozwiązanie. Fakt, że bazowe funkcje  $\varphi_j$  przestrzeni  $S^n$  mają nośniki równe sumie elementów, do których należy węzeł  $z_j$ , a w pozostałe części obszaru  $\Omega$  znikają sprawia, że macierz  $K + M$  jest rozrzedzoną macierzą pasmową. Tak szczególna postać macierzy układu [23], [24], [25] pozwala na efektywne zastosowanie rozmaitych technik algebry liniowej i szczególne ułatwienia w rachunku automatycznym.

Metody bezpośrednie rozwiązania układu równań metody elementu skończonego

W chwili obecnej metody bezpośrednie odgrywają podstawową rolę szczególnie, gdy obszar  $\Omega$  ma odpowiednio prosty kształt (np. jest sumą prostokątów). Najprostszym sposobem jest zastosowanie eliminacji Gaussa, która dla naszej macierzy  $K + M$  jest procesem numerycznie poprawnym. Polega ona na przedstawieniu  $K + M$  jako iloczynu

$$K + M = LU \tag{16}$$

gdzie  $L$  i  $K$  oznaczają odpowiednio macierze trójkątne dolną i górną. Rozwiązanie

$$q = (K + M)^{-1} F$$

otrzymujemy rozwiązując dwa układy o macierzach trójkątnych  $L$  i  $U$ . Ostatecznie

$$q = U^{-1} L^{-1} F$$

W wypadku prostej geometrii obszaru  $\Omega$  stosuje się również wiele innych metod bezpośrednich wykorzystujących tzw. szybkie transformacje Fouriera.

#### Metody iteracyjno rozwiązywania układu równań metody elementu skończonego

Metody tego typu konstruuja ciąg wektorów  $q^j$  takich, że  $\lim q^j = q$  w pewnej normie. Klasycznymi przykładami są odpowiednio zaadaptowane metody kolejnych nadrelaksacji, metoda Gaussa-Seidla, których zbieżność jest na ogół zagwarantowana przez dodatnią określoność macierzy  $K + M$ . Tym niemniej zastosowania metod iteracyjnych spotykamy głównie nie w rozpatrywanym przez nas modelowym zadaniu płaskim (1), lecz gdy  $\Omega$  jest obszarem o skomplikowanym kształcie leżącym w  $R^3$ .

#### Literatura

- [1] Adams R.A.: Sobolev spaces. Englewood Cliffs, 1975 Prentice Hall
- [2] Aubin J.P.: Approximations of elliptic boundary value problems. Nowy Jork: John Wiley 1972
- [3] Lions J.L., Magenes E.: Problèmes aux limites non homogènes et applications. Paryż: Dunod 1968
- [4] Michlin S.G.: Variacionnyye metody matematičeskoj fizyki. Moskwa: Izd. Nauka 1957
- [5] Babuška I., Aziz A.K.: The mathematical foundations of the finite element method with applications to partial differential equations. Nowy Jork: Academic Press 1972, s. 5-359
- [6] Marčuk I.G.: Metody vyčislitelnoj matematiki. Moskwa: Izd. Nauka 1973
- [7] Varga R.S.: Functional analysis and approximation theory in numerical analysis. Philadelphia: SIAM 1971
- [8] Strang G., Fix G.: An analysis of the finite element method. Englewood Cliffs: Prentice Hall 1973
- [9] Oden J.T., Reddy J.N.: An instruction to the mathematical theory of finite elements. Nowy Jork: Wiley-Interscience 1976
- [10] Schultz M.: Spline analysis. Englewood Cliffs: Prentice Hall 1973
- [11] Whiteman J.R.: The mathematics of finite elements and applications. Londyn: Academic Press 1973
- [12] Strang G.: Approximation in the finite element method. Numer. Math. 1972 vol 19 s.81-98
- [13] Babuška I.: Error bounds for finite element method. Numer. Math. 1971 vol 16 s.322-333
- [14] Babuška I.: Finite element method for domains with corners computing. 1970 vol 6 s.264-273
- [15] Ciarlet P.G., Raviart P.A.: Interpolation theory over curved element with applications to finite element methods. Computer Methods in Appl. Math. and Eng. 1972 vol.1 s.217-249
- [16] Clough R.W. and Felippa C.A.: A refined quadrilateral element for analysis of plates in bending. Ohio: 1968 Wright-Patterson
- [17] Ciarlet P.G., Wagschal C.: Multipoint Taylor formulas and applications to the finite element method. Numer. Math. 1971 vol 17 s.84-100
- [18] Bramble J.H., Zlamal M.: Triangular elements in the finite element method. Maths. of Comp. 1970 vol 24 s.112

- [19] Zlamal M.: Curved elements in the finite element method część I i II. SIAM J. Numer. Anal. 1973, vol.10 nr 1; 1974, vol.11 nr 2
- [20] Żenisek A.: Interpolation polynomials on the triangle. Numer. Math. 1970 vol.15 s.283-296
- [21] Ergatoudis I. Ivons B., Zienkiewicz O.C.: Curved isoparametric quadrilateral elements for finite element analysis. Int. J. Solids. Struct. 1968 vol.4 s.31-42
- [22] Fix G., Nassif N.: On finite element approximations to time dependent problems. Numer. Meth. 1972 vol.19 s.127-135
- [23] Irons B.M.: Economical computer techniques for numerically integrated finite elements. Int. J. for Num. Meth. in Eng 1969 vol.1 s.201-203.
- [24] Irons B.M.: A frontal solution program for finite element analysis. Int. J. for Num. Meth. in Eng. 1970 vol.2 s.5-32
- [25] Varga R.S.: Matrix iterative analysis. Englewood Cliffs: Prentice Hall 1962



mgr Adam LUTOBORSKI  
dr inż. Ryszard PAWLIK

Instytut Maszyn Matematycznych

## Systemy metody elementu skończonego - analiza systemów istniejących i koncepcja systemu wzorcowego

### 1. Wstęp

W procesie projektowania konstrukcji bardzo ważnym elementem jest analiza wytrzymałościowa. Klasyczne metody mechaniki i teorii wytrzymałości praktycznie nie wystarczają już w analizie konstrukcji o skomplikowanym kształcie, poddanych zmiennym obciążeniom.

Brak efektywnych metod wykonywania dokładnych obliczeń wytrzymałościowych skomplikowanych konstrukcji z jednej strony, a rosnąco potrzeby na coraz lżejsze konstrukcje, a więc o dokładnie określonych parametrach wytrzymałościowych, spowodował skoncentrowanie wysiłków inżynierów-konstruktorów i matematyków-numeryków nad opracowaniem nowych metod analizy konstrukcji. Równocześnie szybki rozwój sprzętu cyfrowego, a szczególnie wzrost szybkości działania maszyn cyfrowych oraz powiększanie się ich pamięci, dały konstruktorom i numerykom odpowiednie narzędzie do tych poszukiwań. W rezultacie w latach sześćdziesiątych powstała metoda elementu skończonego. Zasadnicze cechy metody scharakteryzujemy w punkcie 2 niniejszego opracowania, a dla dokładniejszego zaznaczenia się z matematycznymi podstawami metody elementu skończonego odsyłamy czytelnika do literatury [11].

Na całym świecie powstało i powstaje coraz więcej systemów obliczeniowych służących do analizy konstrukcji opartych na metodzie elementu skończonego. Jak wynika z samej istoty metody, wymagają one użycia dużych, a więc kosztownych maszyn cyfrowych. Ponadto wykonanie obliczeń metodą elementu skończonego wymaga przygotowania bardzo dużej liczby danych, przy czym w istniejących systemach programowych (obliczeniowych) dane te muszą być przygotowane w ściśle określony sposób. Np. ilość danych potrzebnych do opisanego kształtu i obciążeń działających na konstrukcję lotniczą, karoserię samochodową czy zapórę wodną, wynosi od kilku do kilkunastu tysięcy liczb. Systemy te na wyjściu generują również bardzo dużą liczbę wyników, których przegląd lub ewentualna selekcja wymaga sporego nakładu pracy.

Trudności te powodują, że poznanie sposobów efektywnego wykorzystywania tych systemów jest bardzo pracochłonne i wymaga często nawet półrocznego okresu nauki. Oczywiście jest również, że dla efektywnego korzystania z metody elementu skończonego potrzebna jest duża wiedza inżynierska.

Dlatego też, coraz częściej podejmuje się próby ułatwienia eksploatacji systemów MES właśnie usprawnieniem pierwszej i ostatniej fazy korzystania z systemu. Jak wynika z analizowanych przez autorów materiałów zrealizowane próby tego typu poczynają prowadzą do znacznego zwiększenia efektywności wykorzystywania istniejących i nowopowstałych systemów.

Cheąc sensownie podejść do problematyki usprawniania sposobów wykorzystywania systemów MES należy przede wszystkim zdać sobie sprawę z istoty zadań, roli i struktury istniejących systemów tego typu.

Analizę tę można rozpocząć od prostego (w stosunku do systemu obliczeń inżynierskich) stwierdzenia, że proces analizy konstrukcji oparty na systemach obliczeniowych metody elementu skończonego składa się z trzech podstawowych etapów:

- etap I - przygotowanie danych,
- etap II - właściwe obliczenia systemowe,
- etap III - analiza wyników.

W przedstawionym opracowaniu przeanalizowaliśmy wszystkie trzy etapy procesu obliczeniowego z punktu widzenia zadań, które ono rozwiązują. Dało to podstawę do sformułowania listy zadań, któ-

re powinny być rozwiązane na poszczególnych etapach tego procesu. Zwróciliśmy szczególną uwagę na ułatwienie pracy użytkownika z systemem, a więc etapy I i III.

Analiza ta doprowadziła do powstania koncepcji wzorcowego systemu metody elementu skończonego o mocno rozbudowanych możliwościach w zakresie etapu I i III, które decydują o łatwości korzystania z systemu.

W dalszej części artykułu podaliśmy warunki, które muszą być spełnione, aby system taki można eksploatować z pełnym wykorzystaniem jego możliwości. W zakończeniu przedstawiliśmy krótką charakterystykę systemów wykorzystywanych w Polsce.

Ponieważ problemy poruszane w niniejszym artykule dotyczą zarówno różnych dziedzin inżynierii od budownictwa po mechanikę, jak i pewnych aspektów matematycznych, a równocześnie rozpatrują zagadnienia informatyczne, uznaliśmy za wskazane ujednolicić na potrzeby tego artykułu to pojęcia i terminy, które mogą być różnie rozumiane przez różnych czytelników.

- System obliczeniowy MES - program lub pakiet programów, korzystający z odpowiednio zorganizowanych zbiorów danych umieszczonych w pamięci zewnętrznej, wymagający do 0.5 Mb pamięci operacyjnej.
- Skończenie elementowy model konstrukcji - model konstrukcji rzeczywistej zestawiony z elementów skończonych wchodzących w skład biblioteki elementów (np. prętów, trójkątnych elementów powłokowych, prostopadłościemych elementów bryłowych itd.), nazywany również niekiedy siatką.
- Biblioteka elementów - zbiór programów (na ogół fortranowskich) definiujących kształt elementów skończonych i rodzaj przyjętej w nich interpolacji przemieszczeń i naprężeń oraz obliczających sztywności poszczególnych rodzajów elementów.
- Preprocesor systemu obliczeniowego MES - samodzielny program, którego wyniki używa się jako danych w systemie obliczeniowym MES.
- Postprocesor systemu obliczeniowego MES - samodzielny program, w którym jako danych używa się wyników obliczeń systemu obliczeniowego MES. Pre- i postprocesory mają za zadanie ułatwić komunikację użytkownika z systemem MES.
- Diagnostyka błędów - obejmuje klasyfikację i lokalizację błędów.
- Wzorcowy system obliczeniowy MES - system MES uwzględniający w maksymalnym stopniu potrzeby użytkownika. System taki nie istnieje. Istniejące systemy tylko w części realizują postulowane w opracowaniu funkcje systemu wzorcowego.

## 2. Analiza konstrukcji metodą elementu skończonego i pierwsze programy obliczeniowe

Metodę podziału konstrukcji na elementy oraz badanie jej statycznych, a następnie dynamicznych własności jako dyskretnego układu tych elementów sformułowano w końcu lat pięćdziesiątych. Głównymi ośrodkami rozwoju tej teorii stały się Techniczny Uniwersytet w Stuttgarcie (RFN), a w nim zespół prof. J.H. Argyrisa, Uniwersytet Walijski w Swansea (Wielka Brytania), w którym pracuje zespół prof. O.C. Zienkiewicza oraz Uniwersytet w Berkeley (Stany Zjednoczone) z zespołem prof. R.W. Clough'a [1], [2], [3]. Tam też, niezależnie od siebie, powstały pierwsze programy realizujące statyczną analizę konstrukcji, a następnie analizę dynamiczną w ramach liniowej teorii sprężystości. W zaproponowanych metodach skończenie elementowe aproksymacje zadań wariacyjnych mechaniki [1], doprowadziły do układu równań liniowych w postaci

$$M\ddot{q} + C\dot{q} + Kq = F \quad (1)$$

gdzie  $M$  oznacza tak zwaną macierz masową konstrukcji,  $C$  - macierz tłumienia,  $K$  - macierz sztywności. Wektory  $q$ ,  $\dot{q}$ ,  $\ddot{q}$  i  $F$  oznaczają odpowiednio uogólnione wektory przemieszczeń, prędkości, przyspieszeń węzłowych oraz wektor uogólnionych obciążeń. W wypadku statycznym układ (1) przyjmuje prostszą postać:

$$Kq = F \quad (2)$$

Badania teoretyczne [1] dotyczące metody elementu skończonego pozwoliły na ustalenie odpowiednich warunków gwarantujących istnienie rozwiązania układów (1) i (2), i co więcej, sformułowano [2] algorytmiczne podstawy budowy macierzy M, C i K oraz numerycznego rozwiązania układów analitycznej i dynamicznej. Powyższe sformułowanie problemu pozwoliło na wyodrębnienie szeregu wspólnych podprogramów numerycznych stanowiących integralną część wszystkich opracowanych już wówczas programów obliczeniowych metody elementu skończonego; są to programy przygotowujące dane, programy numeryczne rozwiązujące (1) i (2), biblioteka elementów i programy interpretujące wyniki. Wielkość programów obliczeniowych MES i możliwość wykorzystania ich do rozwiązania szerokiej klasy zadań spowodowały, że nazywa się je również systemami obliczeniowymi metody elementu skończonego.

Koncentracja wysiłków badawczych w wymienionych już i nowopowstałych ośrodkach doprowadziła w efekcie w latach 1964 - 1966 do powstania pierwszych w polni zasługujących na tę nazwę systemów: SAMIS, FORMAT, a następnie w roku 1966 systemów NASTRAN i ASKA, które zdobyły szeroki krąg użytkowników ze względu na swą wszechstronność. Dwa ostatnie są doskonałe i rozwijane nadal [5],[7].

W części 3 niniejszego opracowania scharakteryzowaliśmy poszczególne fragmenty wzorcowego systemu metody elementu skończonego i uzupełniliśmy je przykładami istniejących już rozwiązań w odpowiednich systemach.

Niektóre spośród istniejących systemów MES tylko w szczegółach różnią się od przedstawionego w niniejszym opracowaniu abstrakcyjnego systemu wzorcowego, jednak przeważająca większość systemów, ze względu na pełną orientację problemową lub ograniczenia natury sprzętowej przy ich uruchamianiu, tylko we fragmentach odpowiada systemowi wzorcowemu.

Tak więc w dalszym ciągu systemem wzorcowym nazywać będziemy system odpowiadający charakterystyce przedstawionej w punkcie 3.

### 3. Charakterystyka wzorcowego systemu obliczeniowego metody elementu skończonego

#### Przygotowanie danych - preprocesory □

Do danych w metodzie elementu skończonego zaliczamy:

- współrzędne węzłów siatki podziału konstrukcji na elementy,
- informacje o typie, liczbie i sposobie łączenia elementów,
- informacje o wyróżnionych grupach elementów - superelementach,
- informacje o warunkach brzegowych, tzn. o sposobie zamocowania konstrukcji, jej temperaturze, początkowych przemieszczeniach lub naprężeniach,
- charakterystyki materiałowe,
- stałe fizyczne.

Liczby danych dla dużych zadań sięgają setek tysięcy, w związku z tym prawidłowe ich przygotowanie stanowi dużą trudność i wymaga wielokrotnych poprawek. Dużą pomoc na etapie przygotowania danych, a więc przy budowie skończonego elementowego modelu obliczanej konstrukcji oddają programy zwane preprocesorami. Preprocesory dostarczają także pewnych informacji o przewidywanym czasie obliczeń zasadniczych oraz innych własnościach modelu.

Do podstawowych preprocesorów należą obecnie preprocesor graficzny oraz generator siatki.

Preprocesor graficzny pozwala użytkownikowi naocznie sprawdzić, czy wygenerowany model konstrukcji w pełni odpowiada rzeczywistości obliczanej konstrukcji. Typowe istniejące preprocesory graficzne umożliwiają między innymi:

- rysowanie całej siatki lub jej fragmentów w różnych skalach,
- schematyczne przedstawienie siatki po obrocie w przestrzeni,
- rysowanie obliczanej konstrukcji lub jej fragmentów w perspektywie.

Pozwala to szybko zauważyć brak węzłów lub elementy zdegenerowane, powstałe w wyniku błędnego podania współrzędnych węzłów.

Drugim podstawowym preprocesorem jest generator siatki. Automatyczna generacja siatki oznacza przede wszystkim automatyczny podział konstrukcji na elementy, np. w przypadku płaskim automatyczną triangulację lub podział na inne wielokąty wypukłe. Możliwe jest także zagęszczenie podziału w wyróżnionych punktach badanego obszaru. Generator siatki uwalnia użytkownika od znużającego podawania współrzędnych wszystkich węzłów, wystarczy jedynie odpowiedni opis zarysu konstrukcji, typów elementów oraz żądanej gęstości siatki. Problem stworzenia generatora siatki o bardziej uniwersalnym przeznaczeniu jest złożony teoretycznie i trudny w realizacji i dlatego nie wszystkie systemy już obecnie nim dysponują.

Ponadto preprocesory powinny realizować następujące funkcje mające na celu sprawdzenie poprawności zbudowanego modelu:

- podawanie wykazu liczby i rodzaju użytych elementów,
- podawanie rodzaju użytych obciążeń i punktów ich przyłożenia,
- podawanie warunków brzegowych,
- podawanie maksymalnych i minimalnych odległości, między węzłami w całej konstrukcji lub jej częściach,
- podawanie odległości między wskazanymi węzłami siatki.

Uzyskanie powyższych informacji pozwala na lepsze sprawdzenie modelu rozwiązywanej konstrukcji, a także umożliwia diagnostykę błędów. Bardzo przydatną funkcją, którą może realizować preprocesor jest szacowanie czasu obliczeń poszczególnych etapów obliczeń zasadniczych. Informacja ta pozwala na racjonalne planowanie przebiegu dalszych obliczeń.

Bardzo dużym ułatwieniem procesu przygotowywania danych może być praca w trybie konwersacyjnym. Praca taka pozwala znacznie skrócić czas przygotowywania danych przez możliwość szybkiego poprawiania popełnionych błędów.

Biblioteka elementów, programy numeryczne i program sterujący □

Jakość rozwiązania zadania statycznej lub dynamicznej analizy konstrukcji, zależy przede wszystkim od sposobu aproksymacji kształtu konstrukcji za pomocą jej skończenie elementowego modelu i od aproksymacji rzeczywistych przemieszczeń, odkształceń i naprężeń lub innych poszukiwanych wielkości fizycznych będących funkcjami określonymi na przyjętym modelu. Zbiór programów realizujących te funkcje nazywany jest w literaturze biblioteką elementów. Zawartość biblioteki elementów zależy od zakresu zastosowań systemu. System wzorcowy powinien gwarantować łatwy dostęp do poszczególnych programów w bibliotece i dawać możliwość ich natychmiastowego wykorzystania przy pisaniu odrębnych programów problemowo zorientowanych. We wzorcowym systemie powinna być również zapewniona możliwość rozszerzenia biblioteki o nowe elementy. Na przykład, biblioteka systemu ASKA zawiera około 50 różnych elementów, m.in. najprostsze elementy prętowe, płaskie, trójkątne i czworokątne elementy membranowe, krzywoliniowe, osiowoosymetryczne elementy pierścieniowe oraz szereg elementów trójwymiarowych czworościennych i utworzonych z nich makroelementów. Co więcej, współczesne biblioteki systemów MARC, SAP IV zawierają elementy stanowiące model pewnych typowych fragmentów konstrukcji. Są nimi np. elementy belkowe, płytowe, rurowe itp. Nowoczesną bibliotekę elementów uzupełniają jeszcze elementy specjalne o własnościach aproksymacyjnych, pozwalających na modelowanie luzów i pęknięć w konstrukcjach.

Wybór odpowiednich programów z biblioteki elementów umożliwia budowę skończenie elementowego modelu konstrukcji, a następnie pozwala na automatyczną generację macierzy masowej  $M$ , macierzy tłumienia  $C$  i macierzy sztywności  $K$  oraz wektora obciążeń  $F$  w rozwiązywanym układzie (1) lub (2). Otrzymany tą drogą liniowy układ równań (1) zawiera na ogół kilkanaście tysięcy równań. Jest to jednak układ o rozrzedzonej, pasmowej macierzy.

Twórcy systemów zrykają organizują obliczenia w ten sposób, że elementy tej macierzy znajdują się w pamięci zewnętrznej. Algorytmy rozwiązujące ten układ sposobem możliwie ekonomicznym i numerycznie poprawnym stanowią zwykle część zbioru programów numerycznych systemu MES. Sprawność tych programów decyduje zazwyczaj o sprawności centralnej części systemu.



Wyniki analizy konstrukcji i ich interpretacja za pomocą postprocesora

W wyniku przeprowadzonych obliczeń konstrukcji metodą elementu skończonego otrzymuje się dla każdego węzła sieci siły działające na ten punkt, przemieszczenie tego punktu wywołane tymi siłami oraz występujące w nim naprężenie. Wobec wspomnianej ogromnej liczby węzłów, oczywiste jest, że interpretacja tak dużej liczby wyników jest zadaniem niezmiernie trudnym. Analiza oraz ewentualna selekcja wyników trwa więc wielokrotnie dłużej niż właściwe obliczenia. Niezbędna jest więc jakaś pomoc dla użytkownika systemu. W systemie wzorcowym rolę takiej pomocy pełnią postprocesory. Ich głównymi zadaniami są:

- graficzne przedstawienie wyników za pomocą plottera,
- szukanie ekstremalnych wartości obliczonych wielkości (naprężeń, przemieszczeń),
- obliczanie pewnych funkcji na wielkościach obliczonych, na podstawie których można oceniać konstrukcję (kryteria równowagi).

Obecnie prawie wszystkie systemy wyposażone są w postprocesory, które wykonują następujące rysunki:

- odkształcona sieć (model odkształceń konstrukcji),
- izolacje naprężeń,
- kierunki sił działających na poszczególne elementy.

Na przykład system SESAM-69 ma postprocesor graficzny GETPLOT sterujący plotterem firmy Cal-Comp lub Kingmatic [8]. Za pomocą tego postprocesora można uzyskać wykresy wielkości sił tnących i momentów zginających oraz wielkości przemieszczeń i naprężeń.

#### 4. Warunki efektywnego stosowania systemu mes

Warunki ograniczające □

Przyjmując, zgodnie z tym co było powiedziane, że wzorcowy system MES powinien być przeznaczony do eksploatacji na dużych maszynach, trzeba równocześnie konsekwentnie dążyć do tego, żeby w ręku sprawnego inżyniera-informatyka stał się narzędziem efektywnym. Obecnie, na ogół ze względu na swą specyfikę i skomplikowaną budowę, sprawne i pełne wykorzystanie takich systemów wymaga wielu miesięcy szkolenia i praktycznych doświadczeń. Powoduje to duże trudności w szybkim wprowadzaniu tej techniki obliczeń wytrzymałościowych do praktyki w biurach konstrukcyjnych i projektowych. Perspektywa długiego szkolenia i zmuśnej kilkumiesięcznej pracy zniechęca potencjalnych użytkowników systemów mes. Nawet inżynierowie początkowo zafascynowani perspektywami tej metody, po kilku tygodniach rezygnują ze stosowania tych metod i wracają do tradycyjnego projektowania. Pozostają jedynie ci, którzy nie mogą swoich zadań policzyć innymi metodami lub ci, którzy na swej drodze do informatyki napotkali życzliwość i dobrą organizację.

Naszym zdaniem, głównym środkiem do poprawy sytuacji i pokonania tych trudności są zmiany w organizacji eksploatacji tego typu systemów. Obecnie przy wykorzystywaniu systemów mes stosowane są dwie metody. Pierwsza z nich polega na tym, że użytkownik sam przygotowuje dane dla systemu i sam usuwa błędy. Jest to metoda mało efektywna, absorbująca cenny czas użytkownika-konstruktora. Drugą metodą jest zlecenie obliczeń wyspecjalizowanemu ośrodkom; specjaliści z tych ośrodków wykonują wszelkie prace związane z policzeniem otrzymanego zadania. Metoda ta pozornie wygodna dla użytkownika, ma jednak podstawową wadę, a mianowicie, całkowite odsunięcie użytkownika od procesu tworzenia skończonego elementowego modelu obliczanej konstrukcji. A właśnie budowa tego modelu ma decydujący wpływ na jakość otrzymanych wyników. Należy więc dążyć do włączenia konstruktora do procesu budowy modelu obliczanej konstrukcji. Będzie to możliwe tylko wtedy, gdy proces ten będzie przebiegał szybko i sprawnie. Jedynym sposobem osiągnięcia tego celu jest konwersacyjna budowa takiego modelu. Może być ona realizowana poza centralnymi częściami systemu mes, za pomocą odpowiedniego preprocesora, np. na minikomputerze. Przy sprawnej i szybkiej realizacji procesu budowy modelu użytkownik-konstruktor będzie mógł całkowicie sam tworzyć model obliczanej konstrukcji. Byłoby to rozwiązaniem najlepszym. Prace nad powstaniem takiego procesora są bardzo istotne, choć niezmiernie trudne i pracochłonne.

Do czasu opracowania sposobów szybkiej konwersacyjnej budowy modelu obliczanej konstrukcji, za pomocą których użytkownik-konstruktor sam będzie mógł zbudować poprawny model obliczanej konstrukcji należy organizować w ośrodkach obliczeniowych zespoły informatyków specjalizujących się tylko w eksploatacji systemów MES, doskonale znających możliwości i zasady eksploatacji danego systemu. Zespół ten musi składać się z inżynierów lub fizyków specjalizujących się w mechanice, matematyków-numeryków, programistów systemowych i programistów. Należałoby również zapewnić członkom tego zespołu kontakty z innymi ośrodkami pracującymi z podobnymi systemami [14]. Specjaliści ci otrzymaliby do policzenia zadania od inżynierów, uzgadniając z nimi precyzyjnie dane zadania, a po wykonaniu obliczeń przekazywali im wyniki. Niewątpliwie, obecnie organizacja taka jest praktycznie jedynym rozwiązaniem problemu efektywnego wykorzystania systemów metody elementu skończonego i szybkiego jej wdrożenia do praktyki.

Specjalnego zwrócenia uwagi przy organizacji eksploatacji systemów MES wymaga etap interpretacji wyników. Otrzymane wyniki należy jak najszybciej dostarczyć użytkownikowi w możliwie najbardziej czytelnej formie. Prace nad organizacją obróbki otrzymywanych wyników są także bardzo ważne i nie można o nich zapominać.

#### Warunki sprzętowe

Systemy MES wykonują obliczenia wymagające stosunkowo dużej pamięci operacyjnej. Z tego powodu EMC musi być wyposażona w pamięć operacyjną około 0,5 Mb oraz szybki arytmometr o odpowiednio zrealizowanej arytmetyce zmiennopozycyjnej, zapewniającej zadaną dokładność obliczeń, gdyż obecnie obliczenia MES prowadzi się na ogół w podwójnej precyzji. Przy takich parametrach maszyny, czas obliczeń - zależnie od analizowanego problemu - waha się od kilkunastu minut do kilkunastu godzin. Ponadto do sprawnej pracy z systemem pożądana są następujące urządzenia peryferyjne:

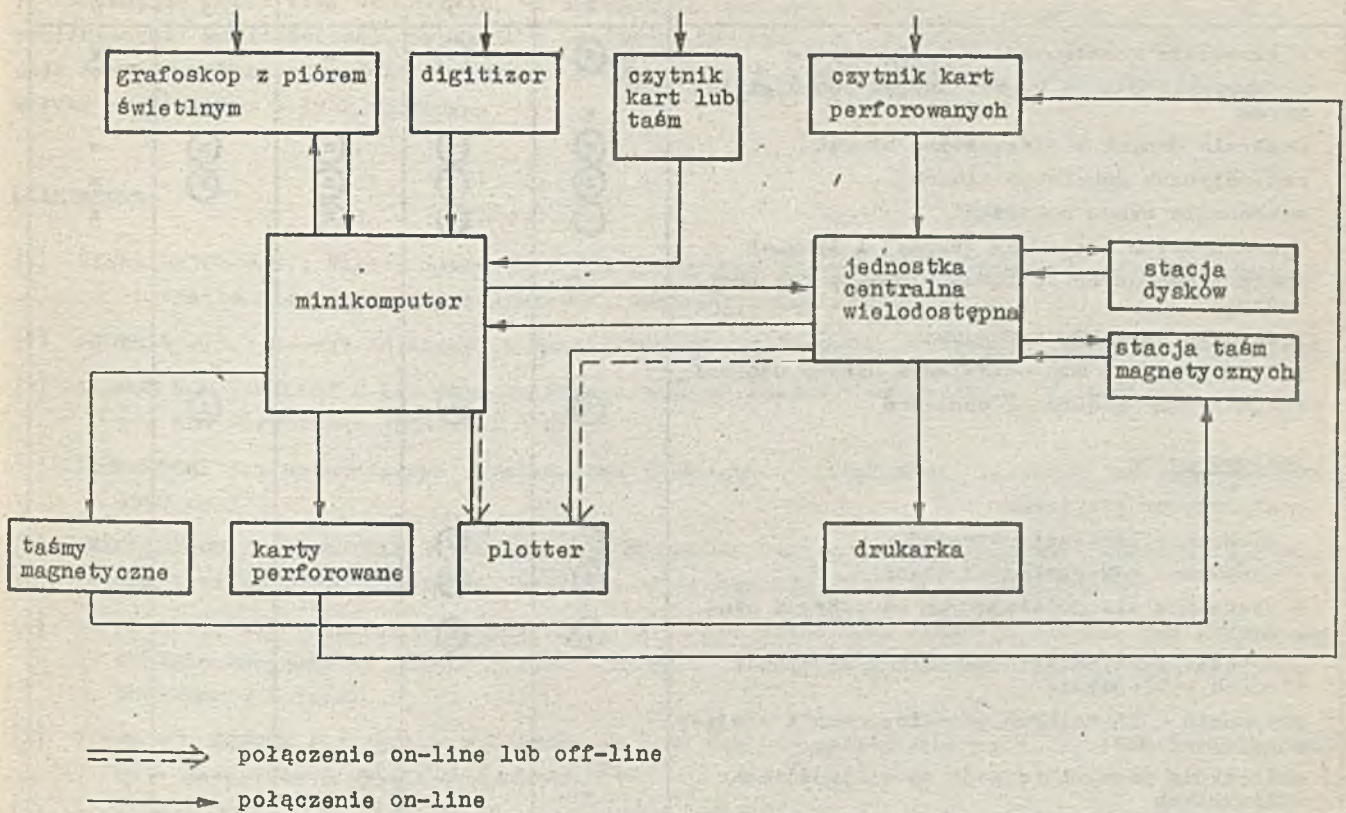
- czytnik kart perforowanych,
- drukarka wierszowa,
- stacje dysków magnetycznych,
- stacje taśm magnetycznych,
- digitizer,
- plotter,
- końcówki alfanumeryczne,
- grafoskop z piórom świetlnym.

Do prawidłowego i efektywnego wykorzystania tak bogatego sprzętu, niezbędno jest dostosowanie do niego oprogramowanie. Do podstawowych środków programistycznych należy zaliczyć systemy wielodostępne i systemy konwersacyjne. Systemy wielodostępne umożliwiają pracę wielu użytkownikom jednocześnie. W czasie, gdy jedni użytkownicy przygotowują dane, inni wykonują już obliczenia zasadnicze, a jeszcze inni interpretują wyniki za pomocą postprocesorów. Systemy konwersacyjne, jak opisano w poprzednim punkcie, są szczególnie przydatne w czasie przygotowywania danych. W tej fazie korzystania z systemów MES szczególnie często występują przerwania w obliczeniach ze względu na błędy w modelu. Warto także do przygotowywania danych wykorzystać minikomputery we współpracy z wyżej wymienionymi urządzeniami peryferyjnymi. Preprocesory opracowane na minikomputery pozwolą stosunkowo tanio zbudować model obliczanej konstrukcji. Schemat funkcjonalny przykładu konfiguracji sprzętowej przedstawia poniższy rysunek (rys. 1).

#### 5. Informacja o działających systemach MES i ich rozwoju

Według informacji szwedzkiego Linköping Institute of Technology [5] istnieje na świecie około 140 działających systemów metody elementu skończonego, wśród których przede wszystkim należy wymienić systemy: ASKA, NASTRAN, ICES-STRUDL, AMSA20, ASAS, DASY, DERSAFE, COSA, FINISS, MARC, SESAM 69, SAP-IV, STARDYNE, TITUS.

Obecnie powstają w Stanach Zjednoczonych systemy ADINA i TOPAS opracowywane przez specjalne firmy. Wiele z wymienionych systemów rozwija się nadal i są one doskonałe w zastosowaniach daleko odbiegających od wzorcowego zadania statyki i dynamiki konstrukcji. Stosuje się w nich najnowsze wyniki teoretyczne, dotyczące uogólnień wzorcowej metody elementu skończonego, czyli tzw. metod mieszanych. Oprócz tego doskonałe są inne programy numeryczne, stanowiące wspólną część wszystkich systemów. Niektóre z systemów dostosowuje się do określonych typów problemów np. techniki nuklearnej lub konstrukcji maszynowych czy budowlanych.



Rys. 1. Schemat funkcjonalny konfiguracji sprzętowej dla wzorcowego systemu mes

W Polsce zainstalowano wersje systemów zachodnich, tj. ASKA i SAP IV z analizą liniowosprężystą, statyczną i dynamiczną. System ASKA w postaci binarnej i SAP IV w postaci źródłowej są użytkowane przez OBR Maszyn Budowlanych "Bumar". ASKA i SAP IV są systemami nastawionymi na szeroki krąg użytkowników [6], [7]. Ośrodek ten dysponuje również prostą wersją systemu ICES STRUDL przeznaczonego dla budownictwa. System SESAM-69 w wersji dostosowanej do badania konstrukcji okrętowych działa w Gdańsku w Zakładzie Informatyki Przemysłu Okrętowego [8].

Pełne możliwości systemów ASKA, SAP IV, SESAM 69 są znacznie większe niż ich wyżej wymienione wersje działające w Polsce.

Należy również wymieniać systemy powstałe w Polsce: system KM Katedry Mechaniki WAT [9], [10] oraz SES [11] Politechniki Gdańskiej, a także program STRAINS [12] powstały w Politechnice Warszawskiej i SHAFT w OBRTIKM Tokoma. Ponadto w wielu środowiskach powstają programy do rozwiązywania poszczególnych konkretnych problemów.

Dokładną charakterystykę wybranych systemów, stosowanych w kraju i uznanych za reprezentatywne, podaje następujące zestawienie:

Charakterystyka systemów obliczeniowych mes stosowanych w kraju

Informacja programowa o systemie	ASKA	SESAM-69	SAP IV	KM	STRUDL II
1	2	3	4	5	6
<u>Preprocesory</u>					
preprocesor graficzny					
- rysowanie całej siatki lub jej fragmentów	⊗	⊗	x		x
- schematyczne przedstawienie obrotów siatki w przestrzeni	⊗	?	x		x

1	2	3	4	5	6
- rysowanie konstrukcji w perspektywie	(x)	(x)	x		x
redagowanie danych z grafoskopem lub digitizerem	x	x			
kontrola danych z diagnostyką błędów	(x)	(x)	(x)	(x)	x
automatyczna generacja siatek	(x)	(x)	(x)	(x)	x
szacowanie czasu obliczeń	(x)	(x)	(x)		x
wykaz użytych elementów (rodzaj i liczba)					
wykaz stosowanych obciążeń i punkty ich przyłożenia					
podawanie warunków brzegowych					
podawanie max i min odległości między węzłami					
automatyczna generacja obciążeń	(x)	(x)		(x)	
<u>Postprocesory</u>					
postprocesor graficzny					
- rysowanie izol linii naprężeń	(x)	(x)	x		x
- rysowanie odkształconej sieci	(x)	(x)	x		x
- rysowanie sił działających na wybrano elementy	(x)	(x)	?		
podawanie maksymalnych wartości naprężeń i miejsce wystąpienia					
podawanie maksymalnych przemieszczeń i miejsce wystąpienia					
obliczanie pewnych funkcji na wielkościach obliczonych					
podawanie max i min odległości między węzłami sieci odkształconej					
<u>Informacje ogólne</u>					
możliwość restartu	(x)	(x)	x		x
swobodny format danych	x			x	x
<u>Implmentacja systemu</u>					
IBM serii 360, 370	(x)	(x)	(x)	x	x
CDC 6400, 6600, 7600	x		x		
UNIVAC	x		(x)		x
interaktywno wykorzystanie sprzętu peryferyjnego					
- plotter	(x)	(x)	x		(x)
- digitizer	x	x			
- grafoskop	x	x			x
- minikomputer	?	?			
minimalna pojemność pamięci operacyjnej w krajowej implementacji	250k	250k	250k	250k	250k
maksymalna liczba zmiennych	30000	kilka tys.	kilka tys.	kilka tys.	kilka tys.

x - wymieniony system oznacza się daną własnością

(x) - zainstalowana w Polsce wersja wymienionego systemu zagranicznego oznacza się daną własnością

? - brak informacji

Z przedstawionego zestawienia wynika, że żaden z eksploatowanych w Polsce systemów nie spełnia równocześnie wszystkich wymagań systemu wzorcowego. Dotyczy to szczególnie możliwości preprocesorów i postprocesorów.

Reasumując można więc stwierdzić, że aczkolwiek realizacja systemu wzorcowego, z wszystkimi wymienionymi możliwościami, wymagałaby wielkich nakładów sił i środków, to jednak należy istniejąco systemy wzbogacać o takie brakujące preprocesory i postprocesory, które najskuteczniej ułatwiłyby korzystanie z tych systemów.

#### Literatura

- [1] ZIENKIEWICZ O.C.: Finite Elements - The Background Story. W: Whiteman J.R.: The Mathematics of Finite Elements and Applications, Academic Press London 1973
- [2] ARGYRIS J.H.: Recent Advances in Matrix Methods of Structural Analysis. Pergamon Press 1964
- [3] CLOUGH R.W., WILSON E.L.: Dynamic Finite Element Analysis of Arbitrary Thin Shells. Computers and Structures 1971 vol. 1 nr 1
- [4] LUTOBORSKI A.: Matematyczne podstawy metody elementu skończonego. Biuletyn Informacyjny OSK 1977 nr 5/6
- [5] FREDRIKSSON B., MACKERLE J.: Structural Mechanics Finite Element Computer Programs. Surveys and Availability. Linköping Institute of Technology 1975, Linköping, Sweden.
- [6] BATHE K.J., WILSON E.L., PETERSON F.E.: SAP IV. A Structural Analysis Program For Static and Dynamic Response of Linear Systems. Berkley 1973 (odbitka kserograficzna - materiały w posiadaniu autora)
- [7] FUCHS G., SCHIREM F.: ASKA - a Computer System for Structural Engineers (odbitka kserograficzna - materiały w posiadaniu autora)
- [8] Biblioteka Programów. SESAM-69 - Instrukcja użytkownika. Gdańsk: Centrum Techniki Okrętowej. 1974.
- [9] SZMELTER J., DACKO M.: Instrukcja programu "Analiza statyczna konstrukcji powłokowo-prętowych", wersja 1. Warszawa: WAT 1976
- [10] Programy metody elementu skończonego. Warszawa: Arkady 1973
- [11] Metoda sztywnych elementów skończonych. Warszawa: Arkady 1975
- [12] BZYMEK Z., KAMIŃSKI R., GRYSZKIEWICZ M.: Podręcznik użytkownika systemu STRAINS 71. Warszawa: Politechnika Warszawska 1974
- [13] RUMIŃSKI K., POLCH E.Z., PAWLAK D.: Problemy efektywnego wdrażania systemów opartych na MES do praktyki konstrukcyjnej. W: Zastosowanie informatyki w przedsiębiorstwach przemysłu maszynowego. Cz. II. Warszawa: SIMP 1977 s. 19-29.
- [14] LUTOBORSKI A.: Systemy metody elementu skończonego. Warszawa: IMM 1977. Archiwum opracowań IMM nr 28.



mgr inż. Jorzy MOCAŁA  
Instytut Maszyn Matematycznych

## Zagadnienia modelu procesu projektowania w komputeryzacji projektowania technicznego

### 1. Wprowadzenie w problematykę

Komputeryzacja procesu projektowania narzuca nie tylko analizę procesu projektowania opartą na metodologii projektowania, tak jak to jest przedstawione w pracy [7], ale także wymusza rozpatrzenie procesu projektowania z punktu widzenia informatyki, tzn. na płaszczyźnie takich terminów jak język [12], struktury danych [20].

Zastosowanie mc w projektowaniu można uznać za opracowanie układu: proces projektowania - maszyna cyfrowa. Jak uczy doświadczenie z prac badawczych nad systemami teorie matematyczne, jak: teoria automatów, teoria masowej obsługi itp. pozwalają [3] rozwiązywać jedynie niektóre problemy systemów. Za pomocą tych teorii można badać systemy zawierające elementy, które w wyniku formalizacji są albo wszystkie automatami lub wszystkie - systemami masowej obsługi itd. Jeśli elementy systemu opisywalne są przez różne teorie (niektóre są automatami, inne systemami masowej obsługi, itp.), to bez posługiwania się bardziej ogólnymi teoriami nie można badać systemu w całości.

Ponieważ obszarem zainteresowań jest całość układu (projektowanie - maszyna cyfrowa) należy uzgodnić opisy elementów układu uzyskane w postępowaniu badawczym. Elementy układu należą do różnych klas i tradycyjnie badane są przez różne dyscypliny naukowe. Projektowaniem zajmuje się metodologia projektowania [5], [8], [9] zbudowana na wzór metodologii nauk [1], maszynami cyfrowymi - informatyka nauka, której obszar zainteresowań przedstawiają prace [11], [13], [15], [16], [18], [22].

Warto zwrócić uwagę, że przetwarzanie informacji nierozłącznie związane jest z językiem (węzłowy problem informatyki). Język jest jedynym narzędziem do formułowania informacji [12]; jej wymiana i opracowanie mogą się odbywać tylko wówczas, gdy dysponujemy językiem, który tworzy abstrakcyjnie, jakimi są informacje, zamienia na "przedmioty fizyczne" będące nośnikami informacji.

Dlatego też wydaje się, że język będzie dobrą podstawą uzgadniania opisów elementów rozpatrywanego układu. Z punktu widzenia badań metodologii projektowania w pracy [6] sygnalizowano potrzebę rozpatrzenia procesu projektowania w postaci procesu przetwarzania informacji mającej formę wyrażon należących do pewnych języków systemu projektującego oraz systemów współdziałających z nim.

Na gruncie informatyki rozróżnia się języki tzw. uniwersalne (wymagania nakładane na te języki przez użytkownika mc przedstawiono np. w [2]) i języki opisu danych [21]. Języki uniwersalne wymagają jednoznacznego zdefiniowania obiektów przez podanie repertuaru operacji na nich wykonywanych i operacji przez podanie obiektów biorących w niej udział. Języki opisu danych (ostatnia tendencja w informatyce) pozwalają opisywać dane niezależnie od ewentualnych procesów przetwarzania danych.

W opracowaniu przedstawiono koncepcję modelu procesu projektowania wzorowaną na idei języka opisu danych. Prezentację koncepcji modelu poprzedzono przeglądem modeli procesu projektowania użytych w pracach zastosowaniowych mc do projektowania.

### 2. Przegląd wybranych modeli procesu projektowania technicznego

Ogólna charakterystyka prezentowanych modeli

W rozdziale przedstawiono trzy modele procesu projektowania układów sterowania: .

- M1 - projektowanie systemów sterowania obrabiarkami,
- M2 - projektowanie układów sterowania procesami technologicznymi,
- M3 - projektowanie złożonych systemów automatyki (bez precyzowania obiektów sterowania).

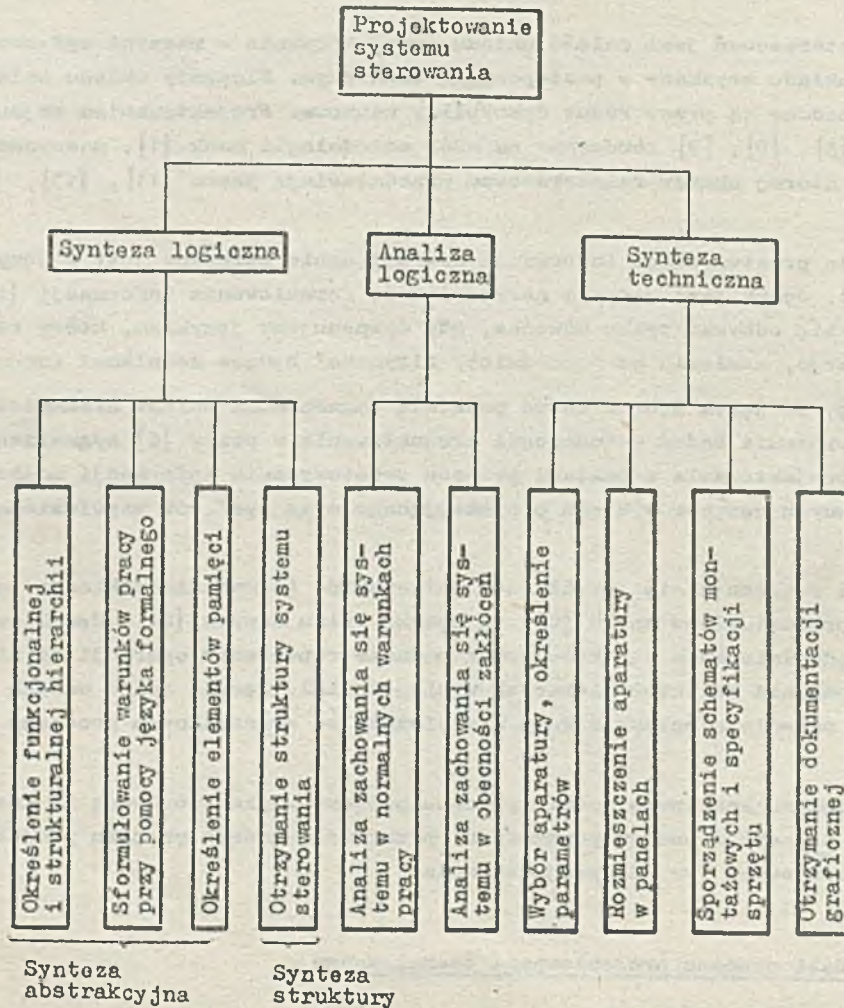
Opisy te zaczerpnięto z prac [4], [17], [19] dotyczących automatyzacji konkretnych procesów projektowania za pomocą elektronicznych maszyn cyfrowych. Na podstawie tych modeli przeprowadzono automatyzację wyżej wymienionych procesów. Celem tego rozdziału jest dokonanie porównań modeli. Starano się więc zachować formę prezentacji i terminologię, najbardziej zbliżoną do formy użytej w cytowanych pracach.

#### Model procesu projektowania systemów sterowania obrabiarkami - M1

Przedmiotem projektowania jest system sterowania, który powinien umożliwiać:

- wybór potrzebnego reżimu pracy obrabiarki przez operatora,
- właściwą kolejność ruchów roboczych odpowiadających technologicznemu procesowi w wybranym reżimie pracy,
- bezpieczną, bezawaryjną i niezawodną pracę obrabiarki,
- zatrzymanie obrabiarki,
- podjęcie przerwanej pracy obrabiarki,
- powrót części roboczych z dowolnego położenia do położenia wyjściowego,
- sygnalizację pracy obrabiarki.

Proces projektowania obiektu spełniającego wyżej wymienione funkcje w postaci schematu blokowego przedstawia rys. 1. Dzieli się on wg pracy [4] na niżej wymienione etapy.



Rys. 1. Schemat blokowy procesu projektowania systemu sterowania obrabiarkami [4]



- Określenie funkcjonalnej i strukturalnej hierarchii systemu sterowania. Funkcje wypełnione przez system dzieli się na główne i wspomagające, a te z kolei na podstawowe i uzupełniające. Funkcje uzupełniające wiąże się z warunkami, które zapewniają realizację funkcji podstawowych. Realizację funkcji systemu zapewniają podsystemy (elementy) systemu sterowania. Między elementami systemu istnieje zależność hierarchiczna wyznaczana na podstawie zawierania się zbiorów sygnałów wyjściowych poszczególnych elementów systemu.
- Sformułowanie warunków pracy systemu za pomocą języka formalnego. Dla każdego podsystemu określa się, przy jakich sygnałach wejściowych pożądane są sygnały na wyjściu. Związki między we/wy opisuje się w języku "tablic stanów".
- Określenie elementów pamięci. Jeśli sygnały wejściowe są jednoznaczne, tzn. nie ma takich samych sygnałów wejściowych, dla których muszą być różne sygnały wyjściowe, to ten etap pomija się, w przeciwnym wypadku należy uzupełnić schemat systemu sterowania elementami pamięci. Należy określić liczbę elementów pamięci i miejsce elementów.
- Otrzymanie struktury systemu sterowania. System sterowania składa się do tej pory z elementów powiązanych ze sobą hierarchiczną strukturą. Każdy element systemu jest opisany w języku "tablic stanów". Te same elementy we/wy systemu mogą wchodzić do różnych podsystemów. Należy opisać cały system funkcją przejść.
- Analiza systemu w normalnych warunkach pracy. Analiza ma zapewnić realizację żądanych funkcji przez system.
- Analiza systemu w obecności zakłóceń. Dostarcza informacji o zachowaniu się systemu w warunkach:
  - nastąpiła zmiana struktury w wyniku zmiany liczby przekaźników,
  - nastąpiło przekłamanie sygnałów wejściowych,
  - nastąpiło przekłamanie sygnałów wyjściowych.
- Wybór aparatury. Do tej pory system sterowania jest określony w postaci schematu logicznego. Należy teraz wybrać fizyczne elementy realizujące elementy logiczne.
- Rozmieszczenie aparatury w panelach.
- Specyfikacje sprzętu i schematy montażowe.
- Sporządzenie dokumentacji graficznej.

Model procesu projektowania automatycznej regulacji i kontroli procesów technologicznych w rafinerii - M2

Projektowanie wg pracy [17] można rozpatrywać jako system o określonej strukturze związków między elementami systemu. Interesująca jest jedynie - zdaniem autorów - struktura systemu, a nie cechy fizyczne elementów systemu.

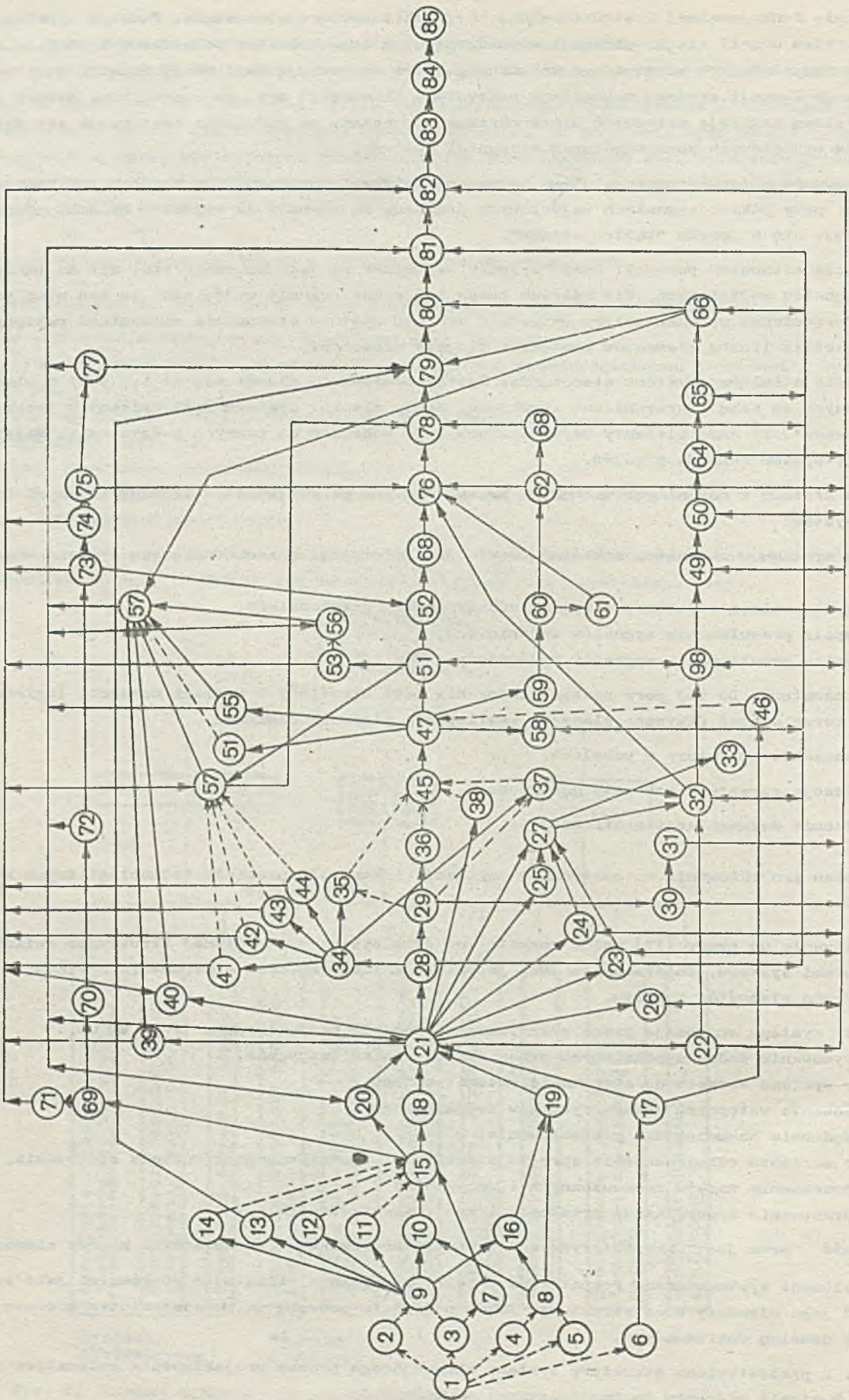
Elementy systemu oznaczają prace wykonywane w procesie projektowania (rys. 2) np.:

- 9 - sprecyzowanie zadań wypełnianych przez sterowanie automatyczne,
- 12 - wybór systemu sprzętu do sterowania automatycznego,
- 13 - opracowanie wstępnych opisów rysunków technicznych,
- 14 - sporządzenie harmonogramu projektowania,
- 51 - wybór wariantu rozmieszczenia sprzętu sterowania automatycznego w punkcie sterowania,
- 62,63 - opracowanie znaków mnemonicznych i schematów,
- 65,66 - opracowanie specyfikacji przewodów i kabli elektrycznych.

Złożoność prac jest scharakteryzowana przez liczbę związków zachodzących między elementami.

Każdy element systemu można rozpatrywać z kolei oddzielnie, traktując go również jako system i określić jego elementy oraz strukturę. Takie podejście pozwala na przedstawienie procesu projektowania z dowolną dokładnością.

Na rys. 2 przedstawiono strukturę systemu obrazującego proces projektowania automatycznej regulacji i kontroli procesów technologicznych w rafinerii.



Rys. 2. Model strukturalny procesu projektovania

Model projektowania systemów automatycznego sterowania - M3

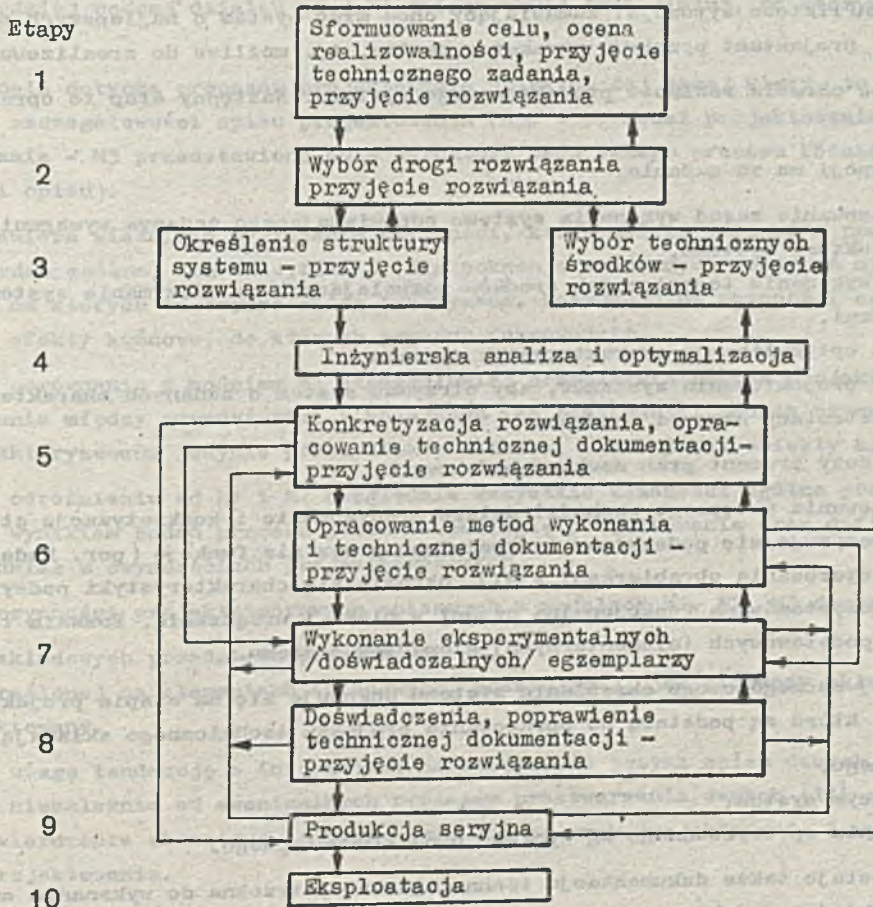
Proces projektowania systemów automatycznego sterowania przedstawia wg pracy [19] rys. 3. Etapy procesu połączone są strzałkami w dwu kierunkach, co ma oznaczać, że etapy mogą być powtarzane (proces projektowania jest iteracyjny). Grupując etapy w stadia uzyskuje się model ogólny projektowania systemów (rys. 4). Pierwszym etapem procesu projektowania jest formułowanie zadania technicznego. Zadanie techniczne zawiera informacje dotyczące:

- przeznaczenia systemu,
- warunków gabarytowych i masowych,
- wymagań dokładności, stabilności parametrów systemu,
- warunków eksploatacji systemu,
- wymagań niezawodności pracy systemu,
- warunków produkcji,
- terminu zakończenia projektowania,

Pierwszą postać technicznego zadania formułuje zamawiający system. Projektant systemu analizuje zadanie z punktu widzenia:

- możliwości zrealizowania takiego systemu,
- potrzeby podstawowych prac naukowo-badawczych,
- pracochłonności systemu,
- możliwości dokładniejszego sprecyzowania wymagań nakładanych na system.

Zadanie techniczne, zmodyfikowane przez projektanta, musi być akceptowane przez zamawiającego



Rys. 3. Podstawowe etapy projektowania systemu sterowania automatycznego [19]



- karty technologiczne,
- oprzyrządowanie i wyposażenie technologiczne.

Technologię należy opracowywać w miarę wcześniej biorąc pod uwagę warunki i możliwości zakładu produkcyjnego. Opracowanie technologii po wykonaniu dokumentacji technicznej utrudnia i opóźnia opanowanie produkcji.

W procesie seryjnej produkcji sprawdza się ostatecznie przyjęte rozwiązania techniczne i opracowuje się proces produkcji uwzględniając specyficzne własności produkcji seryjnej.

W procesie eksploatacji projektant systemu uzyskuje informację pozwalającą dokonać modyfikacji systemu w celu doprowadzenia parametrów systemu do parametrów zadanych.

Każdy etap zawiera pięć procedur projektowych:

- syntezę,
- sporządzenie modelu abstrakcyjnego obiektu projektowanego,
- sporządzenie modelu fizycznego obiektu projektowanego,
- analizę,
- przyjęcie rozwiązania.

Przy przechodzeniu z etapu na etap zachodzi uszczegółowianie modeli obiektu projektowanego i pogłębianie analizy, co powoduje przybliżanie się systemu do wymaganych (w zadaniu technicznym) charakterystyk.

#### Porównanie modeli

Z przedstawionych poprzednio modeli procesu projektowania, korzystano przy wprowadzaniu mc do projektowania układów sterowania. Modele budowane były na zasadach metodologii projektowania, tzn. wypuklają bardziej rodzaj działań projektotwórczych od "materiału", na którym te działania zachodzą.

Chociaż modele dotyczą procesów projektowania obiektów tej samej klasy, to można zauważyć dużą rozpiętość szczegółowości opisu projektowania (np. w p. Model projektowania systemów automatycznego sterowania - M3 przedstawiono dwie struktury tego samego procesu różniące się stopniem szczegółowości opisu).

Model M1 zawiera właściwie wyliczenie czynności, które należy wykonać w procesie projektowania i dokonuje bardzo ogólnej ich klasyfikacji. Nie pokazuje wzajemnych powiązań między czynnościami, ani obiektów, na których te czynności są wykonywane. Poszczególne czynności scharakteryzowane są jedynie przez efekty końcowe, do których powinny doprowadzić.

Model M2 w porównaniu z modelem M1 szczegółowiej przedstawia proces projektowania. Pokazuje liczne powiązania między czynnościami i kolejność ich realizacji. Jednak czynności, podobnie jak w M1, są scharakteryzowane jedynie przez efekty końcowe, a nie przez obiekty biorące w nich udział.

Model M3 w odróżnieniu od M1 i M2 uwzględnia wszystkie własności ogólne procesu projektowania [ 9 ] będące wynikiem badań procesu przez metodologię projektowania oraz definiuje niektóre obiekty biorące udział w czynnościach projektotwórczych.

Wykonanie czynności projektotwórczych opisanych w modelach M1, M2, M3 doprowadza do ustalenia:

- elementów składowych przedmiotu projektowanego,
- relacji określonej na elementach składowych, pokazującej, jak elementy składowe tworzą przedmiot projektowany.

Biorąc pod uwagę tendencję w informatyce do stworzenia języka opisu danych, który pozwalałby opisywać dane niezależnie od ewentualnych procesów przetwarzania danych [21] można na podstawie powyższego stwierdzenia skonstruować taki język dla potrzeb projektowania i wykorzystać go do budowy modelu projektowania.

### 3. Propozycja modelu procesu projektowania technicznego

#### Podstawowe założenia

W punkcie 1 przedstawiono tożę, że przy komputerowym wspomaganii projektowania należy proces projektowania wyrazić na gruncie pojęcia języka, podstawowego pojęcia informatyki. W punkcie 2 przedstawiono modele procesu projektowania wykorzystane do wprowadzania mc do projektowania, a także podane uzasadnienie nieprzydatności tych modeli przy realizacji oprogramowywania, mającego służyć wspomaganii projektowania.

Obecnie na bazie tych punktów będą sformułowane założenia, na podstawie których tworzony jest model procesu projektowania zgodny z językami spotykanymi w informatyce (problem założeń, i idealizacji w badaniach porusza praca [14]).

Założenie 1. Projektowanie traktuje się jako tworzenie opisu przedmiotu projektowanego w pewnym języku L.

Założenie 2. Z każdym opisem związany jest obraz przedmiotu projektowanego.

Założenie 3. Obraz składa się z elementów obrazu postaci (nazwa, wartość). Nazwa pozwala na identyfikację elementu obrazu w zbiorze podobnych elementów, drugi składnik pary - wartość odpowiada fragmentowi informacji o przedmiocie projektowanym.

Założenie 4. Wartość należy do zbioru dopuszczalnych wartości zawierającego co najmniej dwa elementy: nie dotyczy, brak informacji.

Przyjęcie założenia 1 i 2 jest wynikiem rozróżniania w informacji dwóch składników: składnika fizycznego i składnika semantycznego obejmującego treść znaczeniową informacji.

Założenie 3 wprowadza postać obiektów odpowiadającą definicji danej [20], na których ewentualnie dokonywać się będzie operacji. Dla uzasadnienia przyjęcia założenia 4 podany model procesu projektowania form wtryskowych przedstawiony w pracy [23]. Proces projektowania jest podzielony na 13 etapów. Każdy etap zawiera rozwiązania techniczne, które są przyjmowane w wyniku wykonania pewnych czynności na tym etapie, np.:

- etap 2 - wybór typu formy
  - forma prosta?
  - forma z segmentami ruchomymi?
  - kombinacja tych typów?
- etap 8 - sposoby napędu ruchomych elementów
  - mechaniczny?
  - hydrauliczny?
  - pneumatyczny?
  - elektryczny?

Są to zbiory wartości w postaci niejawniej. Element zbioru wartości "nie dotyczy" oznacza, że w tym projekcie nie ma takiego rozwiązania, natomiast "brak informacji" oznacza, że jeszcze nie podjęto decyzji - nie ustalono wymienionego w "nazwie" szczegółu projektu.

#### Zarys formalny proponowanego modelu

Model procesu projektowania wg założenia 1 budowany jest na podstawie języka. Formalna definicja języka podana w pracy [12] ma postać:

$$L = \langle A, W, D, Z \rangle$$

A - alfabet języka,

W - zbiór wyrażeń poprawnych języka,  $W \subseteq A^*$

D - dziedzina języka,

Z - znaczenie języka,  $Z \subseteq W \times D$

Założenie 3 sugeruje następującą interpretację dziedziny języka:

$$D = \langle N, O_b \rangle$$

N - zbiór nazw,  $N = N_C \cup N_E \cup N_R$

N<sub>C</sub> - zbiór nazw cech,

N<sub>E</sub> - zbiór nazw elementów składowych,

NR - zbiór nazw relacji elementów składowych,

$O_b$  - zbiór obrazów,  $O_b \subseteq O_b^*$  - przestrzeń obrazów,

$O_b^* = P \cup E \cup R$  więc  $O_b = P' \cup E' \cup R'$ ,  $P' \subseteq P$ ,  $E' \subseteq E$ ,  $R' \subseteq R$

$P = P_1 \times P_2 \times \dots \times P_n$  - przestrzeń wartości cech,

$P_i$ ,  $i = 1, \dots, n$  - zbiór dopuszczalnych wartości cechy  $i$ , zawiera co najmniej dwa elementy: "nie dotyczy", "brak informacji",

$E = E_1 \times E_2 \times \dots \times E_m$  - przestrzeń wartości elementów składowych,

$E_j$ ,  $j = 1, \dots, m$  - zbiór dopuszczalnych wartości elementu składowego  $j$ , zawiera co najmniej dwa elementy: "nie dotyczy", "brak informacji",

$R = R_1 \times R_2 \times \dots \times R_k$  - przestrzeń wartości relacji elementów składowych,

$R_l$ ,  $l = 1, \dots, k$  - zbiór dopuszczalnych wartości relacji  $l$ , elementów składowych, zawiera co najmniej dwa elementy: "nie dotyczy", "brak informacji".

Przy tak określonej dziedzinie języka elementu zbioru  $Z$  mogą być następujące:

$$\begin{aligned} z \in Z \rightarrow z = \langle w, d \rangle \langle \rangle w \in W \wedge (d = \langle NC, p \in P' \rangle \vee \\ d = \langle NE, o \in E' \rangle \vee d = \langle NE \cup NR, a \in E' \cup R' \rangle \vee \\ d = \langle N, o \in O_b \rangle) \end{aligned}$$

Oznacza to, że opis obiektu projektowanego może zawierać obraz składający się z samych cech, (taki obraz nazwany jest obrazem prostym), lub zawiera obraz składający się tylko z elementów składowych (taki obraz nazwany jest specyfikacjami) lub zawiera obraz zawierający elementy składowe i relacje na nich określone (obraz morfologiczny), lub zawiera obraz składający się z cech, elementów składowych i relacji (obraz zupełny).

Opis obiektu projektowanego można sporządzić uwzględniając wyżej wymienione rodzaje obrazów, a dla każdego rodzaju obrazu można przy opisie uwzględniać różne jego elementy. Opis należy zatem zwińzać z kryterium, dla którego jest tworzony obraz. Ale nie tylko. W procesie form wtryskowych [23] wymieniono różną postać dokumentacji konstrukcyjnej. Wskazuje to, że opis zależy także od tego dla kogo jest tworzony. Ta sama konstrukcja może być różnie opisana, w zależności od poziomu systemu, który ją będzie wytwarzał.

Można więc przyjąć, że proces projektowania jest to sporządzenie opisu obiektu projektowanego w języku  $L$  wg kryterium "wytwarzanie" na podstawie opisu obiektu wg kryterium "potrzeby".

Automatyzacja procesu projektowania polega na określeniu algorytmu przetwarzania obrazu zawartego w opisie wg kryterium "potrzeby" na obraz zawarty w opisie wg kryterium "wytwarzanie". Przetworzenie to wynika z zależności między wartościami elementów ustalonych na podstawie praw fizyki, norm, badań doświadczalnych itd.

#### 4. Uwagi końcowe

Zarysowany w poprzednich punktach model procesu projektowania posłużył do automatyzacji projektowania napowietrznych linii elektroenergetycznych. Prace nad automatyzacją procesu projektowania prowadzone były wg schematu:

- ustalenie języka  $L$  dla procesu projektowania napowietrznych linii elektroenergetycznych,
- ustalenie obrazu linii elektroenergetycznej wg kryterium "potrzeby" i wg kryterium "wytwarzanie",
- ustalenie zbioru obrazów pośrednich występujących między obrazami wg kryterium "potrzeby" i "wytwarzanie",
- napisanie programów przekształcających obrazę na podstawie istniejących zależności między wartościami elementów wynikających z praw fizyki, norm itd.

Przedstawiony model może być także przydatny do budowy ogólnego systemu informatycznego automatyzującego szeroką klasę procesów projektowych. System informatyczny musiałby umożliwiać definiowanie opisów obiektów projektowanych i operacji przekształcających obrazy zawarte w opisach (niektóre przekształcenia mógłby wykonywać projektant). W zależności od tego jak bogaty język L znalazły projektant - mógłby uzyskiwać różny stopień automatyzacji procesu projektowania: od prostych obliczeń, aż do integracji procesu projektowania konstrukcji i technologii. Przesłanki do takiej integracji przedstawiono poniżej [24].

W czasie życia wyrobu można wyróżnić 6 faz (ze względu na to, że wyrób jest elementem różnych systemów):

- koncepcja,
- projektowanie konstrukcji,
- opracowanie technologii,
- wytwarzanie,
- eksploatacja i użytkowanie,
- unicostwienie.

Podczas procesu projektowania można podejmować prace nad opracowaniem technologii biorąc pod uwagę możliwość systemu wytwarzania, a także, podczas projektowania konstrukcji maszyn i urządzeń powinno uwzględniać się podstawowe wymagania metod wytwarzania, np. spawania, odlewania, kucia itp. W każdej z faz projektowania (rys. 5) zróżnicowany jest poziom informacji o konstrukcji będący podstawą opracowania technologii [10]. Poniżej przedstawiono fazy procesu i zakres możliwych prac nad opracowaniem technologii.

● Projekt wstępny. W fazie projektu wstępnego precyzowany jest podstawowy schemat konstrukcji, ogólny układ i wymiary gabarytowe. Wyrób zostaje podzielony na podstawowe zespoły i sekcje. Daje to podstawy do:

- ustalenia niezbędnych badań i opracowań nowych procesów technologicznych,
- zaprojektowania lub adaptowania oprzyrządowania specjalnego,
- opracowania nowych procesów technologicznych,
- unifikacji poszczególnych zespołów,
- organizacji montażu wyrobu,
- doboru materiału i rodzaju surowców.

● Projekt techniczny. Koncepcja konstrukcyjna jest coraz szczegółowiej rozwijana, uściśla się wymiary i kształty podstawowych części. Daje to podstawy do:

- ustalania sposobu wykonania podstawowych zespołów i części,
- sprecyzowania rodzaju i kolejności prac montażowych,
- wyeliminowania lub ograniczenia docierania części,
- ustalenia rodzaju powłok i obróbki cieplnej,
- wykonania unikalnego wyposażenia technologicznego,
- ustalenia zakresu i metod kontroli.

● Projekt roboczy. Konstrukcja wyrobu jest ostatecznie sprecyzowana i rozwinięta. Dla wszystkich części są określone kształty i wymiary, tolerancje wykonawcze i jakość powierzchni. Daje to podstawy do:

- unifikacji konstrukcji,
- typizacji procesów technologicznych,
- przeprowadzania intensywnych badań (przy zastosowaniu materiałów niezupełnie znanych) w celu zapobieżenia trudnościom produkcyjnym.

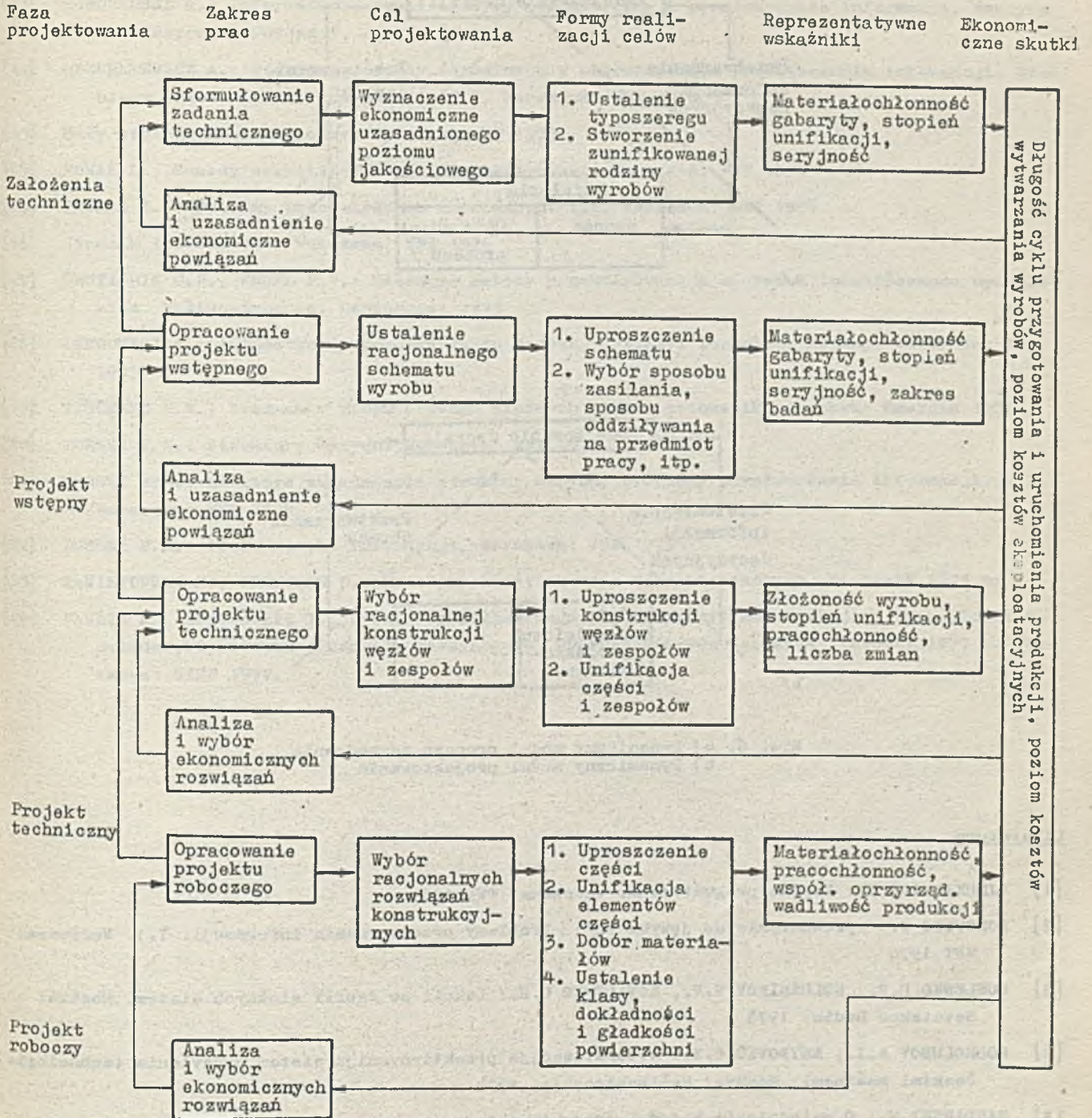
Pośrednią formą użytkowania systemu informatycznego byłoby użytkowanie systemu wynikające z ogólnych własności procesu projektowania [9].

Wspomaganie procesu projektowania nie może dotyczyć:

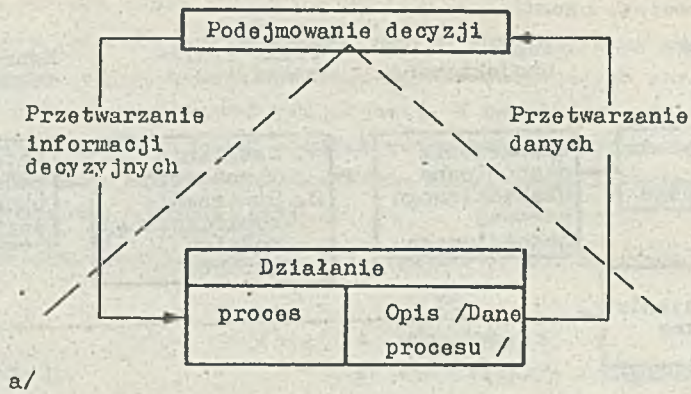
- automatyzacji szeregu działań składowych procesu,
- pomocy przy analizie, syntezie i ocenie.

Wspomaganie projektowania oparte na prezentowanym modelu można porównać do modelu przetwarzania danych przedstawionego w pracy [18]. Model obrazuje rys. 6, w którym utrzymano terminologię użytą w podanej pracy.

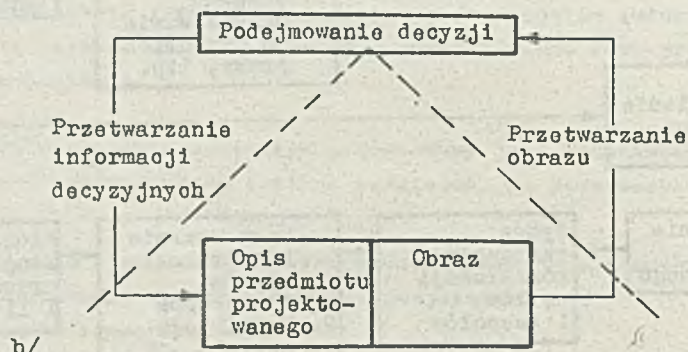




Rys. 5. Schemat blokowy procesu projektowania konstrukcji



a/



b/

Rys. 6. a) Dynamiczny model procesu zarządzania  
b) Dynamiczny model projektowania

#### Literatura

- [1] AJDUKIEWICZ K.: Logika pragmatyczna. Warszawa: PWN 1965
- [2] BOROWIEC J.: Wprowadzenie do języka PL/1. Problemy przetwarzania informacji. T.1. Warszawa: WNT 1970
- [3] BUSLENKO H.P., KOLUŠNIKOV V.V., KOVALENKO U.H.: Lekcii po teorii složnych sistem. Moskva: Sovetskoe Radio 1973
- [4] DOBROLUBOV A.I., AKYNOVIČ C.I.: Avtomatizacija proektirovanija sistem upravlenia technologi-českimi mašinami. Moskva: Mašinstroenie 1974
- [5] GASPARSKI W.: O metodologii badań i projektowania systemowego. II Konferencja Metodologii Projektowania. Warszawa: PWN 1974
- [6] GASPARSKI W.: Sytuacja problemowa w projektowaniu (Procedura i język). Prakseologia 1976 nr 2/58
- [7] GASPARSKI W.: Wiedza o projektowaniu a jego wspomaganie. Prace naukowe ICT Politechniki Wrocławskiej 1977 nr 15
- [8] GREGORY S.A.: Nauka o projektowaniu. Prakseologia 1972 nr 41
- [9] JELENIEWSKI T., SIELICKI A.: Metodologia i komputerowe wspomaganie projektowania technicznego. Wrocław: Politechnika Wroclawska 1975
- [10] KASPRZYK S.: Analiza technologiczności konstrukcji wyrobu. Mechanik 1976 nr 5

- [11] MARCZYŃSKI R.: Informatyka, czyli maszyny matematyczne i przetwarzanie informacji. Maszyny Matematyczne 1969 nr 1
- [12] MAZURKIEWICZ A.: Problemy języków formalnych w automatycznym przetwarzaniu informacji. Problemy przetwarzania informacji. T.1. Warszawa: WNT 1970
- [13] Mały słownik cybernetyczny. Warszawa: WP 1973
- [14] NOWAK L.: Zasady marksistowskiej filozofii nauki. Warszawa: PWN 1974
- [15] PAWLAK Z.: Problemy przetwarzania informacji. T.2. Warszawa: WNT 1974
- [16] Słownik informatyki. Warszawa: WNT 1976
- [17] ŠESTICHIN O.F., ENGEL P.V.: Mašinnye metody proektirovanija sistem avtomatičeskogo upravlenija. Mašinostroenie, Leningrad: 1973
- [18] TARGOWSKI A.: Automatyczne przetwarzanie danych. Systemy. Technika. Metody. Warszawa: PWN 1973
- [19] TIŠČENKO N.V.: Vvedenie v proektirovanie složnych sistem avtomatiki. Moskva: Energia 1976
- [20] TURSKI W.M.: Struktury danych. Warszawa: WNT 1971
- [21] TURSKI W.M.: Niektóre zagadnienia struktur danych. Problemy przetwarzania informacji. T.2. Warszawa: WNT 1974
- [22] TURSKI W.M.: Propedeutyka informacji. Warszawa: PWN 1975
- [23] ZAWISTOWSKI H., FRENKLER D.: Metodyka konstruowania form wtryskowych. Mechanik 1975 nr 1
- [24] PAWLIK R., GONTARCZYK T., MOCAŁA J.: Integracja projektowania konstrukcji i technologii w przedsiębiorstwach przemysłu maszynowego. COMPCONTROL'77. Warszawa 14-18.XI.1977 r. Warszawa: SIMP 1977.



dr inż. Ryszard PAWLIK  
Instytut Maszyn Matematycznych

## Podstawowe wymagania stawiane programom konwersacyjnym

### 1. Wstęp

Zgodnie z tradycją, mówiąc o narzędziach projektanta, mamy na myśli przede wszystkim deskę kreślarską, suwak logarytmiczny, przyrządy kreślarskie itp. Obecnie do zbioru tych narzędzi należy włączyć także maszynę cyfrową wraz z jej bogatym wyposażeniem w urządzenia peryferyjne, szczególnie takie jak monitory ekranowe, plottery, digitizery, nie mówiąc już o tak podstawowych, jak drukarka, stacje taśm i dysków, czytniki kart czy taśm perforowanych, bez których maszyna cyfrowa jest prawie bezużytecznym narzędziem. Pojawienie się tego nowego narzędzia, bardzo drogiego i skomplikowanego, stworzyło potrzebę badań nad metodami jego racjonalnego wykorzystywania. Jedną z bardziej przydatnych metod współpracy człowieka z maszyną cyfrową, szczególnie w procesie projektowania, jest praca w trybie konwersacyjnym.

Należy więc przewidywać, że w najbliższych latach poważnie wzrośnie zapotrzebowanie na programy konwersacyjne. Przyczyni się do tego znaczny wzrost liczby sprzętu cyfrowego, umożliwiającego ekonomiczną realizację programów konwersacyjnych. Dotyczy to szczególnie wszelkiego rodzaju mini-komputerów, które szczególnie nadają się do tego typu pracy. Ponadto należy oczekiwać wzrostu liczby średnich i dużych maszyn cyfrowych wyposażonych w systemy wielodostępne, umożliwiających realizację prac w trybie konwersacyjnym.

W niniejszym artykule omówiono niektóre zagadnienia związane z projektowaniem programów konwersacyjnych, mające wpływ na jakość tego programu.

Problem projektowania dobrych programów konwersacyjnych jest zadaniem bardzo trudnym i złożonym. Projektanci programów, czy też systemów konwersacyjnych muszą rozwiązać wiele problemów nieznanych lub nie tak istotnych przy tworzeniu programów tradycyjnych.

Podstawowym problemem jest zapewnienie optymalnego średniego czasu reakcji. Jest to w programach konwersacyjnych bardzo ważne zagadnienie, decydujące o powodzeniu całej pracy włożonej w opracowywanie programu (punkt 4.).

Drugim podstawowym zagadnieniem, na które należy zwrócić szczególną uwagę przy pracy nad programami konwersacyjnymi jest odporność tych programów na błędy popełniane przez użytkownika. Jest to oczywiście ważne nie tylko w programach konwersacyjnych, lecz w tego typu programach jest szczególnie istotne (punkt 5.).

### 2. Rola programów konwersacyjnych w komputeryzacji procesu projektowania

W klasycznej już metodzie przetwarzania danych maszyna realizuje ustalony program i podczas jego wykonywania w zasadzie nie ma miejsca na interwencję człowieka. Projektantowi pozostaje jedynie zadanie dostarczenia danych i odbioru wyników (przetwarzanie wsadowe). Metoda ta, obecnie najczęściej stosowana, jest dobra i efektywna dla pewnej klasy zadań projektowych. Do klasy tej należą zadania o dobrze znanym i jednoznacznie ustalonym procesie projektowania, dającym się łatwo algorytmizować. W efekcie otrzymuje się rozwiązania typowe, nieoryginalne, a jedynie tylko czas ich otrzymania jest szybszy od realizacji sposobami tradycyjnymi i dokładniej mogą być policzone.

Metoda ta jest prawie całkowicie nieprzydatna dla tych procesów projektowania, których struktura nie jest dokładnie znana, a więc nie można ich także w pełni zalgorytmizować. Maszyna cyfrowa wówczas może jedynie wspomagać proces projektowania. Dla projektantów właśnie te nie w pełni

określono procesy są szczególnie interesujące, ponieważ w wyniku działania tych właśnie procesów otrzymuje się konstrukcje nietypowe - oryginalne.

Z analizy przebiegu procesu projektowania wynika [4], że obliczenia w pełnym tego słowa znaczeniu zajmują niewielką część ogółu operacji przetwarzania danych w tym procesie. Najwięcej wysiłku przysparza poszukiwanie rozwiązań zbliżonych, co często ma miejsce w zadaniach optymalizacyjnych, przeszukiwanie katalogów, poszukiwanie danych liczbowych, a następnie analiza i interpretacja zebranych danych. Sprawność procesu projektowego w dużej mierze zależy od sprawnego manipulowania dużą ilością danych oraz od łatwości dostępu do dowolnej informacji w dowolnej chwili. Dotyczy to szczególnie wspomnianych procesów dla zadań nietypowych, nie w pełni zalgorytmizowanych.

W pracach tego typu, w tym w przetwarzaniu dużej ilości danych, maszyny cyfrowe są praktycznie niezbędne. Należy jednak zadbać o to, żeby praca projektanta z maszyną cyfrową była łatwa, efektywna i przyjemna.

Jak już było powiedziano, trudno, nie w pełni zalgorytmizowane procesy projektowe są nieciągłe. Oznacza to, że występują w nim fazy obliczeń, fazy przeszukiwania zbiorów danych oraz fazy podejmowania decyzji. Projektant na podstawie wyników częściowych podejmuje decyzje o dalszym przebiegu procesu projektowania. Wykorzystywanie maszyny w trybie wsadowym jest w takiej sytuacji prawie niemożliwe, ponieważ utrudnia to i opóźnia proces projektowania. Jedynym efektywnym sposobem współpracy człowieka z maszyną cyfrową jest wtedy praca w trybie konwersacyjnym.

Jak już powiedziano komputerowe wspomaganie procesu projektowania jest bardziej efektywne przy konwersacyjnej pracy człowieka z maszyną cyfrową. Warto więc zastanowić się, jakie czynniki wpływają na jakość programu konwersacyjnego, co decyduje o tym, że jeden program jest lepszy od drugiego? W.M. Turski w pracy [5] pisze, że dobry program powinien być:

- poprawny,
- przystosowalny,
- odporny,
- stabilny.

Oczywiste jest, że warunki to dotyczą także programów konwersacyjnych, lecz nie wyczerpują zagadnień związanych z jakością tych programów. Dla oceny jakości programów eksploatowanych w trybie konwersacyjnym należy uwzględnić bardzo istotny element działania takiego programu, a mianowicie - człowieka.

Działający program konwersacyjny tworzy para człowiek - maszyna (program). W programach eksploatowanych wsadowo użytkownik ma kontakt z programem dwa razy: na początku - przy przygotowywaniu danych i na końcu - przy interpretacji wyników. Przy pracy w trybie konwersacyjnym tak nie jest, użytkownik i program tworzą jeden "organizm" wzajemnie się uzupełniając i nie mogą pracować niezależnie od siebie. Program konwersacyjny ma współpracować z człowiekiem i projektanci tych programów nie mogą o tym zapominać.

Wychodząc z tego podstawowego założenia przy opracowywaniu programów konwersacyjnych należy uwzględniać możliwie najwięcej cech obu partnerów układu człowiek - maszyna. Człowiek w tym układzie szczególnie w procesie komputerowo wspomaganego projektowania spełnia rolę wiodącą i maszyna nie może mu w pracy przeszkadzać, a odwrotnie, powinna go zmusić do intensywnej pracy bez szkody dla jej jakości. Może to nastąpić wyłącznie wtedy, gdy przy projektowaniu programu konwersacyjnego zostaną uwzględnione psycho-fizyczne możliwości człowieka.

Jak twierdzą psychologowie człowiek wykorzystuje bardzo małą część potencjalnych możliwości swego mózgu. Poprawny program konwersacyjny może wydatnie zwiększyć sprawność działania człowieka, pod warunkiem, że będzie uwzględniał jego możliwości.

Do najważniejszych czynników wpływających na jakość programu konwersacyjnego, a wynikających z uwzględnienia cech psycho-fizycznych człowieka należą:

- budowa dialogu,
- czas reakcji,
- odporność na błędy.

### 3. Budowa dialogu

Przystępując do projektowania dialogu człowieka z maszyną cyfrową należy na wstępie określić budowę tego dialogu. J. Martin <sup>1</sup> podaje 23 metody prowadzenia dialogu za pomocą alfanumerycznych monitorów ekranowych. Podział ten jest bardzo szczegółowy i jednocześnie sztuczny, a kryterium tego podziału jest konkretna technika realizacji dialogu. Z doświadczeń autora niniejszego artykułu wynika, że nie warto tak szczegółowo dzielić metod prowadzenia dialogu. Projektant dialogu z reguły ograniczony jest dość mocno sprzętem, którym dysponuje i zwykle wybiera wariant optymalny dla danego sprzętu. Przy czym mówiąc o sprzęcie ma się na myśli również oprogramowanie podstawowe. Colowe jest natomiast określanie typu projektowanego dialogu wyłącznie z punktu widzenia użytkownika, a nie techniki realizacji. Biorąc za kryterium podziału aktywność użytkownika podczas prowadzenia dialogu, można wprowadzić dwa podstawowe typy dialogów:

- bierny użytkownik - użytkownik odpowiada na pytania stawiane przez maszynę,
- czynny użytkownik - maszyna wykonuje polecenia użytkownika.

Oczywiście w praktyce, szczególnie przy bardziej skomplikowanych zadaniach, w jednym dialogu mogą wystąpić obydwa typy razem, będzie to wtedy dialog mieszany. W dialogu takim, w jednej fazie stroną aktywną może być maszyna, a w drugim użytkownik.

Budowa projektowanego dialogu ma duży wpływ na jego jakość, a więc na ocenę programu przez użytkownika. Warto zatem zastanowić się co ma wpływ na to, że raz lepszy jest dialog bierny, a innym razem czynny. Można tu wskazać dwa główne czynniki decydujące o wyborze typu dialogu. Jednym z nich, najważniejszym jest rozwiązywany problem, a drugim przewidywane możliwości potencjalnego użytkownika.

Z analizy przebiegu procesu rozwiązywania różnego rodzaju zadań za pomocą maszyn cyfrowych wynika, że zadania te można podzielić na dwie podstawowe klasy.

Do pierwszej klasy można zaliczyć te zadania, dla których proces ich rozwiązywania jest w pełni określony. Dla takich zadań wydaje się, że dobre są dialogi typu biernego. Jeżeli ponadto będą to zadania, dla których kolejność wprowadzania danych oraz ich struktura są ściśle określone, to jedyne dialogi bierne będą efektywne. Dialog tego typu uwalnia użytkownika od kłopotliwego i niekiedy trudnego pamiętania o strukturze danych i kolejności ich wprowadzania, a więc pozwala na efektywną, szybką i bezbłędną pracę.

Użytkownik, odpowiadając na pytania zadawane przez maszynę wprowadza do programu dane oraz steruje przebiegiem procesu przetwarzania tych danych zwracając szczególną uwagę na te fragmenty, w których należy podejmować decyzje. Maszyna stawiając użytkownikowi poszczególne pytania, przedstawia mu możliwie bogaty zestaw dotychczas uzyskanych wyników, na podstawie których może on podjąć decyzje dotyczące dalszego przebiegu procesu projektowania. Tak zorganizowany dialog nie wymaga od użytkownika dużej inicjatywy, ponieważ nie ma on dużego wpływu na przebieg procesu projektowania. Programy konwersacyjne o opisanej powyżej organizacji dialogu będzie wtedy mogli obsługiwać użytkownik o nieco mniejszych kwalifikacjach.

Rozwiązując tego typu zadania konwersacyjnie, zyskujemy na czasie w stosunku do pracy wsadowej, ponieważ unika się przestojów związanych z przerwami na analizę częściowych wyników i podejmowanie decyzji.

Do drugiej klasy zadań należy zaliczyć te zadania, których proces rozwiązywania nie jest dokładnie znany, a więc nie dają się one w pełni algorytmizować. Wydaje się, że dla takich zadań lepszy jest dialog typu czynnego. Można wtedy zaprogramować te elementy przetwarzania danych, których struktura jest w pełni określona, a proces obliczeniowy ustalony. Użytkownik wykorzystuje maszynę do tych czynności, które może on wykonać dając jej polecenia w rodzaju: "wykonaj ...", "policz ..." itp., a maszyna po wykonaniu tych poleceń czeka na dalsze dyspozycje. Organizacja taka ma tę zaletę, że przebieg procesu rozwiązywania zadania zależy tylko od użytkownika, maszyna nie narzuca mu kolejności działania. Ponadto organizacja taka zapewnia łatwość rozbudowy systemu w miarę coraz lepszego poznawania i określania poszczególnych jego elementów.

#### 4. Czas reakcji

Drugim podstawowym problemem, jaki musi rozwiązać projektant dialogu człowieka z maszyną cyfrową jest zapewnienie optymalnego średniego czasu reakcji maszyny.

Rozróżnia się trzy poziomy, na których bada się czas reakcji systemu (p. 6) :

- poziom leksykalny,
- poziom syntaktyczny,
- poziom sementyczny.

Poziom leksykalny jest interesujący dla producentów urządzeń końcowych oraz programistów systemowych i nie będziemy się nim zajmować. Poziom syntaktyczny głównie dotyczy programistów systemów, twórców systemów operacyjnych, natomiast programistów systemów użytkowych w niewielkim stopniu. Można jedynie powiedzieć, że maksymalnym czasem reakcji na tym poziomie jest czas ok. 1 s.

Programistów systemów użytkowych, tzn. systemów przeznaczonych do rozwiązywania konkretnych problemów obliczeniowych czy problemów przetwarzania danych typu informacyjnego, a więc obsługi wszelkiego rodzaju banków danych itp. interesuje przede wszystkim poziom sementyczny. Poziom sementyczny dotyczy czasu reakcji maszyny na podstawowe dla danego typu rozwiązywanego problemu, zadania stawiane maszynie.

Czas reakcji maszyny jest to czas, który upływa między naciśnięciem ostatniego klawisza przez użytkownika, a uzyskaniem pierwszego znaku odpowiedzi maszyny (p. 1).

Według badań przeprowadzonych przez psychologów amerykańskich wynika, że optymalny jest średni czas reakcji od 1 do 4 s. [2]. Czasy dłuższe niż 4 sekundy są zwykle zbyt długie, nużą użytkownika. Mogą one powodować zdenerwowanie użytkownika, co nie sprzyja efektywnej pracy. Czasy dłuższe niż 15 sekund wykluczają możliwość pracy konwersacyjnej. Stwierdzono także, że z psychologicznego punktu widzenia zbyt krótkie czasy reakcji także są niekorzystne. Użytkownik nie nadąża za maszyną, co go niepokoi i zmusza do szybszej pracy, co jest źródłem dodatkowych błędów.

Średni czas reakcji maszyny ma bardzo duży wpływ na jakość pracy użytkownika. Prawidłowo dobrany średni czas reakcji zapewnia pełne wykorzystanie możliwości projektanta, a tym samym może decydować o efektach projektowania. Źle dobrany średni czas reakcji a szczególnie zbyt duży, zmniejsza zainteresowanie użytkownika do pracy konwersacyjnej. Zbyteczne wydłużenie czasu reakcji rozprasza użytkownika powodując, że zaczyna on interesować się innymi, nieistotnymi dla aktualnego procesu projektowego sprawami. Na skutek tego projektant, gdy w końcu uzyska odpowiedź i musi podjąć decyzję, może podjąć nie najlepszą lub co gorsze błędną decyzję niwecząc w ten sposób dotychczas wykonaną pracę.

Utrzymanie średniego czasu reakcji w optymalnych granicach nie jest sprawą prostą. Do podstawowych czynników mających wpływ na czas reakcji maszyny należą:

- rozwiązywany problem,
- organizacja programu,
- sprzęt, na którym program jest eksploatowany,
- otoczenie, w którym program jest eksploatowany.

Projektant dialogu nie ma oczywiście wpływu na rozwiązywany problem. Może jedynie starać się o przedstawienie przebiegu procesu przetwarzania danych w postaci najlepszej dla konwersacji. Mamy tu na myśli głównie rozbicie obliczeń na takie fragmenty, które pozwolą na utrzymanie średniego czasu reakcji maszyny w optymalnych granicach. Należy jednak brać tu pod uwagę wpływ szybkości działania maszyny, na której eksploatowany będzie dany program. Niezbędne jest tu doświadczenie programisty. W niektórych bardziej skomplikowanych sytuacjach celowe może być wykonanie nawet doświadczeń polegających na praktycznym określeniu czasu trwania obliczeń poszczególnych fragmentów programu.

Podział programu na elementy, dla których czas obliczeń mieści się w odpowiednich granicach z jednej strony, a stanowiące jednocześnie element dialogu odpowiadający jednej odpowiedzi nie jest proste. Odpowiednim narzędziem do takiej analizy może być dynamiczny profil programu. Dynamiczny profil programu określa względny czas wykonywania jego poszczególnych części [3]. Analizując dy-



namieniony profil programu można zorientować się, które jego fragmenty będą wykonywane wolno, a które szybko. Projektując dialog należy próbować rozbić te części programu, które będą wykonywane zbyt długo, oczywiście jeżeli jest to celowe z punktu widzenia budowy dialogu.

Powstaje jednak pytanie, co zrobić, gdy przewidywany czas reakcji jest zbyt długi, a danego fragmentu nie można w sposób sensowny rozbić na części. Niektórzy z autorów np. [1], zalecają wówczas stosować tzw. odpowiedzi tymczasowe. Odpowiedź tymczasowa ma za zadanie zająć użytkownika i poinformować go, że wszystko przebiega normalnie i zaraz otrzyma odpowiedź właściwą, a także o to, by zatrzymać jego uwagę przy końcówce komputerowej, ponieważ każde odwrócenie uwagi użytkownika od rozwiązywanego problemu działa na niekorzyść rozwiązywanego zadania.

Należy jednak zdawać sobie sprawę z tego, że odpowiedzi tymczasowe są jedynie złem koniecznym i można je stosować dopiero po wyczerpaniu innych możliwości, ponieważ nie zapewniają ciągłości myślenia użytkownika, a nawet mogą mu przeszkadzać, jeżeli są zbyt często stosowane.

Warto również wyraźnie podkreślić, że nie każdy problem nadaje się do rozwiązywania w trybie konwersacyjnym. Jednym z czynników decydujących o tym, czy dany problem nadaje się do rozwiązywania w trybie konwersacyjnym może być właśnie czas reakcji. Jeżeli z analizy problemu wynika, że nie można utrzymać średniego czasu reakcji w optymalnych granicach, może to oznaczać, że rozwiązywany problem nie nadaje się do rozwiązywania w trybie konwersacyjnym, a bardziej celowe jest przewidzieć eksploatację programu w trybie wsadowym.

Przy analizie rozwiązywanego problemu i projektowaniu dialogu nie można pominąć także maszyny, na której eksploatowany będzie dany program. Jeden i ten sam program eksploatowany na dwóch różnych maszynach może dawać bardzo duże różnice w czasach reakcji. Wynika to z szybkości działania jednostki arytmetycznej, gdy program wykonuje dużo obliczeń matematycznych lub czasu dostępu do pamięci zewnętrznych przy częstych odwołaniach do niej, co ma miejsce przy przetwarzaniu dużej ilości danych. Z tego względu bardzo owocne mogą być doświadczenia z określeniem czasu wykonywania typowych dla danego problemu operacji. Pozwala to uniknąć potem przykrych doświadczeń, gdy czasy reakcji są złe i należy przerabiać gotowy program.

Nie można także przy projektowaniu programu konwersacyjnego pomijać otoczenia, w którym program będzie eksploatowany. Programy konwersacyjne z reguły eksploatowane są bądź w systemach wielodostępnych, szczególnie na maszynach o dużej i średniej mocy obliczeniowej. Otoczeniem programu konwersacyjnego nazywać będziemy wszystkich innych użytkowników mających dostęp do jednostki centralnej lub urządzeń peryferyjnych maszyny, na której eksploatowany jest rozważany program i mających wpływ na czas odpowiedzi maszyny na zadany jej problem do rozwiązania.

Dla przykładu można podać, że w eksploatowanym w IMM systemie wielodostępnym VM-CMS czas reakcji maszyny przy dziesięciu użytkownikach jest kilkakrotnie dłuższy niż przy jednym lub dwóch użytkownikach. Istotne jest także, jakie problemy rozwiązują ci użytkownicy. Świadczy to o tym, że problem otoczenia i jego wpływu na czas reakcji jest istotny i należy go uwzględniać.

Inaczej kształtują się te problemy gdy program eksploatowany jest w systemie minikomputerowym.

Należy także podkreślić, że problem wpływu maszyny i otoczenia, w którym eksploatowany ma być dany program jest bardzo ważny, szczególnie przy przenoszeniu programów konwersacyjnych z maszyny na maszynę. Nie zawsze i nie wszystkie programy konwersacyjne będzie można przenieść na inną maszynę, mimo zgodności języków programowania.

##### 5. Czulość na błędy popełniane przez użytkownika

Niezależnie od metody konwersacji użytkownika z maszyną cyfrową bardzo ważnym zagadnieniem jest odporność programu na błędy popełniane przez użytkownika podczas konwersacji. Nie istnieje idealny użytkownik, który nie popełnia błędów. Należy liczyć się z tym, że użytkownik będzie popełniał błędy. Będą to błędy dwójakiego rodzaju: pojawiające się podczas wykonywania programu i powstające przy wprowadzaniu danych (różnego typu pomyłki). Błędy powstające przy wykonywaniu programu niejednokrotnie użytkownikowi jest bardzo trudno zlokalizować. Należy więc tak zaplanować

wać program nby mógł przekazać wtedy użytkownikowi możliwie najbogatszą informację o rodzaju błędu i miejscu jego wystąpienia. Bardzo istotno jest zapewnienie możliwości wczesnego wykrywania wystąpienia różnego typu błędów. Należy unikać sytuacji, po których program przerywa działanie (pada). Pojawiają się wtedy komunikaty od systemu operacyjnego, których przeciętny użytkownik nie jest w stanie w pełni wykorzystać.

Najczęstszym źródłem błędów będą błędy powstałe przy wprowadzaniu danych. Należy przyjąć zasadę, że wprowadzenie każdej informacji do maszyny musi być sprawdzone z punktu widzenia poprawności. Jednym z podstawowych celów projektantów dialogów powinno być zminimalizowanie prawdopodobieństwa wystąpienia błędu, po którym program kończy działanie.

Jako przykład przedstawimy rozwiązanie zagadnienia bezpiecznego wprowadzania danych dla programów pisanych w języku FORTRAN. Standardowe metody wprowadzania danych w FORTRAN-ie są wrażliwe na błędy popełniane przez użytkownika. Wystąpienie błędu w danych z reguły powoduje przerwanie wykonywanego programu z odpowiednim komunikatem. Użytkownik musi wtedy znaleźć błąd w danych, poprawić go i rozpocząć wykonywanie programu od początku. Przy dialogu człowieka z maszyną cyfrową za pomocą szybkiego urządzenia alfanumerycznego nie można do tego dopuścić. Dane należy czytać tak, żeby prawdopodobieństwo przerwania wykonywania programu, na skutek błędu w danych, było praktycznie równe zeru. Można to zrobić w ten sposób, że wczytuje się całe rekordy formatem A, a następnie je analizuje. Autor opracował taki program, w którym można wprowadzać dane liczbowe. Programem tym można wprowadzić do 50 liczb dowolnego typu. Formatem SOA1 wczytuje się cały rekord na tablicę o długości 80, a następnie znak po znaku analizuje się wprowadzone dane. Ponieważ do wprowadzania liczb nie potrzeba całego alfabetu języka FORTRAN wybrano 15-elementowy jego podzbiór. Do zbioru tego należy 10 cyfr oraz 5 znaków specjalnych, którymi są: + - . , .

Pozostałe symbole z alfabetu FORTRAN-u są pomijano. Każda liczba musi kończyć się przecinkiem. Znaczenie symboli + - . jest identyczne jak w FORTRAN-ie. Symbol . oznacza, że w następnym rekordzie będzie dalszy ciąg wprowadzanych liczb. Jeżeli w danych jest błąd, program informuje o tym użytkownika i prosi o podanie danych jeszcze raz.

## 6. Wnioski

Przedstawione w artykule zagadnienia związane z projektowaniem programów konwersacyjnych nie wyczerpują całości tego bardzo obszernego tematu. Celem pracy było przedstawienie niektórych problemów, które musi rozwiązać każdy projektant dialogu człowieka z maszyną.

Szybki rozwój urządzeń końcowych, za pomocą których można łatwo i szybko wymieniać informacje z maszyną zwiększa zapotrzebowanie na programy konwersacyjne. Praca konwersacyjna jest efektywniejsza i przy dobrze zaprojektowanym dialogu efektywniejsza od pracy w trybie tradycyjnym.

Należy jednak zdawać sobie sprawę z trudności, które należy pokonać przy uruchamianiu programów konwersacyjnych.

Najważniejszym problemem jest zapewnienie optymalnego czasu reakcji maszyny. Zagadnienia związane z budową dialogu oraz jego odpornością na błędy są ważne, jednak nie mają tak dużego wpływu na jakość programu konwersacyjnego. Wynika to z faktu, że na to, aby program rzeczywiście był programem konwersacyjnym, musi się odbywać w czasie jego działania konwersacja, tzn. wymiana informacji między użytkownikiem a maszyną rozwiązującą jego problem.

Zadania, dla których nie można zapewnić czasu reakcji w optymalnych granicach na danym sprzęcie, nie nadają się po prostu do rozwiązywania w trybie konwersacyjnym z powodu zbyt wolno działającego sprzętu lub z samej istoty rozwiązywanego problemu.

Do pracy w trybie konwersacyjnym najlepiej nadają się zadania, w których następuje częsta wymiana informacji między maszyną a człowiekiem oraz obecność użytkownika i jego decyzje mają istotny wpływ na otrzymane rozwiązanie. Do zadań takich można zaliczyć zadanie związane z procesem projektowania, szczególnie dla nietypowych procesów projektowania. Natomiast zadania, w których proces wprowadzania danych i proces właściwego rozwiązywania zadania są dość wyraźnie rozdzielone nie nadają się do rozwiązywania w trybie konwersacyjnym, chyba

ne nie nadają się do rozwiązywania w trybie konwersacyjnym, chyba że są to niewielkie zadania i główną zaletą konwersacji ma być szybkość uzyskania wyników.

Głównym celem powyższego artykułu było zwrócenie uwagi programistów na podstawowe problemy związane z uruchamianiem programów konwersacyjnych oraz pokazanie, że nie wszystkie zadania można rozwiązywać w trybie konwersacyjnym.

#### Literatura

- [1] MARTIN J.: Dialog człowieka z maszyną cyfrową. Warszawa: WNT 1976
- [2] MILLER R.B.: Response Time in Man-Computer Conversational Transactions. AFIPS Conference Proceedings, Washington, D.C., Thompson Book Company 1968
- [3] SAMET P.A.: Detailed Analysis of a Program - an Instructive Horror Story. SOFTWARE - PRACTICE AND EXPERIENCE 1975 Vol. 5 s. 211-213
- [4] JELENIEWSKI T., SIELICKI A.: Metodologia i komputerowe wspomaganie projektowania technicznego. Wyd. Politechniki Wrocławskiej, Wrocław 1975.
- [5] TURSKI W.M.: Propedeutyka informatyki. Warszawa: PWN 1975
- [6] POOCH U.W.: Computer Graphics and Man-Machine Interactions.



dr inż. Ryszard PAWLIK  
 Urszula ORŁOWSKA  
 Instytut Maszyn Matematycznych

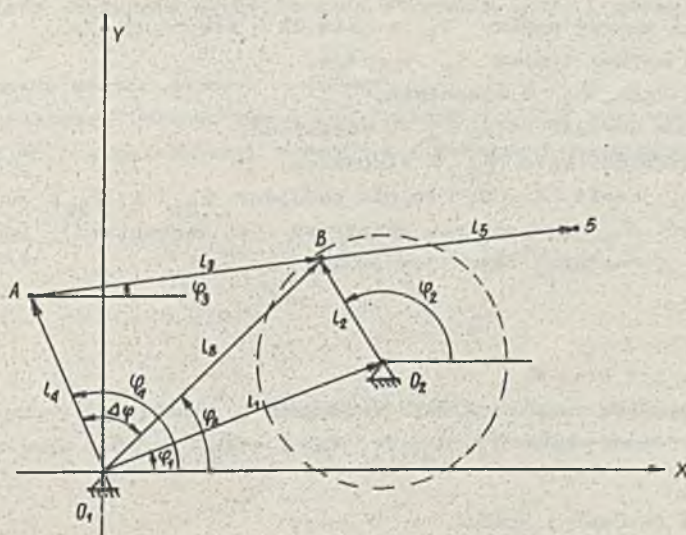
## Metoda projektowania dużych programów na przykładzie programu do projektowania pewnego mechanizmu płaskiego

### 1. Wstęp

Celem niniejszego artykułu jest przedstawienie procesu powstawania dużego programu komputerowego przeznaczonego do obliczeń inżynierskich. Prezentowaną metodę przedstawimy na przykładzie programu przeznaczonego do analizy kinematycznej mechanizmu płaskiego opartego na czworoboku przegubowym. Opisane będą kolejne etapy opracowywania programu bez szczegółowego opisu algorytmu. Podstawą przedstawionej metody jest podział programu na moduły, a następnie szczegółowy ich opis. Pokazano będą zalety takiego podejścia do tworzenia dużych programów przeznaczonych do obliczeń inżynierskich i im podobnych.

### 2. Sformułowanie zadania

Omawiany program ma służyć do analizy kinematycznej mechanizmu płaskiego, opartego na czworoboku przegubowym, stosowanego w łącznikach elektrycznych do realizacji ruchu prostoliniowego styków ruchomych. Schemat mechanizmu przedstawiono na rys. 1.



Rys. 1. Schemat mechanizmu opartego na czworoboku kinematycznym

Mechanizm ten zamocowany jest w punktach  $O_1$  i  $O_2$ . Elementy  $l_1$ ,  $l_2$ ,  $l_3$  i  $l_4$  nazywane są ogniwami (często ogniwo  $l_3$  nazywane jest także łącznikiem). W łącznikach elektrycznych ogniwo  $l_2$  zwykle jest członem napędzającym (nazywane jest wtedy korbą) i porusza się z prędkością kątową  $\omega_2$ .

Rozwiązując zadanie można przedstawić następująco: na podstawie danych długości ogniw mechanizmu  $l_1, l_2, l_3, l_4, l_5$ , kąta  $\varphi_1$ , prędkości kątowej osi  $O_2 - \omega_2$ , przyrostu kąta  $\Delta\varphi_2$  oraz przedziału  $\varphi_{2p}, \varphi_{2k}$  (rys. 1) określić:

- możliwość realizacji czworoboku przegubowego,

- typ mechanizmu,
- położenie punktów A, B, S dla następujących wartości kąta  $\varphi_2$ :  
 $\varphi_{2p}, \varphi_{2p} + \Delta\varphi_2, \varphi_{2p} + 2\Delta\varphi_2, \dots, \varphi_{2k}$
- wartość prędkości punktów A, S, wartość prędkości względnej punktu S, prędkość kątową ogniwa  $l_3$  w ruchu obrotowym względem punktu A oraz prędkość kątową ogniwa  $l_4$  w ruchu obrotowym względem osi  $O_1$ ,
- przyspieszeń punktów A i S,
- przyspieszenie kątowe ogniwa  $l_3$  w ruchu obrotowym względem punktu A oraz przyspieszenie kątowe ogniwa  $l_4$  w ruchu obrotowym względem osi  $O_1$ .

Wszystkie prędkości i przyspieszenia należy wyliczyć dla takich samych wartości kąta  $\varphi_2$ , jak dla położenia.

3. Budowa programu

Zakładamy, że program będzie miał budowę modułową. Dla określenia liczby modułów i ich zadań wprowadziliśmy opis programu przez określenie, co musimy dostarczyć do programu (będą to dane) oraz co chcemy otrzymać w wyniku działania programu (będą to wyniki).

• DANE

- $L_1, L_2, L_3, L_4, L_5$  - długości poszczególnych ogniw (rys. 1) wyrażono w cm,
- $\varphi_1$  - kąt, jaki tworzy ogniwo  $l_1$  z osią OX w stopniach,
- $\omega_2$  - prędkość kątowa ogniwa  $l_2$  w rad/s,
- $\Delta\varphi_2$  - przyrost kąta  $\varphi_2$  w stopniach,
- $\varphi_{2p}$  - początkowa wartość kąta  $\varphi_2$  w stopniach,
- $\varphi_{2k}$  - końcowa wartość kąta  $\varphi_2$  w stopniach,
- K - parametr, jeżeli  $K = 0$ , to nie podajemy  $\varphi_{2p}$  i  $\varphi_{2k}$ , a liczymy dla pełnego zakresu zmian kąta  $\varphi_2$  (zakres ten zależy od typu mechanizmu), jeżeli  $K \neq 0$ , to liczymy  $\varphi_{2p}$  do  $\varphi_{2k}$  i wartości te należy podać.

• WYNIKI

Położenie punktów A, B oraz S.  
 Położenia te wyraża się za pomocą współrzędnych X oraz Y.  
 Oznaczone są one odpowiednio  $X_A, X_B, Y_A, Y_B, X_S, Y_S$ .

Prędkości:

- $V_A$  - wartość prędkości punktu A w cm/s,
- $V_S$  - wartość prędkości punktu S w cm/s,
- $V_{SA}$  - wartość prędkości punktu S względem punktu A w cm/s,
- $\omega_3$  - prędkość kątowa ogniwa  $l_3$  w rad/s,
- $\omega_4$  - prędkość kątowa ogniwa  $l_4$  w rad/s.

Przyspieszenia:

- $A_{AN}$  - składowa normalna przyspieszenia punktu A w  $cm/s^2$ ,
- $A_{AT}$  - składowa styczna przyspieszenia punktu A w  $cm/s^2$ ,
- $A_{SAN}$  - składowa normalna przyspieszenia w ruchu punktu S wokół punktu A w  $cm/s^2$ ,
- $A_A$  - przyspieszenie punktu A w  $cm/s^2$ ,
- $A_{SAT}$  - składowa styczna przyspieszenia w ruchu punktu s wokół punktu A w  $cm/s^2$ ,

- $\Lambda_{SA}$  - przyspieszenie w ruchu punktu S względem punktu A w  $\text{cm/s}^2$ ,
- $\alpha_4$  - przyspieszenie kątowe ogniwa  $l_4$  w  $\text{rad/s}^2$ ,
- $\alpha_3$  - przyspieszenie kątowe ogniwa  $l_3$  w  $\text{rad/s}^2$ ,
- $\Lambda_S$  - przyspieszenie punktu S w  $\text{cm/s}^2$ ,

- M - zmienna określająca typ mechanizmu,
- M=0 - nie można zrealizować mechanizmu,
- M=1 - mechanizm korbowo-wahaczowy z napędem od korby,
- M=2 - mechanizm korbowo-wahaczowy z napędem od wahacza,
- M=3 - mechanizm dwukorbowy,
- M=4 - mechanizm dwuwahaczowy.

Mając określone zmienne zewnętrznie - dane i wyniki, należy na podstawie analizy przebiegu procesu obliczania tego typu mechanizmów zaprojektować rozbić program na moduły.

Z analizy tej (p. 1, 2 i dalsze) wynika następująca kolejność obliczeń:

- sprawdzenie możliwości realizacji mechanizmu oraz określenie jego typu gdy realizacja jest możliwa,
- obliczanie położenia skrajnych i zwrotnych mechanizmu,
- obliczanie położenia,
- obliczanie prędkości,
- obliczanie przyspieszeń.

Kolejność ta musi być ściśle zachowana, ponieważ np. położenia skrajne liczy się różnie, w zależności od typu mechanizmu, a przyspieszenie na podstawie położenia i prędkości.

Naturalne jest więc rozbić program na 5 modułów, z których każdy wykonuje jedno z powyższych zadań. W dalszej części należy ściśle wyspecyfikować poszczególne moduły, określić nazwę oraz jakie potrzebuje dane, tzn. określić dane dla danego modułu oraz co chcemy uzyskać w wyniku działania danego modułu. Ponadto należy dokładnie określić skąd otrzymuje dane i gdzie przekazuje wyniki.

Ustalono niżej wymienione nazwy poszczególnych modułów:

- TYP - sprawdza możliwość realizacji mechanizmu i ewentualnie określa jego typ,
- ZWROT - oblicza wartość kąta  $\varphi_2$  w położeniach skrajnych i zwrotnych mechanizmu,
- TRAP - oblicza położenia,
- PRĘDK - oblicza prędkości,
- PRZYSP - oblicza przyspieszenia.

#### 4. Specyfikacja modułów

W punkcie tym dokonano dokładnej specyfikacji wszystkich modułów. Opisu każdego modułu dokonano w dwóch punktach, tzn. określono zadanie modułu, zbiór danych i wyników oraz skąd otrzymuje dane i gdzie przekazuje wyniki.

Moduł TYP

DANE

$L_1, L_2, L_3, L_4$  - długość ogniw

WYNIKI

M - określa typ

Zmienna M zostaje przekazana do modułu ZWROT. Dane otrzymuje z programu głównego. Moduł ten na podstawie długości ogniw bada warunki realizacji mechanizmu. Jeżeli mechanizmu nie można zbudować z takich ogniw, zmienna M otrzymuje wartość zero, natomiast gdy można zbudować z podanych długości czworobok przegubowy - to określa jego typ.

Moduł ZWROT

DANE

$L_1, L_2, L_3, L_4, \Delta \varphi_2, K, \varphi_{2p}, \varphi_{2k}, M$

WYNIKI

$\varphi_2$

Dane otrzymuje się z programu głównego oraz z modułu TYP. Wyniki przekazuje do modułów: TRAP, PREDK oraz PRZYSP. Moduł ten oblicza wartości kąta  $\varphi_2$  dla położenia skrajnych i zwrotnych mechanizmu oraz dla kolejnych położenia mechanizmu. Liczba tych położenia zależy od  $\Delta \varphi_2$ .

Moduł TRAP

DANE

$L_1, L_2, L_3, L_4, L_5, \varphi_1, \varphi_2$

WYNIKI

$X_A, Y_A, X_B, Y_B, X_S, Y_S, \varphi_3, \varphi_4$

gdzie:  $\varphi_3$  - kąt, który tworzy ogniwo  $l_3$  z osią X,  
 $\varphi_4$  - kąt, który tworzy ogniwo  $l_4$  z osią X.

Moduł ten dane otrzymuje z programu głównego oraz z modułu ZWROT( $\varphi_2$ ), wyniki przekazuje do programu głównego oraz do modułów PREDK ( $\varphi_3, \varphi_4$ ), PRZYSP ( $\varphi_3, \varphi_4$ ).

Podprogram ten oblicza położenia ogniw mechanizmu napędu styków ruchomych łącznika oraz trajektorie punktów zamocowania styków.

Moduł PREDK

DANE

$L_1, L_2, L_3, L_4, L_5, \varphi_2, \varphi_3, \varphi_4$

WYNIKI

$\omega_3, \omega_4, v_A, v_{SA}, v_S$

Dane otrzymuje z programu głównego oraz modułów ZWROT oraz TRAP. Wyniki przekazuje do programu głównego oraz modułu PRZYSP( $\omega_3, \omega_4$ ).

Moduł ten oblicza prędkości ogniw mechanizmu napędu styków ruchomych łącznika oraz prędkość punktu zamocowania styków.

Moduł PRZYSP

DANE

$L_2, L_3, L_4, L_5, \omega_2, \varphi_2, \varphi_3, \varphi_4, \omega_3, \omega_4$

WYNIKI

$\alpha_3, \alpha_4, \Lambda_{AN}, \Lambda_{AL}, \Lambda_{SAN}, \Lambda_{SAT}, \Lambda_{SA}, \Lambda_S$

Dane otrzymuje z programu głównego z modułów ZWROT( $\varphi_2$ ), TRAP( $\varphi_3, \varphi_4$ ) oraz PREDK( $\omega_3, \omega_4$ ). Wyniki przekazuje do programu głównego.

Moduł ten oblicza przyspieszenia kątowe ogniw mechanizmu napędu styków oraz przyspieszenia punktu zamocowania styków.



## 5. Zakończenie

Na podstawie opisów poszczególnych modułów podanych w poprzednim punkcie przystąpiono do realizacji programu dla maszyny cyfrowej. Program realizowany był w języku FORTRAN IV. Każdy moduł reprezentowany był jako podprogram (SUBROUTINE). W języku FORTRAN są dwa sposoby komunikacji między modułami: za pomocą bloków COMMON oraz za pomocą parametrów formalnych.

Dla wybrania jednej z tych metod przeprowadzono analizę danych i wyników dla każdego modułu. Okazało się, że wiele danych występuje zawsze razem we wszystkich modułach. Dane te zgrupowano w bloki COMMON. Pozostałe dane oraz wyniki przekazywane były do odpowiednich modułów za pomocą parametrów formalnych. Po ustaleniu sposobów komunikacji między modułami oraz nazw i zawartości bloków COMMON uzupełniono specyfikację modułów o listę dostępnych dla każdego z nich bloków COMMON oraz listę parametrów formalnych.

Opisano podejście do projektowania programu ułatwiło znacznie realizację programu. W szczególności dokładne określenie zadań każdego modułu pozwoliło na niezależne uruchamianie ich, a więc skrócenie czasu realizacji. Ponadto program stał się bardzo przejrzysty, co znacznie ułatwia wszelkiego rodzaju zmiany i uzupełnienia w nim.

## Literatura

- [1] AU A., CIOK Z.: Aparaty elektryczne. Cz. I. Warszawa: WPW 1975.
- [2] AU A., MAKSYMUK J., POCHANICE Z.: Podstawy obliczeń aparatów elektroenergetycznych. Warszawa: WNT 1976.
- [3] ARTODOLEVSKIJ I.I.: Teorija mehanizmov dlja vosproizvedenija ploskich krivyh. Moskwa: Akademiya Nauk SSSR 1959.
- [4] ORŁOWSKA U.: Analiza kinematyczna mechanizmu płaskiego opartego na czworoboku kinematycznym. Warszawa: IMM 1977 Archiwum opracowań nr 31.
- [5] WIRTH N.: Systematic Programming. Englewood Cliffs: Prentice Hall 1973.



dr inż. Wojciech NOKRZYCKI  
Instytut Maszyn Matematycznych

## Algorytmy i ocena dokładności układowych generacji krzywych 2-stopnia na siatkach kwadratowych

### 1. Wstęp

#### Geneza problemu

W systemach komputerowych zachodzą ostatnio dość charakterystyczne zmiany polegające na zwiększeniu się roli rozwiązań układowych. Spowodowane jest to coraz częstszym pojawianiem się takich dziedzin zastosowań elektronicznej techniki obliczeniowej, które wymagają od systemów komputerowych zdolności do wykonywania obliczeń numerycznych i innych specjalizowanych funkcji w czasie z góry ograniczonym. Powoduje to konieczność osiągnięcia bardzo dużych szybkości przetwarzania, około kilkudziesięciu milionów operacji na sekundę i większych. Temu kierunkowi zmian sprzyja szybki postęp w technologii podzespołów elektronicznych (duża i bardzo duża skala integracji układów scalonych) prowadzący do dużego wzrostu szybkości i niezawodności działania układów oraz zasadniczego zmniejszenia się ich wymiarów.

Z tych i innych względów obserwuje się więc tendencje do przerzucania na układy elektroniczne wielu funkcji systemów realizowanych dotychczas za pomocą oprogramowania i do budowania wyspecjalizowanych superszybkich bloków (zespołów) funkcjonalnych.

W pracy niniejszej analizowane będą metody cyfrowej generacji krzywych przystosowane do sztywnej realizacji układowej. W szczególności analizowane będą tzw. całkowitoliczbowe metody generacji krzywych 2-stopnia. Metody te mogą znaleźć zastosowanie w konstrukcji generatorów krzywych stosowanych w grafice komputerowej (monitory graficzne, pisaki XY), w cyfrowym sterowaniu obrabiarkami (interpolatory krzywych), jak również w konstrukcji specjalnych jednostek decyzyjno-sterujących służących do sterowania na bieżąco, zwłaszcza procesami szybkozmiennymi.

#### Całkowitoliczbowe metody układowej aproksymacji krzywych

W wielu zastosowaniach operuje się przestrzenią dyskretną, w której zdefiniowane są jedynie punkty o współrzędnych całkowitoliczbowych, np. w mapach topograficznych, monitorach graficznych, pisakach XY, obrabiarkach sterowanych numerycznie i innych urządzeniach tego typu płaszczyzną obrazowania (ewentualnie przestrzenią sterowania) jest płaszczyzna (przestrzeń) dyskretna ze zdefiniowanymi węzłami całkowitoliczbowej siatki współrzędnych. Wykreślanie krzywych na takich płaszczyznach (lub droga ruchu narzędzia skrawającego w obrabiarkach sterowanych numerycznie) odbywa się przez zadanie ciągu punktów, które połączone kolejno odcinkami linii prostej aproksymują zadaną krzywą. Punkty ciągu aproksymującego leżą w węzłach siatki współrzędnych, czyli mają współrzędne całkowitoliczbowe.

W klasycznych metodach wielomianowej aproksymacji krzywych punkty aproksymujące krzywą wyznacza się za pomocą funkcji trygonometrycznych lub rozwinięć szeregowych. Są to operacje długotrwałe i złożone, wymagają bowiem stosowania zmiennoprzecinkowej arytmetyki i dają w wyniku współrzędne punktów będące liczbami rzeczywistymi. Dokładność, z którą punkty te są wyznaczone jest więc w pewnym sensie nie wykorzystana, gdyż punkty te umiejscawiane są następnie w węzłach całkowitoliczbowej siatki współrzędnych, co wymaga zaokrąglenia rzeczywistych wartości współrzędnych punktów do liczb całkowitych. Z drugiej jednak strony, gdyby dokładność ich wyliczenia była mniejsza, to w wyniku nieuwzględniania błędów w kolejnych krokach aproksymacji - kumulacja błędów doprowadziłaby do zupełnej "degeneracji" krzywych. Pojawili się więc problem opracowania takiego algorytmu, który wyznaczałby punkty aproksymujące z dokładnością tylko do liczb całkowitych i był równocześnie dostatecznie prosty i szybki oraz uwzględniał błąd powstały na każdym kroku aproksymacji, aby wyeliminować kumulację błędów.

Często stawia się dodatkowo warunek, aby algorytm był ukierunkowany na realizację układową, gdyż jest ona znacznie szybsza i prostsza, co oznacza, że prowadzić powinien do łatwych i prostych oraz szybko osiągalnych rozwiązań technicznych. Nakłada to na algorytm dodatkowe ograniczenia. Nie mogą więc być używane funkcje trudne w realizacji technicznej, wymagające złożonych rozwiązań układowych, wyznaczenie których z natury rzeczy wprowadza błędy systematyczne spowodowane zaokrągleniem liczb. A więc wzory generacyjne mogą zawierać jedynie proste operacje sumowania liczb całkowitych, mnożenie i dzielenie przez całkowitą potęgę dwójki, operacje sumy i iloczynu logicznego ew. operacje porównywania liczb stałooprzecinkowych lub inne podobnego typu.

W niektórych zastosowaniach mogą być istotne jeszcze i inne cechy, np. zwiększona dokładność i gładkość (w cyfrowym sterowaniu obrabiarkami oraz pisakami XY). Niektóre z wymienionych cech mogą być pierwszoplanowe (kosztem innych) np. dokładność i gładkość kosztem szybkości (w sterowaniu obrabiarkami i pisakami XY) lub szybkości generacji kosztem dokładności (w monitorach graficznych).

W związku z powyższym opracowano kilka metod tzw. całkowitoliczbowej aproksymacji nie obciążonych błędami systematycznymi [1], [2], [3], [5], [6], [11], [12], [15], [18], [19], [25], [28]. Cechą charakterystyczną tych metod jest aproksymowanie krzywych ciągiem punktów o współrzędnych całkowitoliczbowych. Z założenia więc punkty aproksymujące nie leżą dokładnie na żądanej krzywej. Odległość ta w zasadzie nie przekracza odległości między sąsiednimi całkowitoliczbowymi punktami płaszczyzny, czyli jest mniejsza od jednej jednostki. Poza tym odchylenie to powinno być uwzględnione podczas wyznaczania kolejnego punktu ciągu. Istnieje więc zależność między warunkami wyboru kolejnego punktu ciągu i odchyleniem poprzedniego punktu ciągu od żądanej krzywej.

Dla wyznaczenia całkowitoliczbowych punktów ciągu aproksymującego oblicza się bądź ich odległość od żądanej krzywej (wybierając punkty najmniej odległe) [1], [2], [11], [25], [28], bądź wartość modułu funkcji uwikłanej opisującej krzywą  $F(x,y)$  w tych punktach [5], [12], [19], [26], bądź wyznacza się całkowitoliczbowe pierwiastki równania  $F(x,y) = 0$  będące współrzędnymi punktów, które charakteryzują się obydwoma cechami, tj. są całkowitoliczbowe i leżą na konturze krzywej [6].

Istnieją również metody, które nie wymagają wyznaczenia żadnej z tych wielkości [3], [15], [18],[22].

Spośród znanych z literatury całkowitoliczbowych algorytmów aproksymacji na uwagę zasługują trzy poniższe algorytmy generujące krzywe 2-stopnia zadane w postaci uwikłanej  $F(x,y) = 0$ : algorytm Pittoway'a [19], algorytm Partridge'a [18] oraz algorytm Jordana, Lennona i Holma [12]. Pozostałe ze znanych algorytmów dotyczą generacji prostych bądź okręgów.

W algorytmie Pittoway'a dla wyznaczenia punktów aproksymujących krzywą oblicza się odległość od krzywej całkowitoliczbowych punktów płaszczyzny sąsiadujących z krzywą. Odległość ta jest funkcją współrzędnych bieżącego punktu ciągu aproksymującego oraz współczynników równania uwikłanego opisującego krzywą. I chociaż w algorytmie używane są jedynie proste operacje sumowania liczb stałooprzecinkowych, testowania znaków liczb, mnożenia i dzielenia przez 2 i 4 algorytm jest znacznie rozbudowany.

W algorytmie Partridge'a wykorzystuje się fakt, że funkcja  $F(x,y)$  ma po przeciwnych stronach konturu przeciwne znaki. Punkty ciągu wyznacza się w ten sposób, że oblicza się wartości funkcji w sąsiadujących z krzywą całkowitoliczbowych punktach, a na kolejny węzeł ciągu wybierany jest ten z tych punktów, w którym przyrost wartości funkcji ma znak przeciwny do znaku wartości funkcji w punkcie poprzednim ciągu. Algorytm ten jest znacznie prostszy od algorytmu poprzedniego, stąd też jest znacznie szybszy. Krzywe, które on generuje są jednak mało "gładkie", a ich kształt zależy od punktu początkowego i kierunku obiegu krzywej.

W algorytmie Jordana (i innych) przyjęto za miarę oddalenia punktu od konturu moduł wartości funkcji w tym punkcie. Na punkt ciągu wybiera się ten z całkowitoliczbowych punktów płaszczyzny, w którym moduł wartości funkcji jest najmniejszy. Ostatni algorytm generuje bardziej wygładzone krzywe i z większą dokładnością niż algorytm Partridge'a. Szybkość generacji jest jednak nieco mniejsza. W porównaniu z algorytmem Pittoway'a algorytm Jordana jest prostszy i szybszy, chociaż nieco mniej dokładny. Nie zapewnia on również jednoznaczności aproksymacji.

Istotnym zagadnieniem występującym przy wyborze algorytmu do realizacji układowej i konkretnego zastosowania jest sposób oceny właściwości poszczególnych algorytmów. Ponieważ przeprowadzenie

analitycznej oceny, w danym wypadku wydaje się być niemożliwe, najwłaściwsze będzie porównywanie algorytmów ze względu na ich cechy podstawowe (w danym zastosowaniu) takie, jak uniwersalność, dokładność, prostota realizacyjna, szybkość oraz liczba operacji głównych w jednym cyklu generacyjnym.

W dalszej części pracy analizowane będą całkowitoliczbowe aproksymacje krzywych drugiego stopnia, ukierunkowane na realizację układową, dla zastosowań zwłaszcza w grafice komputerowej (monitory graficzne, pisaki XY) oraz numerycznym sterowaniu obrabiarkami.

## 2. ALGORYTMY APROKSYMACJI KRZYWYCH 2-STOPNIA NA SIATKACH KWADRATOWYCH

Przedstawimy trzy nowe całkowitoliczbowe algorytmy aproksymacji krzywych 2-stopnia, w których używane są jedynie operacje sumowania i porównywania liczb stałoprzecinkowych oraz proste operacje na argumentach boolowskich: algorytm aproksymacji na siatce 4-spójnej, 8-spójnej i n-spójnej. Algorytmy te różnią się między sobą złożonością oraz dokładnością i szybkością generacji. Cechy te zależą bowiem w dość istotny sposób od rzędu spójności siatki, na której dokonywana jest aproksymacja krzywej.

Zadażmy na płaszczyźnie  $\mathbb{P}$ , w prostokątnym układzie XY krzywą 2-stopnia w postaci (1)

$$F(x,y) = ax^2 + 2bxy + cy^2 + dx + ey + f = 0 \quad (1)$$

Zadażmy również kwadratową siatkę dyskretną z modulem h za pomocą linii (2)

$$\begin{aligned} x &= h \cdot i \\ y &= (p/q)x + h \cdot i \end{aligned} \quad \text{gdzie } h, p, q, i \text{ - liczby całkowite, } q \neq 0 \quad (2)$$

Wprowadźmy pomocnicze definicje.

### • Definicja 1

Rzędem spójności kwadratowej siatki dyskretniej nazwiemy liczbę półprostych siatkowych wychodzących z węzła siatki.

### • Definicja 2

Bezpośrednio spójnymi węzłami siatki nazwiemy takie dwa węzły, które łączy linia siatkowa bezpośrednio (bez pośrednictwa innych węzłów).

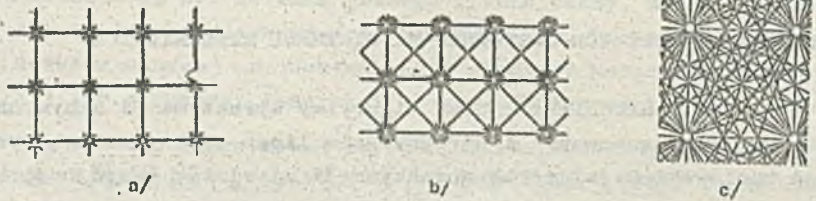
Dla różnych p i q otrzymujemy różne rodzaje siatek:

- p = 0 siatka 4-spójna (rys. 1a)  
x = h · i      y = h · i
- p/q = ±1. Siatka 8-spójna (rys. 1b)  
x = h · i      y = ±x + h · i
- p, q - całkowite  $\left[-\frac{n}{8} \leq p, q \leq \frac{n}{8}\right]$ . Siatka n-spójna (n = 2<sup>k</sup>, k-1, naturalna) rys. 1c  
x = h · i      y = (p/q)x + h · i

Formułujemy zagadnienie aproksymacji, polegające na podaniu algorytmu generującego łamaną złożoną z odcinków linii siatkowych (wyznaczonych przez bezpośrednio spójne węzły siatki) możliwie najbardziej przylegającą do aproksymowanej krzywej oraz na spełnieniu kryterium jednoznaczności aproksymacji, szybkości i prostoty w realizacji układowej algorytmu.

### Aproksymacje na siatce 4-spójnej

Obierzmy na zadanej przez (1) krzywej F(x,y) kierunek, określając w każdym punkcie na krzywej i dostatecznie bliskim jej sąsiedztwie wektory przemieszczeń wzdłuż stycznej do krzywej - za pomocą wzorów (4):



Rys. 1. Siatki kwadratowe:

- a) 4-spójne
- b) 8-spójna
- c) 32-spójna

$$+\vec{v} \triangleq (-F_y, F_x)$$

$$-\vec{v} \triangleq (F_y, -F_x),$$

(4)

gdzie  $F_x \triangleq \frac{\partial F}{\partial x}$ ,  $F_y \triangleq \frac{\partial F}{\partial y}$  są pochodnymi cząstkowymi funkcji  $F(x,y)$  pierwszego rzędu.

Taka definicja kierunku oznacza, że:

- dla kierunku  $+\vec{v}$  znak przemieszczenia  $\Delta x_i$  z punktu  $x_i, y_i$  do punktu  $x_{i+1}, y_{i+1}$  powinien być przeciwny znakowi  $F_y(x_i, y_i)$ , a znak  $\Delta y_i$  powinien być zgodny ze znakiem  $F_x(x_i, y_i)$ ,
- dla kierunku  $-\vec{v}$  - odwrotnie.

Określimy następnie kierunki ruchów wzdłuż linii siatki odpowiadające kierunkowi ruchów wzdłuż stycznej do krzywej za pomocą wzorów (5)

$$\Delta x = \begin{cases} +h, & \text{gdy jest spełnione } (F_y \geq 0) \wedge \bar{D}v(F_y < 0) = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases}$$

$$\Delta y = \begin{cases} +h, & \text{gdy jest spełnione } (F_x < 0) \wedge \bar{D}v(F_x \geq 0) = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases}$$

(5)

gdzie  $D$  jest zmienną boolowską równą:

$$D = 1 \text{ dla } +\vec{v} \text{ (dodatniego)}$$

$$D = 0 \text{ dla } -\vec{v} \text{ (ujemnego).}$$

Wzory (5), lokalizując nachylenie wektora kierunkowego stycznej z dokładnością do  $\frac{\pi}{2}$ , eliminują ruchy wzdłuż linii siatki najbliższe prostopadłym do krzywej, zostawiając możliwość wykonania ruchów  $\Delta x$  bądź  $\Delta y$  wzdłuż tych linii siatkowych, których kierunki są najbardziej zbliżone do kierunku wektora stycznej. Chcemy wykonać ruch do węzła najbliższego krzywej. Obliczanie odległości punktu od krzywej jest jednak operacją dość złożoną. Zauważmy, że moduł wartości funkcji  $F(x,y) = 0$  na konturze oraz  $F(x,y) \neq 0$  poza konturem, spełnia lokalnie warunki metryki i daje się łatwo i szybko wyznaczyć. Można go więc wybrać za miarę odległości węzłów od krzywej.

Rozpatrzmy rozwinięcie funkcji  $F(x,y)$  w szereg Taylora w postaci (6)

$$\begin{aligned} F(x+h, y) &= F(x, y) + hF_x + \frac{h^2}{2} F_{xx} \\ F(x-h, y) &= F(x, y) - hF_x + \frac{h^2}{2} F_{xx} \\ F(x, y+h) &= F(x, y) + hF_y + \frac{h^2}{2} F_{yy} \\ F(x, y-h) &= F(x, y) - hF_y + \frac{h^2}{2} F_{yy} \end{aligned} \tag{6}$$

( $F_{xx}$  i  $F_{yy}$  są pochodnymi cząstkowymi funkcji  $F(x, y)$  drugiego rzędu). Wynika z nich, że aby osiągnąć węzeł najbliższy krzywej należy wykonać taki ruch, który daje najmniejszą (bezwzględnie) wartość lewej strony w dowolnym z wyrażeń (6). Dochodzimy więc do schematu wyboru kolejnych punktów ciągu określonego relacjami (7):

- jeżeli  $|F(x + \Delta x, y)| < |F(x, y + \Delta y)|$   
to należy wykonać ruch wzdłuż osi  $x$ ;
- jeżeli  $|F(x, y + \Delta y)| < |F(x + \Delta x, y)|$   
to należy wykonać ruch wzdłuż osi  $y$ ,

gdzie  $\Delta x$  i  $\Delta y$  są określone przez relacje (5).

Pozostał do rozpatrzenia przypadek graniczny równości modułów

$$|F(x + \Delta x, y)| = |F(x, y + \Delta y)|,$$

który pozwala nałożyć na aproksymację dodatkowe kryterium, np.:

- wybieranie punktu leżącego po tej samej stronie krzywej, co i punkt poprzedni (wygładzanie łamanej),
- wybieranie punktu leżącego po przeciwnej stronie krzywej w stosunku do punktu poprzedniego (zmniejszenie pola między łamaną i krzywą),
- wybieranie punktu, który w sensie innej (niż metryka generacyjna  $|F(x, y)|$ ) metryki jest bliższy krzywej.

Łatwo wykazać, że dwa pierwsze kryteria nie zapewniają jednoznacznej aproksymacji krzywej, a kształt łamanej uzależniony będzie od punktu początkowego aproksymowanego segmentu i kierunku obrotu krzywej. Jednoznaczność aproksymacji uzyskamy wybierając, w przypadku granicznym, węzeł siatki leżący zawsze po tej samej stronie krzywej (tj. zawsze po stronie wypukłej bądź też zawsze po stronie wklęsłej). Łatwo wykazać, że gdy wybierzemy punkt leżący po stronie wypukłej krzywej, otrzymamy również mniejsze szczytowe odchylenie między krzywą i łamaną, gdyż np. dla okręgu oczywista jest nierówność:

$$|F(x, y; R-c)| < |F(x, y; R+0)|$$

gdzie  $c$  jest dowolną rzeczywistą liczbą dodatnią.

Algorytm aproksymacji na siatce 4-spójnej pokazano na rys. 2 (dla  $h=1$ ). Na rys. 3 przedstawiona jest aproksymacja hiperboli, wykonana za pomocą powyższego algorytmu (z modułem siatki  $h = 1$  [cm]) w wyniku jego komputerowej realizacji.

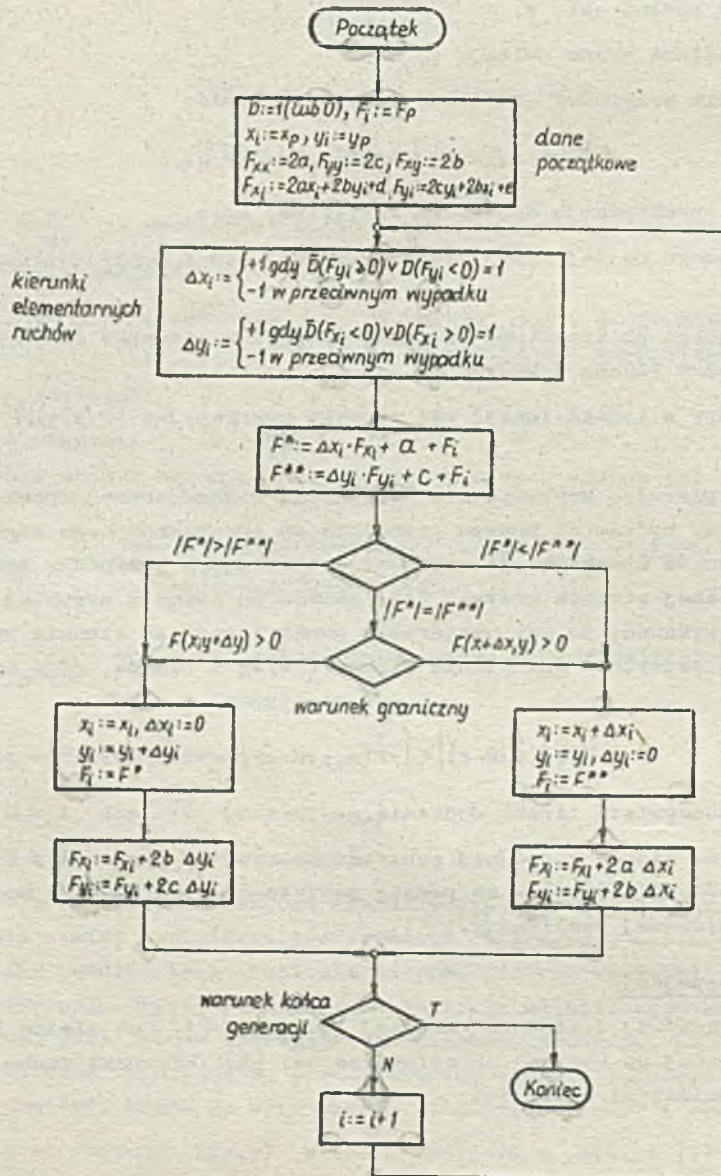
#### Aproksymacja na siatce 8-spójnej

Rozważmy aproksymację krzywej 2-stopnia (zadanej równaniem (1)) na siatce 8-spójnej. Wektory przemieszczeń wzdłuż stycznej do krzywej określimy wzorami (4). Kierunki ruchu wzdłuż linii siatki określimy natomiast poniższymi relacjami:

$$\Delta x = \begin{cases} +h, & \text{gd\u017cy jest sp\u00f3lnione } (F_y > 0) \wedge \bar{D}v(F_y < 0) \\ & \wedge Dv(F_y = 0) \wedge (F_x < 0) = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases} \quad (8)$$

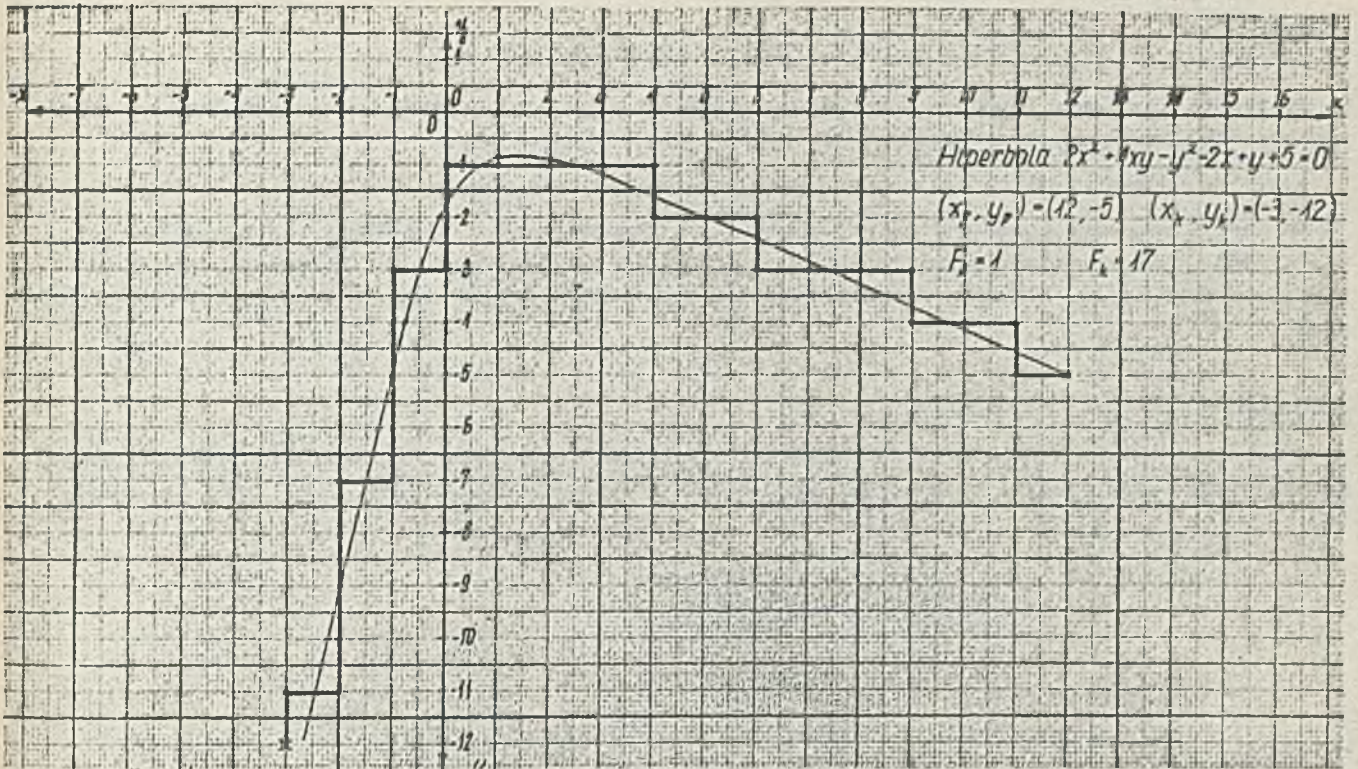
$$\Delta y = \begin{cases} +h, & \text{gd\u017cy jest sp\u00f3lnione } (F_x < 0) \wedge \bar{D}v(F_x > 0) \\ & \wedge Dv(F_x = 0) \wedge (F_y < 0) = 1 \\ -h & \text{w przeciwnym wypadku} \end{cases}$$

kt\u00f3re r\u00f3\u017ani\u0105 si\u0119 od odpowiednich relacji (5) z algorytmu poprzedniego o cz\u0142ony warunkuj\u0105co wyb\u00f3r dopuszczalnego kierunku ruchu, gdy jedna z pochodnych cz\u0105stkowych rz\u0119du pierwszego jest r\u00f3wna zeru. Cz\u0142ony te maj\u0105 istotne znaczenie zw\u0142aszcza w wypadku aproksymacji krzywych o du\u017cych krzywiznach. Relacje (8) eliminuj\u0105 5 z 8 mo\u017cliwych na danej siatce kierunk\u00f3w ruchu. Z tych trzech ruch\u00f3w powinni\u015bmy wybra\u0107 ten, kt\u00f3ry doprowadzi do w\u0119z\u0142a najbli\u017cszego krzywej.



Rys. 2. Algorytm aproksymacji na siatce 4-spoj\u0144nej





Rys. 3. Aproxymacja hiperboli na siatce 4-spójnej z modulem siatki  $h = 1$  cm

Gdy za miarę odległości punktu od krzywej wybierzemy moduł wartości funkcji  $F(x,y)$  to powinniśmy wyznaczać go w trzech punktach i wybierać punkty z modulem najmniejszym. Konieczność wyznaczenia modułu wartości funkcji aż w trzech węzłach wynika z faktu, że relacje (8) lokalizują nachylenie kierunkowego wektora stycznej z dokładnością również do  $\frac{\pi}{2}$ . Wyeliminowanie konieczności wyliczania modułu wartości funkcji w jednym z punktów wymaga natomiast uściślenia kąta zawierania się wektora kierunkowego stycznej z dokładnością do  $\frac{\pi}{4}$ . Zauważmy, że możemy to uzyskać rozważając relację  $|F_x| \leq |F_y|$  między modułami pochodnych cząstkowych w bieżącym węźle, które wyliczane są w celu wyznaczenia wartości funkcji  $F(x,y)$  w tym punkcie. Z porównania tego wyniku, że:

- jeżeli  $|F_x| > |F_y|$  to mniejszy z kątów między styczną do krzywej i osią  $x$  jest mniejszy od  $\frac{\pi}{4}$ ,
- jeżeli  $|F_x| < |F_y|$  to mniejszy z kątów między styczną do krzywej i osią  $y$  jest mniejszy od  $\frac{\pi}{4}$ .

Uwzględniając powyższe możemy napisać relacje (9) eliminujące konieczność wyznaczenia modułu wartości funkcji w jednym z trzech punktów, kosztem prostszych i szybszych operacji logicznych na argumentach boolowskich.

Uwzględniając w rozważaniach kierunek obiegu krzywej  $D$  odpowiednie relacje przyjmą postać:

- jeżeli jest spełnione

$$(|F_x| > |F_y|) \vee D \wedge (F_x = -F_y) \vee \bar{D} \wedge (F_x = F_y) = 1$$

to należy analizować nierówność

$$|F(x + \Delta x, y + \Delta y)| \geq |F(x, y + \Delta y)|, \quad (9)$$

- jeżeli jest spełnione

$$(|F_x| < |F_y|) \vee D \wedge (F_x = F_y) \vee \bar{D} \wedge (F_x = -F_y) = 1$$

to należy analizować nierówność

$$|F(x + \Delta x, y + \Delta y)| \geq |F(x + \Delta x, y)|$$

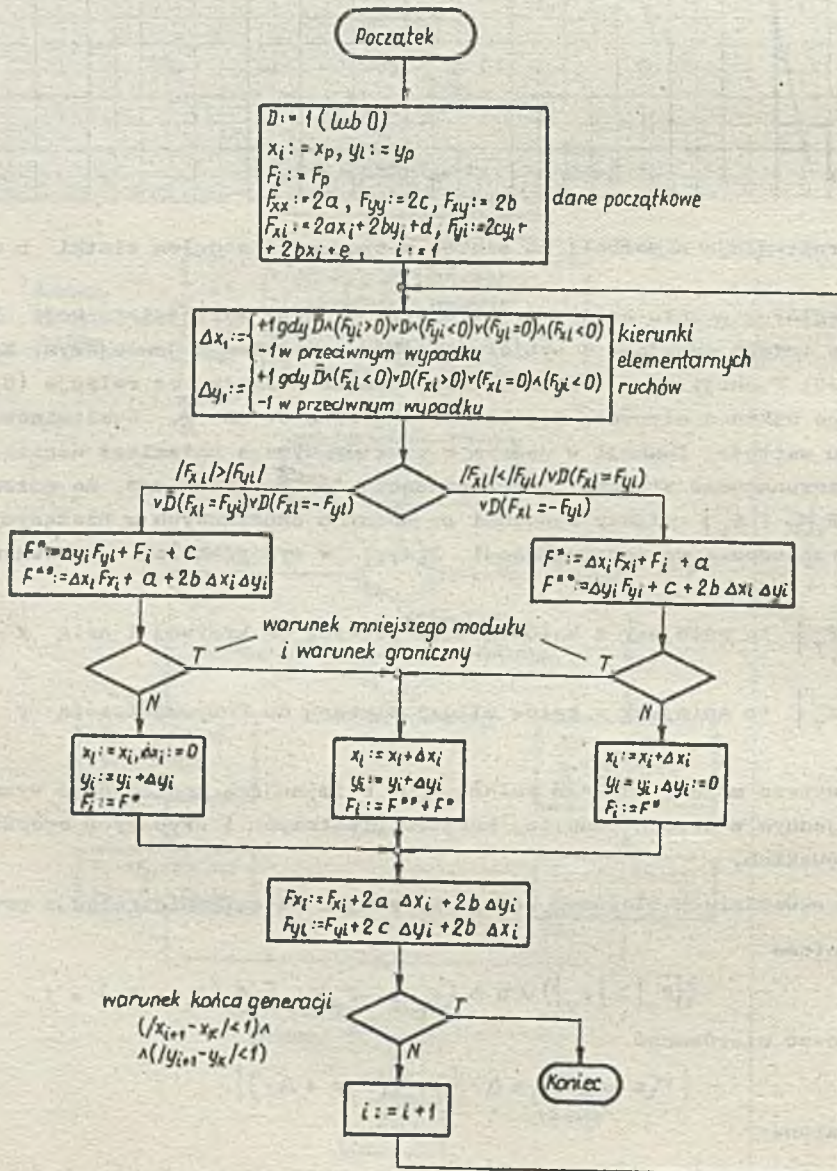
Przypadek graniczny równości modułów funkcji rozstrzygamy na korzyść punktu leżącego po stronie wypukłej krzywej, z identycznym, jak poprzednio uzasadnieniem.

Algorytm aproksymacji na siatce 8-spójnej pokazano na rys. 4 (dla  $h = 1$ ).

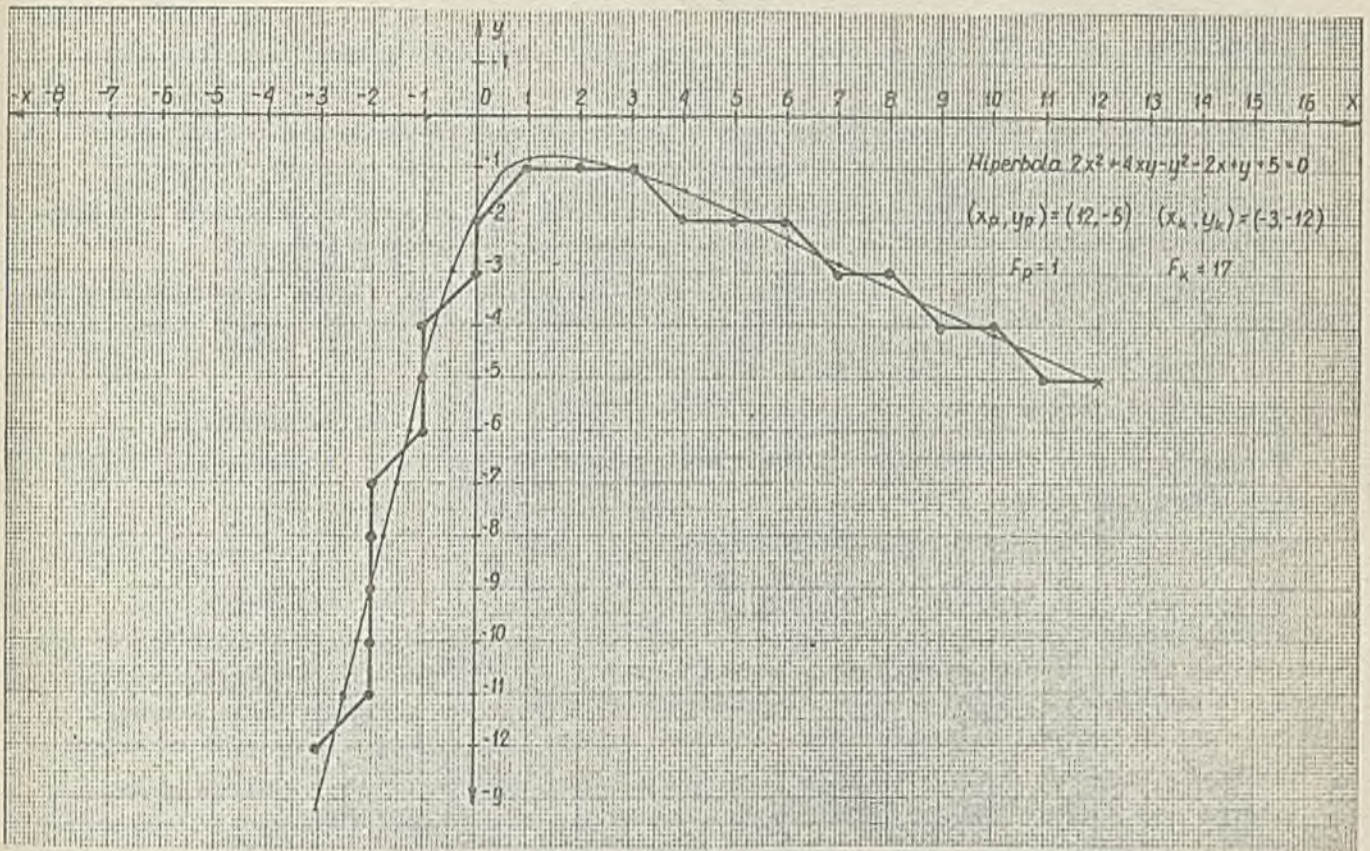
Na rys. 5 pokazano aproksymację wspomnianej już hiperboli na siatce 8-spójnej z modułem  $h = 1$  [cm]. Skonstruowany algorytm aproksymacji na siatce 8-spójnej jest bardziej złożony od algorytmu poprzedniego. Stosowane są w nim jednak tego samego typu operacje, co w algorytmie poprzednim.

Aproksymacja na siatce n-spójnej

W aproksymacji na siatkach 4- i 8-spójnych odległość między kolejnymi węzłami aproksymującej krzywą łańcucha węzłów siatki była z góry ograniczona i wynosiła odpowiednio  $h$  i  $h\sqrt{2}$  ( $h$  - moduł



Rys. 4. Algorytm aproksymacji na siatce 8-spójnej



Rys. 5. Aproxymacja hiperboli na siatce 8-spójnej z modulem  $h = 1$  [cm]

siatki). Stąd też nachylenie odcinków łamanej mogło przyjmować odpowiednio dwie i trzy różne wartości. W aproksymacji na siatce  $n$ -spójnej odległości tej nie ogranicza się z góry (gdyż nie ogranicza się liczby  $n$ ), a tym samym kąty nachylenia odcinków łamanej mogą przyjmować przeliczalną liczbę różnych wartości. Mogą więc ściślej przylegać do krzywej.

Konstruując algorytm aproksymacji na siatce  $n$ -spójnej będziemy dążyć do tego, aby kolejny węzeł ciągu był możliwie najbliższy krzywej i tak, aby każdy odcinek łamanej wyznaczony przez dwa kolejno węzły ciągu "najściślej" przylegał do krzywej oraz był możliwie najdłuższy.

Zakładamy, że aproksymujemy krzywą 2-stopnia w postaci ogólnej, zadaną wzorem (1), w którym współczynniki  $a, b, c, d, e$  i  $f$  są tak dobrane, że  $F(x, y) > 0$  po wypukłej stronie krzywej. Zakładamy również, że wybrano leżący w pobliżu krzywej punkt początkowy  $x_p, y_p$  aproksymowanego odcinka krzywej. Kierunek wzdłuż stycznej do krzywej definiujemy relacjami (4), a relacjami (8) kierunki odpowiednich przemieszczeń wzdłuż głównych linii siatki najbliższe kierunkowi ruchów wzdłuż stycznej do krzywej. Chcemy wyznaczyć następny punkt ciągu. Postępujemy w sposób następujący:

- wykonujemy elementarny ruch z punktu  $x_p, y_p$  do węzła sąsiedniego wzdłuż przekątnej siatki 8-spójnej wyznaczając w węźle  $x_{p+1}, y_{p+1}$  wartość funkcji  $F(x, y)$  i pochodnych 1-stopnia;
- poczynając od węzła  $x_{p+1}, y_{p+1}$  wykonywać będziemy elementarne ruchy wzdłuż krzywej, dokładniej: wzdłuż tej ze współrzędnych  $x$  lub  $y$ , która z wektorem kierunkowym stycznej wystawionym w punkcie  $x_{p+1}, y_{p+1}$  tworzy mniejszy kąt obserwując, czy nie oddalamy się od krzywej; w każdym z kolejnych węzłów obliczać będziemy wartość funkcji i jej pochodnych cząstkowych 1-stopnia; operację tę powtarzać będziemy dopóty, dopóki kolejny elementarny ruch nie przetnie krzywej;
- z dwóch sąsiednich, leżących na przeciwnych stronach krzywej węzłów siatki wybierzemy na węzeł ciągu ten z węzłów siatki, który jest bliższy krzywej w sensie metryki generacyjnej;

- stwierdzając odbioganie od krzywej zmienimy kierunek elementarnych ruchów o kąt  $\pm \frac{\pi}{2}$  (tj. z kierunku osi  $x$  na  $y$ ) lub odwrotnie tak, aby w następnym ruchu osiągnąć przecięcie się z krzywą.

W postępowaniu tym dokładnego sprecyzowania wymagają operacje elementarnych ruchów wzdłuż stycznej do krzywej i zmiana ich kierunku oraz oddalanie się i przecięcie krzywej.

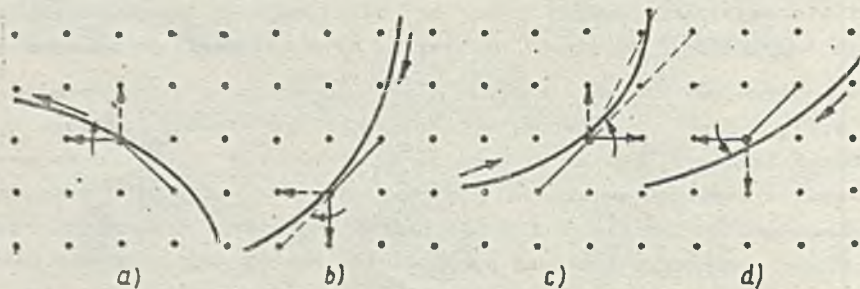
Kierunek elementarnego ruchu wzdłuż krzywej zależy oczywiście od nachylenia wektora kierunkowego stycznej oraz położenia punktu  $x_{p+1}, y_{p+1}$  (wewnątrz czy na zewnątrz krzywej).

Określimy go wychodząc ze związku między pochodną  $f(x,y)$  funkcji  $f(x,y)$  a pochodnymi cząstkowymi  $F_x$  i  $F_y$  formy uwiklanej  $F(x,y)$  tejże funkcji, tj.:  $f'(x,y) = -\frac{F_x}{F_y}$ . Mianowicie:

- ruch odbywa się zgodnie z  $\Delta x$ , gdy
  - a)  $x_{p+1}, y_{p+1}$  leży po wewnętrznej stronie krzywej, a znaki  $F_x$  i  $\Delta x$  są zgodne,
  - b)  $x_{p+1}, y_{p+1}$  leży po zewnętrznej stronie krzywej, a znaki  $F_y$  i  $\Delta y$  są zgodne,
- ruch odbywa się zgodnie z  $\Delta y$ , gdy
  - a)  $x_{p+1}, y_{p+1}$  leży po wewnętrznej stronie krzywej, a znaki  $F_y$  i  $\Delta y$  są zgodne,
  - b)  $x_{p+1}, y_{p+1}$  leży po zewnętrznej stronie krzywej, a znaki  $F_x$  i  $\Delta x$  są zgodne.

Zmiana kierunków elementarnych ruchów powinna nastąpić, gdy nie zachodzi zbliżanie się do krzywej. Wyraża się to niemalejącą wartością modułu wartości  $F(x,y)$  w kolejno osiągniętych węzłach. Zmiana ta uzależniona jest od położenia sąsiedniego węzła (wewnątrz czy na zewnątrz krzywej) i kierunku obiegu krzywej. Modyfikacje (przedstawione na rys. 6.), wyrażamy poniższymi relacjami:

- wykonujemy obrót o  $\frac{\pi}{2}$ , gdy bieżący punkt znajduje się wewnątrz krzywej a kierunek obiegu krzywej jest dodatni, lub gdy punkt znajduje się na zewnątrz krzywej a kierunek obiegu krzywej jest ujemny,
- wykonujemy obrót o  $+\frac{\pi}{2}$ , gdy bieżący punkt znajduje się na zewnątrz krzywej, a kierunek obiegu krzywej jest dodatni lub gdy punkt znajduje się wewnątrz krzywej, a kierunek obiegu krzywej jest ujemny.

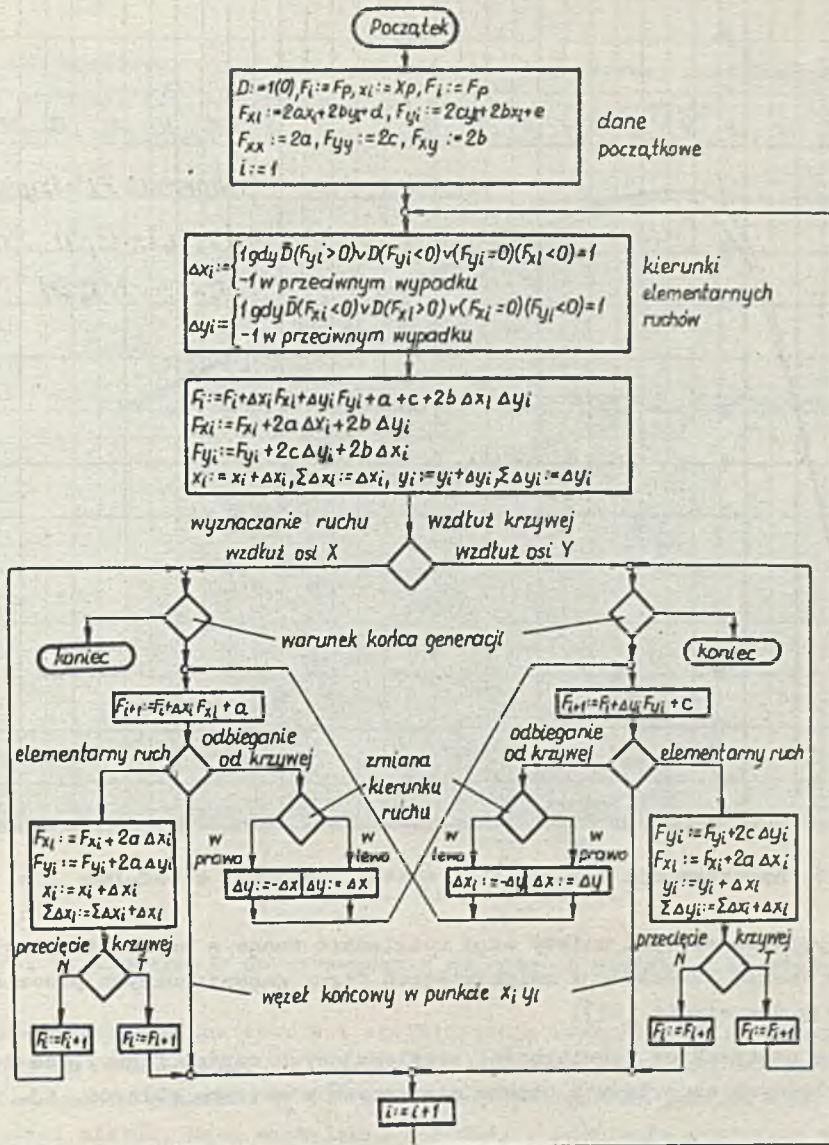


Rys. 6. Zasada modyfikacji kierunków elementarnych ruchów wzdłuż krzywej

Przecięcie krzywej powoduje zmianę znaku wartości funkcji  $F(x,y)$ , stąd też wymaga jedynie sprawdzania znaków jej wartości w kolejno osiągniętych węzłach.

Kolejny odcinek łamanej wyznaczamy przyjmując za jego punkt początkowy punkt końcowy poprzedniego odcinka, natomiast jego punkt końcowy wyznaczamy sposobem wyżej opisanym.

Algorytm aproksymacji na siatce  $n$ -spójnej dla  $h = 1$  pokazano na rys. 7. W porównaniu z algorytmami poprzednimi algorytm ten jest znacznie bardziej złożony logicznie. Używane są w nim jednak operacje tego samego typu co w algorytmach poprzednich. Mimo jego ukierunkowania na realizację układową można z powodzeniem stosować go w realizacji programowej. W tym zastosowaniu algorytm ma następujące cechy dodatnie:



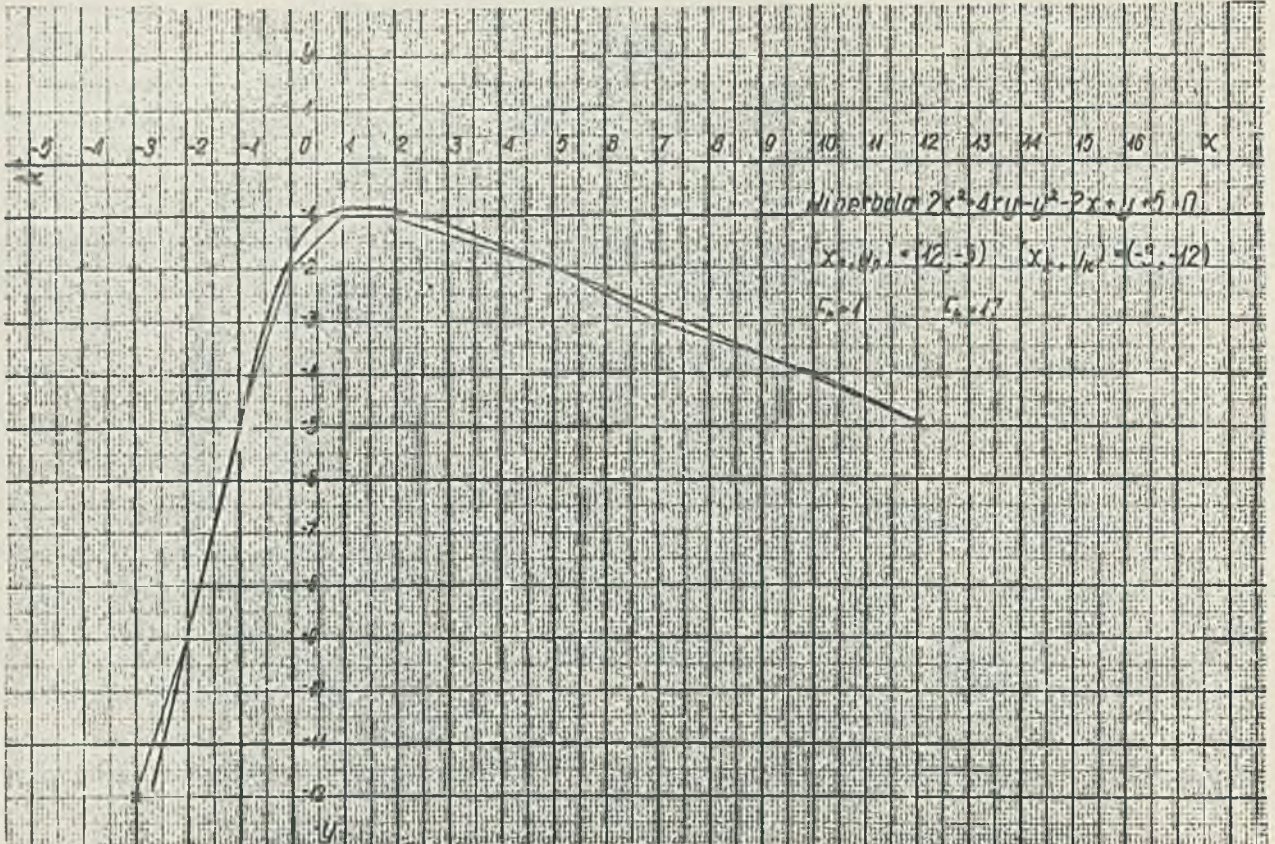
Rys. 7. Algorytm aproksymacji na siatce n-spójnej (n-przeliczalne)

- brak zjawiska kumulacji błędów (deformacji krzywej),
- duża szybkość generacji,
- mała liczba punktów opisujących krzywą (niewielkie wymagania na pojemność pamięci operacyjnej),
- regulowana (wielkością modułu h siatki) dokładność aproksymacji.

Na rys. 8 pokazano aproksymację hiperboli za pomocą omówionego wyżej algorytmu na siatce z modulem  $h = 1$  [cm].

### 3. OCENA JAKOŚCI APROKSYMACJI

Wybór miary jakości aproksymacji powinien być adekwatny do problemu, w ramach którego dokonuje się aproksymacji. Ukierunkowanie algorytmów do zastosowań przede wszystkim w grafice komputerowej wymaga uwzględnienia (w ocenie aproksymacji) cech percepcji wzrokowej człowieka, a zwłaszcza mechanizmów oceny kształtów i zniekształceń figur geometrycznych. Aproksymacja ukierunkowana na wymienione zastosowanie powinna być aproksymacją kształtów figur geometrycznych, tj. aproksymacją



Rys. 8. Aproxymacja hiperboli na siatce  $\pi$ -spójnej z modulem  $h = 1$  [cm]

w zbiorach. Przy wyborze metryki należy więc uwzględnić znane z psychofizjologii percepcji wzrokowej fakty, że w ocenie kształtów i zniekształceń figur geometrycznych przez system wzrokowy człowieka występują dwa stadia [27]:

- ocena lokalnych odchyłek od regularności przylegających części figury, co identyfikuje się z " pomiarem " względnych szczytowych błędów mierzonych w metryce zbiorów, tj. w metryce euklidesowskiej,
- ocena figury jako całości (globalna ocena) ukierunkowana rozmiarami, kształtami i zniekształceniami tak samej figury, jak i figur z nią sąsiadujących.

Ukierunkowanie algorytmów do numerycznego sterowania obrabiarkami wymaga natomiast zastosowania przede wszystkim miary szczytowej (lokalnej). Aproxymacja do tych zastosowań powinna być również aproxymacją kształtów geometrycznych obrabianego detalu, jednak istotno są w zasadzie jedynie odchyłki szczytowe.

Z powyższych względów jakość aproxymacji oceniana będzie w metryce szczytowej (euklidesowskiej) i w metryce polowej. Oszacowana więc zostanie szczytowa odległość między węzłami łamanej i krzywą oraz średni moduł pola zawartego między łamaną i krzywą odniesiony do jednostkowego odcinka krzywej.

#### Szczytowe błędy aproxymacji

#### Oszacowanie bezwzględnych szczytowych błędów aproxymacji

Omówiono w poprzednim punkcie algorytmy generacyjne wybierają na punkty aproxymującego krzywą ciągu to węzły siatki, w których funkcja uwikłana  $f(x, y)$  przyjmowała wartość najmniejszą bezwzględnie. Spośród węzłów aproxymującego krzywą ciągu najbardziej oddalonymi od krzywej będą

oczywiście przeguby łamanej, tj. te punkty ciągła, w których łamana zmienia kierunek. Wyznaczenie szczytowych błędów aproksymacji sprowadza się zatem do wyznaczenia odległości przegubów łamanej od krzywej.

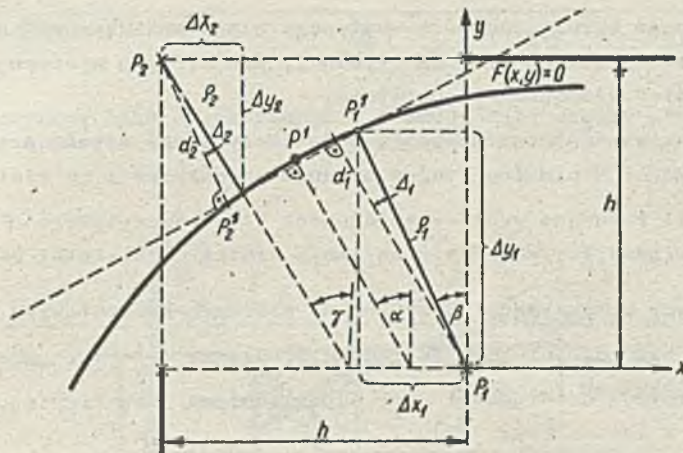
Spośród wszystkich możliwych wzajemnych położeni siatki i krzywej maksimum odchylenia przegubów od krzywej wystąpi wówczas, gdy "kandydująco" na kolejny węzeł łamanej węzły siatki położone będą po przeciwnych stronach krzywej, a moduły wartości funkcji w tych punktach będą jednakowe (rys. 9), zgodnie z równością (10).

$$|F(P_1)| = |F(P_2)| \quad (10)$$

lub równoważną jej równością

$$F(P_1) + F(P_2) = 0 \quad (11)$$

gdyż znaki wartości funkcji po przeciwnych stronach konturu krzywej są przeciwne.



Rys. 9. Ilustracja do wyznaczenia odległości węzłów od krzywej

Dla przypadku przedstawionego na rys. 9 i spełniającego równość (11) wyznaczmy odległości  $\rho_1$  i  $\rho_2$  węzłów  $P_1$  i  $P_2$  od krzywej  $F(x, y) = 0$ . Podstawiając do (11) rozwinięcia  $F(x, y)$  w szereg Taylora w otoczeniu punktów  $P_1$  i  $P_2$  oraz wykorzystując związek na sumę  $\rho_1 + \rho_2$  będący funkcją rzędu spójności siatki, kąta nachylenia stycznej i promienia krzywizny krzywej otrzymamy, przy założonym uproszczeniu (uczynionym dla  $R \gg h$ ):

$$\gamma \approx \alpha \approx \beta$$

wyrażenie ma ekstremalną wartość szczytowego bezwzględnego błędu aproksymacji w postaci:

$$\rho_1 = \frac{R}{\left(\frac{R}{P}\right)^{2/3} + s\left(\frac{Rb}{a^2}\right)^{2/3} - 1} + \frac{h\sqrt{2}}{2} \cos\left(\frac{\pi}{4} - \theta\right) - \sqrt{\left[\left(\frac{R}{P}\right)^{2/3} + s\left(\frac{Rb}{a^2}\right)^{2/3} - 1\right]^2 - \frac{h^2}{2} \cos^2\left(\frac{\pi}{4} - \theta\right)} \quad (12)$$

$$0 \leq \theta \leq \frac{\pi}{2}$$

dla aproksymacji na siatce 4-spójnoej oraz wyrażenie postaci

$$\rho_1 = \frac{R}{\left(\frac{R}{P}\right)^{2/3} + s\left(\frac{Rb}{a^2}\right)^{2/3} - 1} + \frac{h \cos \theta}{2} \sqrt{\left[\left(\frac{R}{P}\right)^{2/3} + s\left(\frac{Rb}{a^2}\right)^{2/3} - 1\right]^2 - \frac{h^2 \sin^2 \theta}{4}} \quad (13)$$

$$0 \leq \theta \leq \frac{\pi}{4}$$

dla aproksymacji na siatce 8-spójnej:  $\theta$  jest kątem nachylenia stycznej do krzywej w stosunku do głównej linii siatki,  $R$  - promień krzywizny w bieżącym punkcie krzywej,  $a$  i  $b$  - półosie krzywej,  $p$  - parametr krzywej;  $S$  przyjmuje wartości:

$$S = \begin{cases} 1 & \text{dla prostej, okręgu i elipsy} \\ 0 & \text{dla paraboli} \\ -1 & \text{dla hiperboli} \end{cases}$$

Bezwzględne szczytowe błędy aproksymacji krzywych 2-stopnia na siatkach kwadratowych są więc odwrotnie proporcjonalne do promienia lokalnej krzywizny i do rzędu spójności siatki oraz zależą od kąta między styczną do krzywej i bliższą z głównych linii siatki. Dla tych samych dopuszczalnych (z założenia) maksymalnych krzywizn krzywych szczytowe bezwzględne wartości błędów aproksymacji elipsy, paraboli i hiperboli nie przekraczają szczytowych wartości błędów dla okręgu.

Porównując kresy górne szczytowego błędów aproksymacji na siatkach 4-spójnej i 8-spójnej można stwierdzić, że:

- maksimum kresu górnego bezwzględnego szczytowego błędów aproksymacji na siatce 8-spójnej występuje dla tych wartości kąta nachylenia stycznej, dla których występuje minimum tegoż błędów w aproksymacji na siatce 4-spójnej i odwrotnie;
- wartość maksymalna kresu górnego bezwzględnego szczytowego błędów aproksymacji na siatce 8-spójnej równa jest wartości minimalnej tegoż błędów w aproksymacji na siatce 4-spójnej.

Na rys. 10 i rys. 11 pokazano wykresy zależności błędów szczytowych  $\rho_1$  w funkcji promienia krzywizny i kąta  $\alpha$  między styczną i główną linią siatki (dla siatki 4- i 8-spójnej).

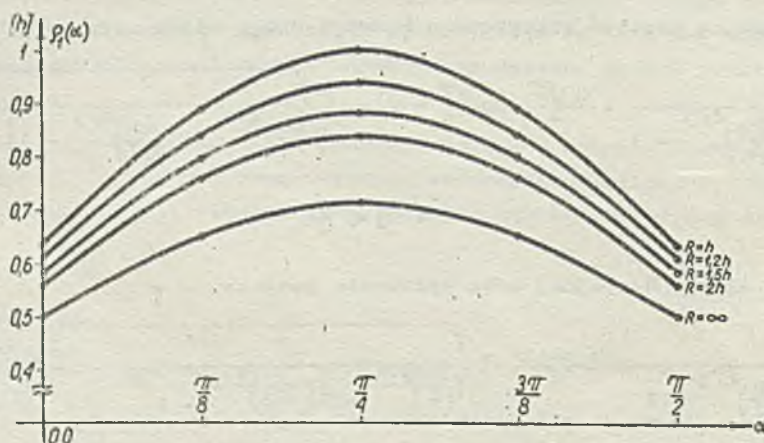
Względne szczytowe błędy aproksymacji krzywych na siatkach kwadratowych

Względny szczytowy błąd definiujemy za pomocą wyrażenia  $\frac{\rho_1}{\tau_1}$  tj. jako stosunek odległości  $\rho_1$  przegubu łamanej od krzywej do długości  $\tau_1$  odpowiedniego łuku krzywej, wyznaczonego przez punkty przecięcia krzywej z łamaną (rys. 12).

W celu wyznaczenia błędów względnego dla poszczególnych krzywych 2-stopnia dokonujemy u p r o s z c z e n i a polegającego na zastąpieniu łuków  $\tau_1$  krzywej cięciwami  $l_1$ , a w miejsce odległości  $\rho_1$  przyjmujemy odległość  $d_1$  przegubu łamanej od cięciwy. Obliczamy więc w miejsce wyrażenia  $\frac{\rho_1}{\tau_1}$  wyrażenie  $\frac{d_1}{l_1}$  (rys. 12).

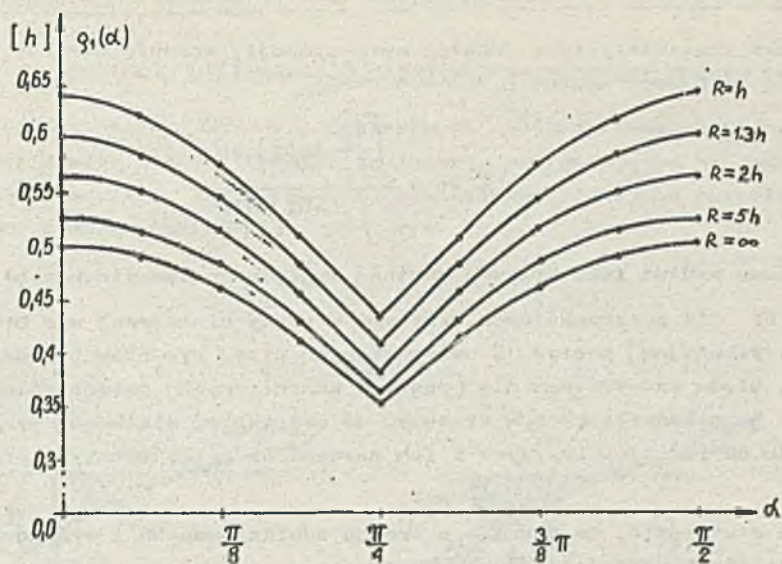
Nietrudno wykazać, że przy tego typu uproszczeniu szczytowy względny błąd aproksymacji krzywych 2-stopnia (również krzywych wyższych stopni) wyraża się wzorem:

$$\frac{d_1}{l_1} = \sin \theta \cos \theta, \quad 0 \leq \theta \leq \frac{\pi}{2}, \quad (14)$$

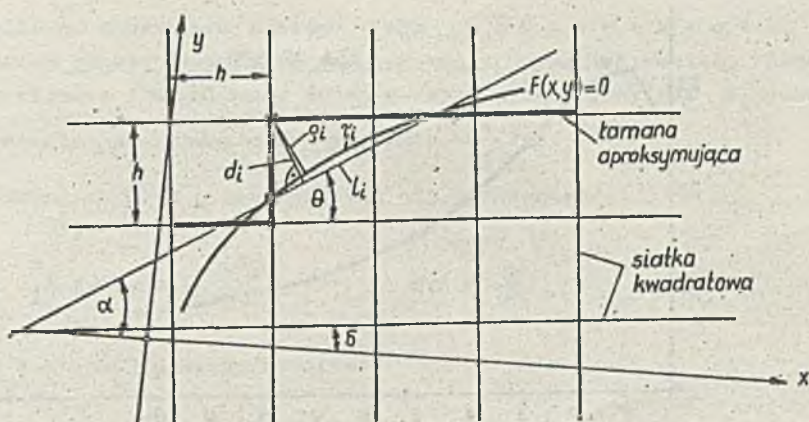


Rys. 10. Szczytowy błąd aproksymacji w funkcji kąta między styczną do krzywej i linią siatki 4-spójnej (promień krzywizny  $R$  - parametr)





Rys. 11. Szczytowy błąd aproksymacji w funkcji kąta między styczną do krzywej i główną linią siatki 8-spójnej (promień krzywizny  $R$  - parametr)



Rys. 12. Określenie względnego szczytowego błędu aproksymacji

(osiągając wartość maksymalną  $\max\left(\frac{d_i}{l_i}\right) = \frac{1}{2}$  dla  $\theta = \frac{\pi}{4}$ ) w wypadku aproksymacji na siatce 4-spójnej oraz wzorem

$$\frac{d_i}{l_i} = \sin \theta \cos \theta - \sin^2 \theta, \quad 0 \leq \theta \leq \frac{\pi}{4} \quad (15)$$

$$\left(\max\left(\frac{d_i}{l_i}\right)\right) = \frac{\sqrt{3+2\sqrt{2}} - 1}{4 + 2\sqrt{2}} \approx 0.207 \quad \text{dla} \quad \theta = \arctg(\sqrt{2} - 1)$$

w wypadku aproksymacji na siatce 8-spójnej ( $\theta$  jest kątem nachylenia cięciwy w stosunku do głównej linii siatki).

Wyrażenie  $\frac{d_i}{l_i}$  jest więc funkcją rzędu spójności siatki oraz wzajemnego położenia krzywej i siatki, a nie rodzaju i stopnia krzywizny krzywej i ma ekstrema dla tych wartości kąta  $\theta$  co i bezwzględny szczytowy błąd.

Średnio wartości względnych szczytowych błędów aproksymacji

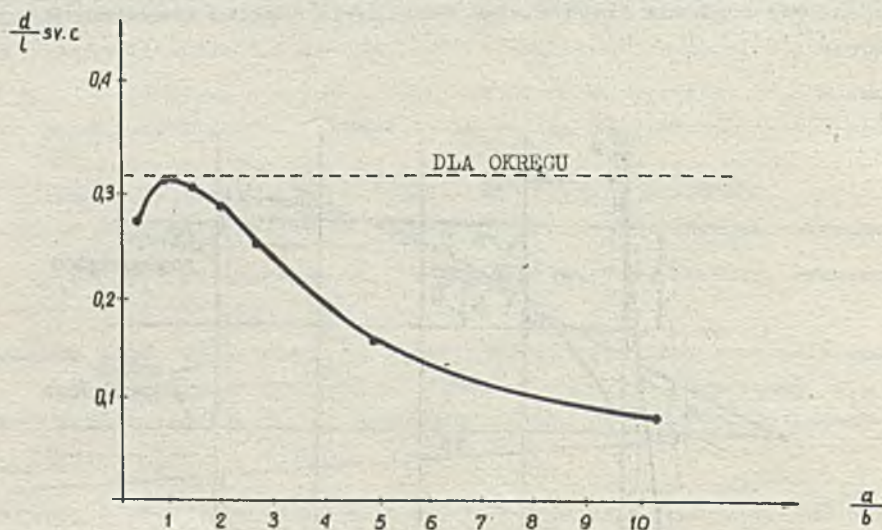
Wyznamy, jako charakterystykę jakości aproksymacji, średnią wartość błędu względnego (szczytowego) określoną za pomocą wzoru:

$$\frac{d}{l} \bar{\epsilon}_r = \frac{\int \frac{d_i}{l_i} ds(\alpha) d\alpha}{\int ds(\alpha) d\alpha} \quad (16)$$

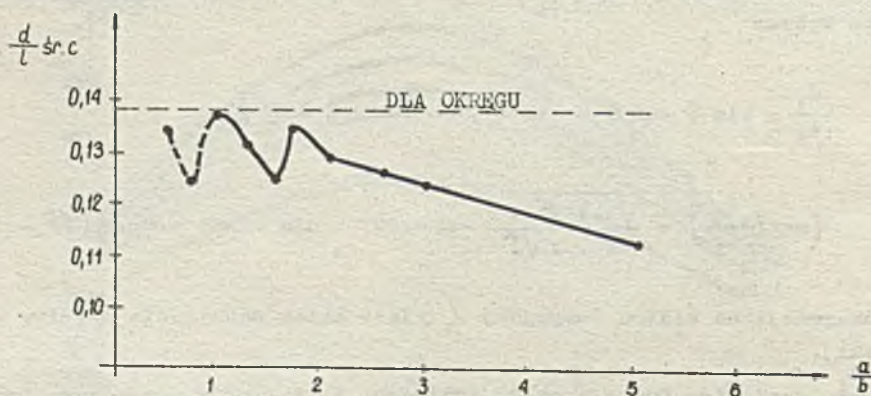
t.j. jako uśrednioną wzdłuż łuku krzywej wartość względnego szczytowego błędu aproksymacji.

Wzorów na  $\frac{d}{l} \bar{\epsilon}_r$  dla poszczególnych krzywych w pracy niniejszej nie przytaczamy ze względu na ich złożoną (niedyskusyjną) postać. Z umieszczonych niżej rysunków przedstawiających średnie wartości względnego błędu szczytowego dla (postaci kanonicznych) poszczególnych krzywych 2-stopnia (aproksymowanych na nieodwróconych w stosunku do osi układu siatkach) wynika istotna zależność rozważanego błędu od rodzajów krzywych i ich parametrów geometrycznych oraz rzędu spójności siatki.

Ogólnie można stwierdzić, że dowolna z trzech zdefiniowanych i wyznaczonych wartości błędu lokalnego, tj. kres górny wartości szczytowej, względny szczytowy błąd oraz wartość średnia względnego szczytowego błędu są ponad dwukrotnie mniejsze w aproksymacji na siatce 8-spójnej.



Rys. 13. Wartość średnia względnego błędu szczytowego aproksymacji elipsy na siatce 4-spójnej

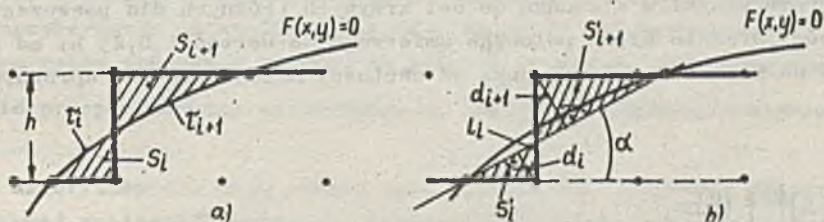


Rys. 14. Wartość średnia względnego błędu szczytowego aproksymacji elipsy na siatce 8-spójnej

Średnio globalna ocena jakości aproksymacji

Średni moduł pola odniesiony do odcinka (krótkiej) cięciwy jako względny połowy błąd

Wprowadzamy, jako miarę jakości aproksymacji krzywych na siatkach kwadratowych, miarę połową, będącą modulem pola między krzywą i aproksymującą ją łamanką, przypadającą na jednostkową długość krzywej. Jakość aproksymacji krzywej  $F(x,y) = 0$  na siatce kwadratowej (o określonym rzędzie spójności) charakteryzować będziemy wyrażeniem  $\frac{S_1}{l_1}$  (rys. 15a).



Rys. 15. Ilustracja do wyznaczania względnego błędu połowego aproksymacji krzywej na siatce 4-spójnej

Dla przejrzystości obliczeń dokonujemy następującego uproszczenia (rys. 15b). Łuk krzywej wyznaczony przez punkty przecięcia się krzywej z dwoma sąsiednimi, równoległymi liniami siatki zastępujemy cięciwą i wyznaczamy średnią arytmetyczną sumy pól  $S'_i$  oraz  $S'_{i+1}$ . W ten sposób w miejsce wyrażenia  $\frac{S_1}{l_1}$  wyznaczamy wyrażenie  $\frac{S'_1}{l_1}$  śr.

Przy tego typu uproszczeniach błąd połowy określa się wyrażeniem:

$$\frac{S'_1}{l_1} \text{ śr} = \frac{h(1 + \cos^2 \alpha)}{8 \cos \alpha} \quad 0 < \alpha < \frac{\pi}{4} \quad (17)$$

w aproksymacji na siatce 4-spójnej oraz wyrażeniem:

$$\frac{S'_1}{l_1} \text{ śr} = \frac{2 + \sin^2 \alpha - 4 \sin \alpha \cos \alpha}{8(\cos \alpha - \sin \alpha)} h, \quad 0 \leq \alpha < \arctan \frac{1}{2} \quad (18)$$

w aproksymacji na siatce 8-spójnej.

Średni moduł pola odniesiony do odcinka (krótkiej) cięciwy zależy więc jedynie od rzędu spójności siatki kwadratowej oraz nie zależy od krzywizny krzywej.

Średnie wartości modułu pola odniesione do jednostkowego odcinka krzywej

Uśrednimy wyrażenie  $\frac{S'_1}{l_1} \text{ śr}$  wzdłuż łuku krzywej, zgodnie z poniższym wzorem:

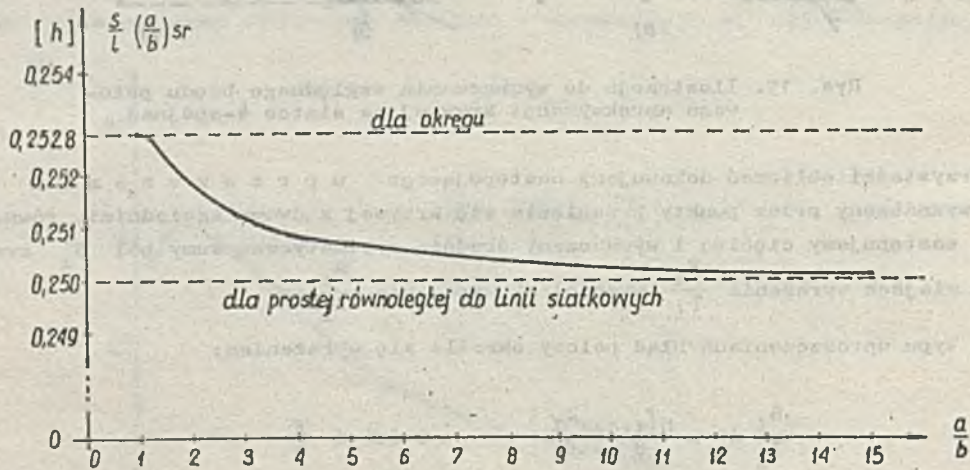
$$\frac{S'_1}{l_1} \text{ śr} = \frac{\int \frac{S'_1}{l_1} \text{ śr}(\alpha) ds(\alpha) d\alpha}{\int ds(\alpha) d\alpha} \quad (19)$$

całkując wyrażenie  $\frac{S'_1}{l_1} \text{ śr}$  wzdłuż łuku krzywej i odnosząc całkę do długości łuku, wzdłuż którego odbywało się całkowanie. W ten sposób uzależnimy stosunek  $\frac{S'_1}{l_1}$  od typu i parametrów krzywych.

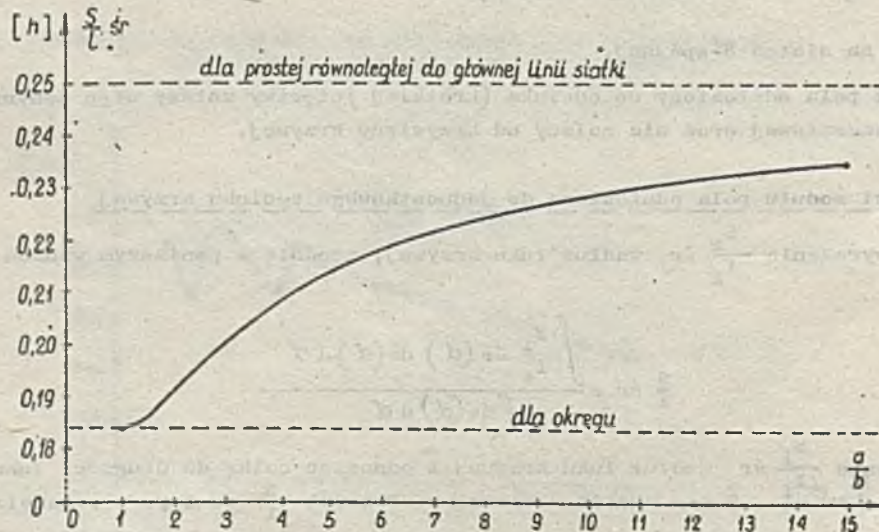
Wzory określające średnie wartości modułu pola odniesionego do jednostkowego odcinka krzywej są dość złożone i niedyskusyjne.

Zależność średniej wartości błędu polowego od podstawowych parametrów poszczególnych krzywych 2-stopnia ilustrują poniższe odpowiednie rysunki. Istotne wnioski, które wynikają z przeprowadzonej analizy wzorów na średnią wartość błędu polowego są następujące:

- wartość średnia błędu polowego w dość istotny sposób zależy od typu i parametrów geometrycznych krzywych oraz od wzajemnego położenia siatki i krzywej;
- dla pewnych wartości parametrów krzywych i ich położenia względem linii siatki, przy których występuje kres górny średniej wartości błędu względnego polowego w aproksymacji na siatce 4-spójnej - występuje kres dolny tegoż błędu w aproksymacji na siatce 8-spójnej; średnia wartość błędu polowego w aproksymacji na siatce 4-spójnej jest większa o ok. 1/3 od średniej wartości tego błędu w aproksymacji na siatce 8-spójnej;
- dla pewnych ekstremalnych wartości parametrów geometrycznych krzywych (lub ich stosunku) i pewnych kątów obrotu siatki w stosunku do osi krzywych (różnych dla poszczególnych krzywych 2-stopnia) wartości średnie błędu polowego zmierzają do wartości 0,25 h: od wartości większych - w aproksymacji na siatce 4-spójnej oraz od wartości mniejszych - w aproksymacji na siatce 8-spójnej.



Rys. 16. Wartość średnia błędu polowego aproksymacji elipsy na siatce 4-spójnej



Rys. 17. Wartość średnia błędu polowego aproksymacji elipsy na siatce 8-spójnej

#### 4. ZAKOŃCZENIE

Cechą charakterystyczną analizowanych algorytmów aproksymacji krzywych na siatkach kwadratowych, wyróżniającą je spośród innych algorytmów cyfrowej generacji jest to, że punkty aproksymujące to krzywe wyznaczane są za pomocą niewielkiej liczby elementarnych operacji sumowania i porównywania liczb stałoprzecinkowych oraz brak zjawiska kumulacji błędów. Wynika stąd atrakcyjność omawianych algorytmów, zarówno z punktu widzenia szybkości generacji, jak też prostoty układowej i dokładności aproksymacji.

Zestawienie podstawowych cech cyfrowych metod generacji, w tym również generacji na siatkach kwadratowych zawiera tabela.

Śród opracowanych algorytmów aproksymacji najprostszy logicznie jest niewątpliwie (jak wynika ze schematów blokowych algorytmów) algorytm aproksymacji na siatce 4-spójnej, najbardziej złożony - algorytm aproksymacji na siatce n-spójnej. Gładkość wygenerowanych łamanych wzrasta jednak wraz ze wzrostem rzędu spójności siatki, na których krzywo są aproksymowane.

Celowe więc będzie przeprowadzenie szczegółowszej analizy porównawczej algorytmów generacji całkowitoliczbowej.

Schematy blokowe algorytmów nie dają pełnej oceny złożoności układowej realizacji algorytmów. Układochłonność sztywnej realizacji układowej (algorytmów) zależy od skali integracji stosowanych układów, od stopnia równośćności wykonywania występujących w algorytmie operacji, od organizacji sterowania generatora oraz innych mniej znaczących czynników.

Tabela.

Zestawienie podstawowych cech cyfrowych metod generacji

Metoda generacji	Szybkość generacji	Błędy generacji	Złożoność układowa	Uniwersalność metody	U w a g i
Na mnożnikach częstotliwości MC	Bardzo duża (ok. 100 ns na jednostkę długości)	Bardzo duże (kilkadziesiąt jednostek) narastające w trakcie generacji	Proste konstrukcyjne	Uniwersalna: zwiększenie stopnia krzywizny lub przestrzenni wymiarowej krzywej wymaga dołączenia dodatkowych mnożników częstotliwości	Zmniejszenie błędów można osiągnąć proporcjonalnym zmniejszeniem przyrostu zmiennej niezależnej (proporcjonalne zmniejszenie szybkości generacji) oraz niewielką rozbudowę układu
Na cyfrowych integratorach (technika CAR)	Średnia	Duże - narastające w trakcie generacji	Średnia złożoność	Uniwersalna: zwiększenie stopnia krzywizny lub przestrzenni wymiarowej krzywej wymaga dołączenia dodatkowego bloku integratorów cyfrowych	Zmniejszenie błędów można osiągnąć proporcjonalnym zmniejszeniem przyrostu zmiennej niezależnej (proporcjonalny spadek prędkości) lub zastosowaniem dokładniejszych wzorów kwadratowych (proporcjonalna rozbudowa układu i znaczny spadek prędkości)
Na cyfrowych mnożnikach macierzowych	Średnia	Średniej wielkości	Duża złożoność	Uniwersalna: zwiększenie stopnia krzywizny lub przestrzenni wymiarowej krzywej wymaga odpowiedniej rozbudowy układu	Zmniejszenie błędów osiąga się wydłużeniem słowa informacyjnego (znaczna rozbudowa układu i spadek prędkości) oraz zmniejszeniem przyrostu zmiennej niezależnej - parametru (proporcjonalny spadek prędkości)
Na siatkach logicznych (na siatkach kwadratowych)	Duża (ok. 500 ns na jednostkę długości)	Małe (poniżej jednej jednostki) nie narastające w trakcie generacji	Średnia złożoność	Generuje tylko krzywe płaskie drugiego stopnia. Zmiana stopnia krzywizny lub przestrzenni wymiarowej krzywej wymaga opracowania innej metody lub rozwinięcia istniejącej	Zmniejszenie błędów generacji osiąga się zmniejszeniem jednostek przyrastania po ośiach współrzędnych x i y (zmniejszeniem modułu siatki h)

Z przeprowadzonego przez autora (w ramach wykonywania projektu wstępnego generatorów krzywych) oszacowania złożoności elementowej poszczególnych algorytmów, przy założonej realizacji w małej i średniej skali integracji TTL i ukierunkowaniem rozwiązań na minimalne czasy generacji wynikają następujące oszacowania liczby układów scalonych niezbędnych do wykonania poszczególnych generatorów (przy 16-bitowych rejestrach i arytmometrach):

- układ generatora aproksymującego na siatce 4-spójnej - ok. 240 układów,
- układ generatora aproksymującego na siatce 8-spójnej - ok. 300 układów,
- układ generatora aproksymującego na siatce n-spójnej - ok. 400 układów.

W założonym rozwiązaniu generatorów czasy wyznaczania jednego wektora łamanej aproksymującej (na siatce n-spójnej - średni czas generacji wektora łamanej odniesiony do średniej długości tego wektora) będą równe odpowiednio: 400 ns, 550 ns i 450 ns.

Parametry te (tj. liczbę układów scalonych oraz czas wyznaczania wektora łamanej) można w znacznym stopniu zmniejszyć realizując algorytmy w dużej i bardzo dużej skali integracji układów scalonych.

Porównując między sobą omówione algorytmy za pomocą współczynnika  $tz$  - będącego iloczynem czasu generacji (w ns) jednego wektora łamanej i ilości układów scalonych niezbędnych do realizacji układu (odpowiedniego) generatora stwierdzamy, że współczynnik ten jest najmniejszy dla generatora aproksymującego na siatce 4-spójnej ( $tz = 96$ ), znacznie większy dla generatora aproksymującego na siatce 8-spójnej ( $tz = 165$ ) i największy dla generatora aproksymującego na siatce n-spójnej ( $tz = 180$ ). Wzrasta więc ze wzrostem gładkości aproksymacji.

Dokonując wyboru algorytmu do określonego zastosowania należy uwzględnić przede wszystkim te dwa czynniki: nakłady (mierzone np. współczynnikiem  $tz$ ) oraz efektywną jakość (oszacowaną wizualnie lub mierzoną w jednej ze stosowanych w pracy metryk).

#### Literatura

- [1] DOTTING R.J., PITTEWAY M.L.V.: Algorithm for drawing ellipses or hyperbolas with a digital plotter (letter to the Editor). The Computer Journal 1968 vol. 11 nr 2 s. 120
- [2] BRESENHAM J.E.: Algorithm for computer control of a digital plotter. IBM System Journal 1965 vol. 4 nr 3 s. 225-230
- [3] CHAIKIN G.M.: Algorithm for high-speed curve generation. Computer Graphics and Image Processing 1974 vol. 3 nr 4 s. 346-349
- [4] COHEN D.: On linear differences curves. V: Computer Graphics 70. International Symposium, Brunel University, 1970
- [5] DANIELSON P.E.: Incremental curve generation. IEEE Transactions on Computers 1970 vol. C-19 nr 9 s. 783-793
- [6] DENERT E.: A method for computing points of a circle only integers. Computer Graphics and Image Processing 1973 vol. 2 nr 1 s. 83-91
- [7] FALK H.: Computer report I. Hart-soft tradeoffs. IEEE Spectrum 1974 vol. 2 nr 1 s. 34-39
- [8] FREEMAN H.: On the encoding of arbitrary geometric configurations IRE Transactions on Electronic Computers 1961 vol. EC-10 nr 2 s. 260-268
- [9] FREEMAN H.: Computer processing of line-drawing images. Computing Surveys 1974 vol. 6 nr 1 s. 57-97
- [10] GIPPENREJTER J.B., UROZAJEVA V.A.: Issledovanie dviżenia glaz pri metričeskich zadačach. Voprosy psichologii 1963 nr 6 s. 76-84
- [11] HORN B.K.P.: Circle generators for display devices. Computer Graphics and Image Processing 1976 vol. 5 nr 2 s. 280-288

- [12] JORDAN B.W., HOLM B.D., LENNON W.J.: An improved algorithm for the generation of nonparametric curves. IEEE Transaction on Computer 1973 vol. C-22 nr 12 s. 1052-1060
- [13] KLAASMAN H.: Some aspects of the accuracy of the approximate position of a straight line on a square grid. Computer Graphics and Image Processing 1975 vol. 4 nr 3 s. 225-235
- [14] KULPA Z.: Area and perimeter measurement of blobs on discrete binary pictures. Computer Graphics and Image Processing 1977 vol. 6 nr 4 s. 434-451
- [15] LAPPALAINEN P., OJALA L.: A novel digital method generating a circle test pattern for television. The Radio and Electronic Engineer 1972 vol. 42 nr 1 s. 21-27
- [16] MOKRZYCKI W.: Przyrostowa generacja krzywych drugiego stopnia w systemach obrazowania informacji. Biuletyn WAT 1975 nr 9/277 s. 73-77
- [17] MOKRZYCKI W.: Analiza cyfrowych metod układowej generacji krzywych drugiego stopnia. Rozprawa doktorska. Warszawa: WAT 1977
- [18] PARTRIDGE M.F.: Algorithm for drawing ellipses or hyperbolas with a digital plotter (Letter to the Editor). The Computer Journal 1968 vol. 11 nr 2 s. 119-120
- [19] PITTEWAY M.L.V.: Algorithm for drawing ellipses or hyperbolas with a digital plotter. The Computer Journal 1967 vol. 10 nr 3 s. 282-289
- [20] PITTEWAY M.L.V.: Integer circles, etc. Some further thoughts. Computer Graphics and Image Processing 1974 vol. 3 nr 3 s. 262-265
- [21] PITTEWAY M.L.V., BOTTING R.J.: Integer Circles, etc. - Three move extension of Bresenham's algorithm. Computer Graphics and Image Processing 1974 vol. 3 nr 3 s. 260-261
- [22] POLJAKOV V.G.: Estestvonnaja diskretnaja model' risunka, nekotorye asimptotičeskie ee svojs-tva i predskazanie choda sledjaščoj razvjortki. Problemy peredači informacii 1970 t. IV vyp. 3
- [23] POLJAKOV V.G.: Linejno-lomanye diskretnye modeli risunka dlja sistem sledjaščej razvjortki. Sb. Ikonika. Prostranstvennaja filtracija izobraženii. Fotografičeskie sistemy. Moskva: Izd. Nauka 1970
- [24] RODNIN P.R., LINKIN G.A., TATARENKO V.N.: Obrabotka fasonnyh poverchnostej na stankach s čislovym programnym upravlenijem. Kijev: Izd. Technika 1976
- [25] STOCTON F.G.: Algorithm 162, move plotting. Communication of the ACM 1963 vol. 4 nr 2 s. 161
- [26] SUTCLIFFE D.C.: An algorithm for drawing the curve  $f(x,y) = 0$ . The Computer Journal 1972 vol. 19 nr 3 s. 246-249
- [27] ZAVALIŠIN N.V., MUČNIK I.B.: Modeli zritel'nogo vospriniatija i algoritmy analiza izobraže-nija. Moskva: Izd. Nauka 1974
- [28] ZIEMKIEWICZ A.: Wybrane procedury graficzne w systemach komputerowych. Prace IMM 1975 zeszyt 2 s. 43-62
- [29] ZORAT A., BONGIOVANI G., LUCCIO F.: Discrete equation of the straight line. IEEE Transaction on Computers 1975 vol. C-24 nr 3 s. 310-313





mgr Danuta KRYŃSKA

Instytut Maszyn Matematycznych

## Analiza wybranych metod numerycznego całkowania układów równań różniczkowych zwyczajnych I rzędu stosowanych w obliczeniach inżynierskich i naukowych

### 1. Wprowadzenie

W pracy porównano kilka metod numerycznego całkowania układów równań różniczkowych zwyczajnych I rzędu postaci:

$$\frac{dy_i}{dx} = f_i(x, y_1, \dots, y_n) \quad i=1, \dots, n \quad (1)$$

Omówiono kryteria wyboru tych metod i kryteria ich porównania, podano powody, dla których to porównanie było prowadzone oraz krótki opis ich działania i wyniki porównania.

Metody testowano w Zakładzie Zastosowań Naukowych i Technicznych IMM w latach 1974 - 1976 na przykładach zaczerpniętych z techniki satelitarnej i ruchu komet (na maszynie IBM 370/145). Przykłady te przyjęto dlatego, gdyż stanowiły bardzo dobry test porównawczy dla równań ruchu; ostatecznie pracę zakończono w Zakładzie Programów dla Konstrukcji Technologii i Eksploatacji Maszyn (DC).

Spośród wybranych metod udało się wyróżnić wypadki, w których zastosowanie konkretnej metody daje dużo lepsze rezultaty niż przy zastosowaniu innych porównywanych w tej pracy.

Do najstarszych metod, które stosuje się w numerycznym rozwiązywaniu układów równań różniczkowych, opisujących zjawiska występujące w technice, należą metody typu Runge-Kutty i metody typu predictor-corrector. Są one bardzo proste do zaprogramowania, a jednocześnie przy odpowiednim doborze kroku dają dokładne rozwiązania i są do tej pory najczęściej stosowane.

Pewną rolę odgrywa tu również tradycja oraz wygoda użytkowników, którzy stosując standardową metodę nie próbują szukać innych, co jak wykaże dalsza analiza jest niesłuszne. Tymczasem, stworzono już wiele nowych metod przybliżonego całkowania równań różniczkowych. Metody te w zależności od natury równania i dokładności rozwiązania, jaką pragnie uzyskać użytkownik, same w trakcie działania tak dobierają krok, tzn. zwiększają go lub zmniejszają, aby tę dokładność uzyskać. Niestety, nie są one jeszcze dostatecznie znane i rozpowszechnione.

Dokonano więc analizy kilku wybranych metod celem wskazania, kiedy stosować te, a nie inne metody i jakie korzyści dają one użytkownikowi. Wydaje się również celowe stworzenie biblioteki procedur, która wraz z opisem rezultatów przeprowadzonej analizy metod w niej zawartych, miałaby za zadanie ułatwić użytkownikowi wybranie najodpowiedniejszej metody, w zależności od typu rozwiązywanego zadania oraz od wymagań użytkownika.

### 2. Metody przyjęte do analizy

#### Wybrane metody

Metody numerycznego całkowania równań różniczkowych zwyczajnych można podzielić na dwie klasy. Jedną z nich stanowią metody stałokrokowe, tj. metody, którym dla całego przedziału całkowania nadaje się stały krok i metody zmiennokrokowe, tj. metody, w których rozmiar kroku jest w trakcie działania metody dostosowywany automatycznie przez samą metodę tak, aby uzyskać żadaną dokładność.

Metody typu Runge-Kutty i predictor-corrector zalicza się do metod stałokrokowych.

Spośród metod typu Runge-Kutty do przeprowadzania analizy, wybrano metodę Runge-Kutty IV rzę-

du [2], zaś spośród metod typu predictor-corrector wybrano metodę Hamminga [2]. Metody te wybrano, ponieważ są najbardziej znane i najczęściej stosowane przez użytkowników.

Spośród metod zmiennokrokowych wybrano metody Bulirsch-Stoera [3], Nordsiecka [4] i Runge-Kutty ze zmiennym krokiem całkowania [1].

Ostatnie trzy metody wybrano z dwóch powodów: pierwszy - mała popularność tych właśnie metod, drugi - każda z nich działa na innej zasadzie niż pozostało; porównywanie metod jest wtedy celowe, gdy istnieją między nimi istotne różnice.

Tak więc metoda Bulirscha-Stoera jest metodą ekstrapolacyjną, metoda Nordsiecka jest metodą iteracyjną, zaś metoda Runge-Kutty ze zmiennym krokiem całkowania jest modyfikacją metody Runge-Kutty IV rzędu. Wszystkie wybrane metody są stabilne.

#### Zasady prowadzonych porównań

Przyjęto następująco kryteria porównań wyżej przedstawionych metod:

- długość czasu obliczeń daną metodą dla zadanej dokładności rozwiązania dla różnych układów równań różniczkowych zwyczajnych,
- wygoda stosowania, tzn. ilość informacji jaką musi dostarczyć użytkownik, aby rozpocząć obliczenia, ilość informacji, jaką można uzyskać przy zastosowaniu danej metody i złożoność algorytmu,
- objętość pamięci zajmowanej przez procedury opisujące dane metody.

Kryteria te wybrano, gdyż należą do ważniejszych spośród spotykanych przy stosowaniu metod numerycznego całkowania równań różniczkowych.

#### Postać modułów programowych biblioteki NCRRZ

Programy opisujące podane metody numerycznego całkowania równań różniczkowych zwyczajnych, stanowią zaczątek biblioteki NCRRZ (numeryczne całkowanie równań różniczkowych zwyczajnych) tworzonej w Zakładzie DC na użytek programów wspomagających prace inżynierskie. Dlatego opracowane były w ten sposób, aby ujednoczyć ich formę i sposób korzystania.

Tak więc każdy z zamieszczonych programów ma postać segmentu SUBROUTINE. Istotną cechą organizacji tych procedur jest to, że każda z nich tworzy zamkniętą całość, tzn. raz wywołana procedura nie odwołuje się w trakcie działania do programu wywołującego.

Po zakończeniu działania, procedury te na wyjściu dają rozwiązania:

- dla metod stałokrokowych - w punkcie  $x_0 + n \cdot h$ , przy czym  $x_0$  jest punktem początkowym przedziału całkowania,  $h$  jest krokiem całkowania, zaś  $n$  jest liczbą kroków, po której użytkownik chce zakończyć działanie procedury;
- dla metod zmiennokrokowych - w punkcie  $x_{max}$  będącym końcem przedziału całkowania zadany przez użytkownika.

Wyjątek stanowi procedura opisująca metodę Runge-Kutty, ze zmiennym krokiem całkowania. W odróżnieniu od pozostałych procedur opisujących metody zmiennokrokowe, procedura ta nie sprawdza, czy osiągnięto punkt końcowy przedziału całkowania; sprawdzenia tego użytkownik musi dokonywać w segmencie wywołującym. Po wyjściu z tej procedury uzyskuje się rozwiązanie w punkcie  $x_0 + h$ , przy czym  $x_0$  jest punktem początkowym przedziału całkowania, zaś  $h$  jest krokiem, który dobrała procedura dla zapewnienia zadanej przez użytkownika dokładności. Tak więc, aby otrzymać rozwiązanie w zadanym punkcie, w tym przypadku nie wystarcza jednokrotne wywołanie procedury.

Każda z podanych procedur korzysta z procedury wyznaczającej prawe strony układu (1), którą użytkownik musi napisać z zachowaniem następujących reguł:

- procedura ta ma być segmentem typu SUBROUTINE o nazwie DIFFUN,
- jako parametry formalne tej procedury należy wprowadzić:  
T - zmienna niezależna,

Y - wektor rozwiązań,

DY - wektor wartości prawych stron w punkcie (T,Y).

Tak więc nagłówek procedury musi mieć postać:

SUBROUTINE DIFFUN (T,Y,DY)

Ponieważ wszystkie przedstawione w tej pracy procedury zaprogramowano w podwójnej precyzji, dlatego też parametry procedury DIFFUN także muszą być podwójnej precyzji.

Do testowania wymienionych metod były użyte:

- procedura DNOR - opisująca metodę Nordsiecka oraz
- procedura RUNGE - opisująca metodę Runge-Kutty IV rzędu, będące opracowaniami własnymi autor-ki,
- procedura RUNKUT, zaczerpnięta z opracowania [1], opisująca zmiennokrokową metodę Runge-Kutty IV rzędu,
- procedura BUSTER, zaczerpnięta z opracowania [1], opisująca metodę Bulirscha-Stoera oraz
- procedura HAMMOD, zaczerpnięta z opracowania [2], opisująca metodę Hamminga.

W procedurach BUSTER i HAMMOD dokonano pewnych zmian w stosunku do wersji oryginalnych w celu ułatwienia korzystania z tych procedur i ujednoczenia ich formy.

W wersji oryginalnej procedury opisującej działanie metody Bulirscha-Stoera, po jednym wołaniu tej procedury otrzymywane było rozwiązanie układu (1) w punkcie  $X + H$ , gdzie  $H$  było wielkością kroku dobraną przez procedurę, przy czym na ogół  $X + H$  nie było punktem interesującym użytkownika. Z tego powodu w procedurze BUSTER wprowadzono nowy parametr TMAX taki, że użytkownik po jednym wywołaniu tej procedury otrzyma wartość rozwiązania w interesującym go punkcie TMAX.

Oprócz tego w tej procedurze wprowadzono wskaźnik podający, ile razy w trakcie działania tej procedury wywoływana była prawa strona równania (1).

W opracowaniu oryginalnym [2] program opisujący metodę Hamminga jest napisany w formie segmentu FUNCTION HAMMING, wyznaczającego część predictor i corrector metody.

Aby korzystać z tego programu użytkownik musi w segmencie wywołującym dokonać następujących operacji:

- wykonać pewną metodą trzy pierwsze kroki całkowania, aby uzyskać rozwiązanie w punkcie  $X + H$ ,  $X + 2 \cdot H$ ,  $X + 3 \cdot H$ , oraz wartości prawych stron układu (1) w punktach  $X$ ,  $X + H$ ,  $X + 2 \cdot H$ ;
- wywołać część predictor funkcji HAMMING;
- wyznaczyć odpowiednie wartości prawych stron układu (1);
- wywołać część corrector funkcji HAMMING;
- wyznaczyć odpowiednie wartości prawych stron układu (1).

Wynik, który użytkownik otrzymuje po dokonaniu tych operacji jest przybliżonym rozwiązaniem (1) w punkcie  $X + 4 \cdot H$ .

Procedura HAMMOD łączy w całość część predictor, część corrector oraz wyznaczenie prawych stron układu (1). Procedura ta składa się z następujących części:

- części, w której metodą Runge-Kutty IV rzędu wykonuje się trzy pierwsze kroki całkowania;
- procedury HAMMING wyznaczającej część predictor i corrector metody Hamminga;
- procedury RUNGE opisującej metodę Runge-Kutty IV rzędu;
- procedury DIFFUN, którą musi dołączyć użytkownik według reguł podanych wyżej.

Wskaźnik NII dołączony do parametrów formalnych procedury HAMMOD wskazuje, po ilu krokach należy zakończyć całkowanie. Przy tak zorganizowanej strukturze programu, po jednym wywołaniu procedury HAMMOD otrzymuje się rozwiązanie w punkcie  $X + NH \cdot H$ , przy czym w trakcie działania procedura HAMMOD nie komunikuje się z segmentem wywołującym.

### 3. Opis algorytmów

Przed przystąpieniem do opisu algorytmów, dla ułatwienia oznaczeń układ (1) będzie zastąpiony pojedynczym równaniem różniczkowym

$$\frac{dy}{dx} = f(x, y); \quad y(x_0) = y_0$$

w którym  $y(x)$  i  $f(x, y)$  są w przypadku ogólnym wektorami:

$$y(x) = \begin{bmatrix} y_1(x) \\ \vdots \\ y_n(x) \end{bmatrix}; \quad f(x, y) = \begin{bmatrix} f_1(x, y, \dots, y_n) \\ \vdots \\ f_n(x, y_1, \dots, y_n) \end{bmatrix}$$

#### Metoda Runge-Kutty IV rzędu $\square$

Metoda ta jest metodą stałokrokową, tzn. użytkownik zadaje stały krok  $h$ , z którym obliczenia są prowadzone na całym przedziale całkowania. Jej działanie polega na przedstawieniu różnicy między dwiema kolejnymi wartościami rozwiązania  $y_{n+1}$  i  $y_n$ , gdzie  $y_{n+1} = y(x_n + h)$ ,  $y_n = y(x_n)$  w postaci

$$y_{n+1} - y_n = \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (3)$$

gdzie  $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}h k_1\right)$$

$$k_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}h k_2\right)$$

$$k_4 = f(x_n + h, y_n + h k_3)$$

przy czym  $x_n = x_0 + n \cdot h$

$x_0$  = punkt początkowy przedziału całkowania.

Jeżeli przez  $y(x)$  oznaczymy dokładne rozwiązanie równania (2) w punkcie  $x$ , to rozwiązanie  $y(x, h)$  otrzymane metodą Runge-Kutty IV rzędu ma postać:

$$y(x, h) = y(x) + O(h^4)$$

#### Metoda Hamminga $\square$

Metoda ta jest metodą stałokrokową typu predictor-corrector. Jest to metoda rzędu IV. Przy oznaczeniach:

$$y_i = y(x_i), \quad x_i = x_0 + i h$$

$$f_i = f(x_i, y_i)$$

$h$  - krok całkowania zadany przez użytkownika

działanie metody opisuje się jak poniżej.

Jako pierwszy etap obliczeń wykonuje się część predictor, tzn. oblicza się początkową, przybliżoną wartość rozwiązania w punkcie  $x_{i+1}$  korzystając z wartości w punktach  $x_i, x_{i-1}, x_{i-2}$  i  $x_{i-3}$ , tzn.

$$y_{i+1}^{(0)} = y_{i-3} + \frac{4}{3} h (2f_i - f_{i-1} + 2f_{i-2}) \quad (4)$$

$$\bar{y}_{i+1}^{(o)} = y_{i+3}^{(o)} + \frac{112}{121} (y_1^{(o)} - y_i^{(o)}) \quad (5)$$

Wartość  $\bar{y}_{i+1}^{(o)}$  otrzymaną w części predictor, poprawia się w części corrector według formuły

$$y_{i+1} = \frac{1}{8} (9y_i - y_{i-2}) + \frac{3}{8} h \left( \bar{y}_{i+1}^{(o)} + 2y'_i - y'_{i-1} \right) \quad (6)$$

gdzie

$$\begin{aligned} y_{i+1}^{(o)} &= f(x_{i+1}, y_{i+1}^{(o)}) \\ y'_i &= f(x_i, y_i) \end{aligned}$$

Wartość  $y_{i+1}$  jest przyjmowana jako rozwiązanie (2) w punkcie  $x_{i+1} = x_0 + (i+1) \cdot h = x_i + h$ .

Metoda Bulirscha-Stoera  $\square$

Działanie tej metody oparte jest na następującej zasadzie: jeżeli pewna metoda dyskretna, w tym wypadku zmodyfikowana reguła środkowego punktu, daje rezultat  $T(h, x) = y(x) + O(h^r)$  dla skończonego kroku  $h \neq 0$ , to dokładne rozwiązanie  $T(0)$  otrzymuje się przez wyekstrapolowaną wartość  $T_m(0)$  wielomianu interpolacyjnego lub funkcję wymierną  $T_m(h)$  spełniającą warunek

$$T_m(h_j) = T(h_j) \quad j = 0, \dots, m$$

dla ciągu  $h_j$  kroków zbieżnych do zera.

Metodą dyskretną zastosowaną w metodzie Bulirscha-Stoera jest zmodyfikowana reguła środkowego punktu o następującym algorytmie:

Jeżeli  $x = x_0 + l \cdot h$   $l$  - całkowite, to

$$T(h) = T(h, x)$$

jest zdefiniowane rekurencyjnie przez:

$$x_{i+1} = x_i + h \quad i = 0, \dots, l-1$$

$$\eta(x_1, h) = y_0 + hf(x_0, y_0); \quad (7)$$

$$\eta(x_{i+1}, h) = \eta(x_{i-1}, h) + 2hf(x_i, \eta(x_i, h)); \quad i = 1, \dots, l-1$$

$$S(h, x) = \frac{1}{2} \left[ \eta(x_1, h) + \eta(x_{l-1}, h) + hf(x_1, \eta(x_1, h)) \right];$$

$$T(h, x) = S\left(\frac{h}{2}, x\right)$$

Jeżeli  $\hat{T}_m^{(1)}(h)$  oznacza funkcję wymierną zdefiniowaną przez warunek:

$$\hat{T}_m^{(1)}(h_k) = T(h_k, x) \quad k = 1, i+1, \dots, i+m$$

gdzie  $\{h_k\}$  jest ciągiem ściśle malejącym kroków dążących do zera, wartości ekstrapolowane

$$T_m^{(i)} = \hat{T}_m^{(1)}(0) = T(0, x)$$

mogą być obliczone z  $T(h_i, x)$  przez następujący układ formuł:

$$\left\{ \begin{array}{l} T_{-1}^{(i)} = 0 ; \\ T_0^{(i)} = T(h_1, x) \\ T_k^{(i)} = T_{k-1}^{(i+1)} + \frac{T_{k-1}^{(i+1)} - T_{k-1}^{(i)}}{\left(\frac{h_1}{h_{1+k}}\right)^2 \left[ 1 - \frac{T_{k-1}^{(i+1)} - T_{k-1}^{(i)}}{T_{k-1}^{(i+1)} - T_{k-2}^{(i+1)}} \right]} ; \end{array} \right. \quad (8)$$

Metoda Nordsiecka VI rzędu  $\square$

Metoda ta jest oparta na następującej zasadzie: jeżeli oznaczymy przez

$$\left. \begin{array}{ll} a(x) = \frac{h}{21} P_5^{II}(x); & c(x) = \frac{h^3}{h1} P_5^{IV}(x) \\ b(x) = \frac{h^2}{31} P_5^{III}(x); & d(x) = \frac{h^4}{51} P_5^V(x) \end{array} \right\} \quad (9)$$

gdzie  $P_5(x)$  jest wielomianem interpolacyjnym dla rozwiązania równania (2)  $y(x)$ , to korzystając z równania (2) oraz z rozwinięcia wielomianu interpolacyjnego  $P_5(x)$  w szereg Taylora dostaniemy formułę na rozwiązanie  $y$  w punkcie  $x + h$  w postaci:

$$\left\{ \begin{array}{l} y(x + h) = y(x) + h \left\{ f(x) + a(x) + b(x) + c(x) + d(x) + \frac{95}{288} e(x) \right\} \quad (10) \\ f^P = f(x) + 2a(x) + 3b(x) + 4c(x) + 5d(x) \\ a(x + h) = a(x) + 3b(x) + 6c(x) + 10d(x) + \frac{25}{24} e(x) \\ b(x + h) = b(x) + 4c(x) + 10d(x) + \frac{35}{72} e(x) \\ c(x + h) = c(x) + 5d(x) + \frac{5}{48} e(x) \\ d(x + h) = d(x) + \frac{1}{120} e(x) \\ e(x) = f(x + h) - f^P \end{array} \right. \quad (11)$$

Ponieważ w chwili startu metody znane są tylko warunki początkowe, tzn.  $x_0$  i  $y_0$ , a nie znane są pochodne  $a, b, c, d$  w punkcie  $x_0$  ani krok całkowania  $h$ , przy którym uzyskana będzie dokładność  $\xi$ , należy je wyznaczyć w początkowym etapie. Początkowo, stosując próbny krok  $h$ , wyznacza się je następująco: podstawia się  $a=b=c=d=0$ , a następnie z równania (2) i (10) stosując dwukrotnie wzór (10) wyznacza się wartości  $y$  w punkcie  $x_0 + h$ , tzn.  $y^{(1)}$  i  $y^{(2)}$ . Przyjmuje się, że  $y^{(2)} = y(x + h)$ . Następnie stosując formuły (10) i (11) wyznacza się wartości  $y, f, a, b, c, d$  w punkcie  $x_0 + h$ . Stosując dalej formuły (10) i (11) oblicza się wartości  $y, f, a, b, c, d$  aż do punktu  $x_0 + 4h$ , po czym znów stosując te same formuły wraca się do punktu  $x_0$ . Otrzymane w ten sposób wartości  $a, b, c, d$  w punkcie  $x_0$  są poprawiane tak, jak poprzednio jeszcze trzy razy, tzn. jeszcze trzy razy wyznaczone są wartości  $y, a, b, c, d, f$  w punkcie  $x + 4h$  i powroty do punktu  $x_0$ . Jeżeli sprawdzane w tym czasie testy stabilności i dokładności dały wynik pozytywny, warunki początkowe dla  $a, b, c, d$  wyznaczone tą metodą przyjmuje się jako startujące do dalszego działania. Jeżeli zaś testy błędów i stabilności nie będą spełnione, wówczas krok  $h$  jest zmniejszony i całe działanie trzeba zaczynać od początku. Już po wyznaczeniu wartości  $h, a, b, c, d$  w punkcie  $x_0$  wartości  $y, f, a, b, c, d$  w punktach  $x + h$  wyznacza się stosując formuły (10) i (11) oraz równanie (2).

Metoda Runge-Kutty ze zmiennym krokiem całkowania  $\square$

Jest to metoda zmiennokrokowa o następującym algorytmie: startując z warunku początkowego  $y(x_n) = y_n$  wyznacza się zwykłą metodą Runge-Kutty IV rzędu wartość w punkcie  $x_n + \frac{h}{2}$  tzn.  $y(x_n + \frac{h}{2}) = y_n$  oraz startując z punktu  $x_{n+\frac{1}{2}} = x_n + \frac{h}{2}$  wartość  $y(x_n + h) = y_{n+1}$

$$y_n \xrightarrow{\cdot \frac{h}{2}} y_{n+\frac{1}{2}} \xrightarrow{\frac{h}{2}} y_{n+1}^{(1)}$$

następnie startując z punktu  $x_n$  metodą Runge-Kutty IV rzędu wyznacza się z krokiem  $h$  wartość  $y$  w punkcie  $x_{n+1}$  tzn.  $y_{n+1}^{(2)}$ .

$$y_n \xrightarrow{h} y_{n+1}^{(2)}$$

dalej przeprowadzane jest porównanie wielkości  $y_{n+1}^{(1)}$  oraz  $y_{n+1}^{(2)}$  i jeżeli różnica ich wartości nie jest zbyt duża, wyznaczana jest wartość  $y_{n+1} = y(x_n + h)$  za pomocą formuły

$$y_{n+1} = \frac{1}{31} (32 y_{n+1}^{(2)} - y_{n+1}^{(1)}) \quad (12)$$

Jeżeli wynik porównania jest niezadowolający, to krok  $h$  zmniejsza się i działanie zaczyna się od początku.

4. Analiza wyników prowadzonych porównań

Porównanie ogólnie

Główną zaletą metody Runge-Kutty IV rzędu jest jej prosty algorytm, w związku z czym bardzo łatwo daje się zaprogramować i dlatego jest atrakcyjna dla użytkownika, który nie ma gotowego programu. Jest to metoda samostartująca, a więc oprócz warunków początkowych i rozmiaru kroku całkowania użytkownik nie musi dawać żadnych dodatkowych informacji wejściowych.

Metoda Runge-Kutty IV rzędu jest metodą stałokrokową, a więc jeżeli użytkownik nie potrafi bez przeprowadzenia testu przewidzieć, jaki krok całkowania da mu oczekiwaną dokładność rozwiązania, musi wykonać kilka prób, zanim taki krok dobierze, tym bardziej, że błąd, który powstaje po wykonaniu jednego kroku tą metodą jest bardzo trudny do oszacowania. Problem ten nie jest istotny, jeżeli prawa strona równania (2) jest prosta. Staje się on ważny, jeżeli prawą stronę równania (2) wyznacza się za pomocą skomplikowanego algorytmu, bo wtedy takie próby pochłaniają bardzo dużo czasu pracy maszyny. Wynika to z faktu, że na jednym kroku całkowania, w metodzie tej prawą stronę równania (2) wyznacza się aż cztery razy.

Ponieważ metoda Runge-Kutty IV rzędu jest stałokrokowa, na całym przedziale całkowania stosuje ten sam krok. Jeżeli użytkownika interesują wyniki po każdym przecałkowanym stałym kroku, warto ją stosować. Jednak, jeżeli użytkownika interesuje tylko wynik ostateczny, warto się zastanowić, czy nie lepiej zastosować jednej z metod zmiennokrokowych. Zdarza się, że nie na całym przedziale całkowania trzeba przyjmować krok jednej zadanej długości, aby otrzymać oczekiwaną dokładność rozwiązania. Istnieją przypadki, gdy największy krok, który można przyjąć dla zadanej dokładności na pewnym odcinku przedziału całkowania, jest kilkaset razy dłuższy od kroku, jaki należy przyjąć na innym odcinku. Stosowanie wówczas stałokrokowej metody Runge-Kutty IV rzędu powoduje niepotrzebne wydłużenie czasu pracy maszyny.

Metoda Hamminga jest również metodą stałokrokową podobnie, jak metoda Runge-Kutty IV rzędu. Tak więc odnoszą się do niej te same zarzuty, co do metody Runge-Kutty dotyczące stałego kroku.

Jeżeli chodzi o ilość wyznaczeń prawej strony równania (2), to w metodzie Hamminga wyznacza się ją tylko 2 razy, jednakże nie wpływa to na skrócenie czasu obliczeń. Metoda ta ma jeszcze jedną niedogodność: stosując ją należy oprócz wartości  $y$  w punkcie  $x_0$  wyznaczyć inną metodą wartości  $y$  w punktach  $x_0 + h$ ,  $x_0 + 2h$  oraz  $y_0 + 3h$ . Nie jest więc ona metodą samostartującą.

Zaletami tej metody są: prosty i krótki algorytm, łatwość, w porównaniu z metodą Runge-Kutty, szacowania błędu powstającego po wyznaczeniu jednego kroku tą metodą.

Pozostałe trzy metody mają bardzo ważną zaletę, a mianowicie zmienny krok całkowania, co bardzo ułatwia i przyspiesza korzystanie z tych metod. Jednak nie zawsze są one lepsze niż metoda Runge-Kutty IV rzędu, czy metoda Hamminga.

Metoda Nordsiecka ze względu na to, że w procesie startu prawą stronę wyznacza się minimum aż 64 razy, jest niewygodna do stosowania w wypadku krótkich przedziałów całkowania. Przy tym, jeżeli krok, jaki użytkownik zada jej jako początkowy, będzie znacznie dłuższy niż niezbędny dla zadanej dokładności, wówczas proces startu przedłuży się tak, że istotnie wpłynie na czas pracy maszyny.

W metodzie Runge-Kutty ze zmiennym krokiem całkowania na każdym dobranym kroku całkowania prawą stronę równania wyznacza się 12 razy. Gdy mamy do czynienia z równaniami, w których krok całkowania na całym przedziale prawie się nie zmienia, fakt ten ogromnie wydłuża czas pracy maszyny. Natomiast warto ją stosować wtedy, gdy wiadomo, że mogą powstać znaczne różnice w doborze kroku na całym przedziale całkowania.

Metoda Bulirscha-Stoera oprócz faktu, że jest zmiennokrokowa, ma jeszcze inne bardzo ważne zalety. Rząd aproksymacji nie jest w niej ustalony i jest automatycznie dobierany dla rozwiązywanego konkretnego problemu. Nie ma też konieczności wyznaczania dodatkowych wielkości, tak jak na przykład w metodzie Nordsiecka - wyznaczanie pochodnych rozwiązania. Jak pokażą wyniki porównania, metoda ta rzeczywiście daje najlepsze wyniki.

#### Charakterystyka wyników testowania

Do pierwszego etapu porównania przedstawionych metod wybrano niżej wymienione zagadnienia początkowo.

- Ruch komety Halley'a opisany układem równań

$$\frac{d^2 y_i}{dx^2} = -GM \frac{y_i}{(y_1^2 + y_2^2 + y_3^2)^{3/2}} \quad i = 1, 2, 3$$

gdzie  $GM = 0.0029591220828559$

z warunkami początkowymi

$$\begin{aligned} x_0 &= 0 \\ y_1(0) &= 0.790018983636331 \\ y_2(0) &= 2.721055615798287 \\ y_3(0) &= 0.883967477972253 \\ \dot{y}_1(0) &= 0.00258830665134 \\ \dot{y}_2(0) &= -0.013040790001677 \\ \dot{y}_3(0) &= -0.002479531107762 \end{aligned}$$

Jako koniec przedziału całkowania przyjęto interwał 24240 dni.

- Ruch keplerowski o orbicie kołowej opisany układem równań tym samym, co przy ruchu komety Halley'a, przy czym

$$\begin{aligned} GM &= R^2 \\ R &= 0.07436574 \end{aligned}$$

Warunki początkowo dla tego układu to:



$$\begin{aligned} x_0 &= 0 \\ y_1(0) &= 4 ; & y_1(0) &= 0 \\ y_2(0) &= 0 ; & y_2(0) &= R/2 \\ y_3(0) &= 0 ; & y_3(0) &= 0 \end{aligned}$$

Jako koniec przedziału całkowania w tym wypadku przyjęto wartość okresu  $P = 16 \pi / T$ .

• Ruch satelity "Interkosmos Kopernik 500".

Prawa strona układu opisującego ten ruch i warunki początkowo są do wglądu w Instytucie Maszyn Matematycznych.

• Równanie Bessela

$$\frac{d^2 y}{dx^2} + \frac{1}{x} \left( y(x) + \frac{dy}{dx} \right) = 0$$

o warunkach początkowych:

$$\begin{aligned} x_0 &= 1 \\ y(x_0) &= 0.2238907791411603 \\ y'(x_0) &= -0.5767248077568665 \end{aligned}$$

Przedział całkowania  $\langle 1, 10 \rangle$

• Równanie liniowe postaci  $\frac{dy}{dx} = ay + bx$

Każda z podanych metod była testowana na powyżej przedstawionych równaniach dla zadanej dokładności  $EPS = 10^{-7}$ . Czasy obliczeń oraz obszary pamięci zajmowane przez procedury opisujące te metody, ujęto w poniższych tabelkach.

Tabela 1

Kometa Halley'a

Długość przedziału całkowania - 24240 dni

PROCEDURA	CZAS OBLICZEŃ	KROK CAŁKOWANIA
BUSTER	7.5 <sup>s</sup>	-
BNOR	12.7 <sup>s</sup>	-
RUNKUT	10.1 <sup>s</sup>	-
RUNGE	3 <sup>m</sup> 05.0 <sup>s</sup>	1 dzień
HAMMOD	4 <sup>m</sup> 35.5 <sup>s</sup>	1 dzień

Tabela 2

Ruch keplerowski o orbicie kołowej

Długość przedziału całkowania = 1 okres=680 min

PROCEDURA	CZAS OBLICZEŃ	KROK CAŁKOWANIA
BUSTER	1.7 <sup>s</sup>	-
DNOR	5.7 <sup>s</sup>	-
RUNKUT	15.2 <sup>s</sup>	-
RUNGE	3.2 <sup>s</sup>	2 min
HAMMOD	3.3 <sup>s</sup>	2 min

Tabela 3

Równanie Bessela  $\frac{d^2 y}{dt^2} + \frac{1}{t} \left( y(t) + \frac{dy}{dt} \right) = 0$

Przedział całkowania  $\langle 1, 10 \rangle$

PROCEDURA	CZAS OBLICZEŃ	KROK CAŁKOWANIA
BUSTER	0,47 <sup>s</sup>	-
DNOR	0.75 <sup>s</sup>	-
RUNKUT	0.73 <sup>s</sup>	-
RUNGE	0.88 <sup>s</sup>	0.05
HAMMOD	0.97 <sup>s</sup>	0.05

Tabela 4

Jedno okrążenie satelity "Interkosmos Kopernik 500"

PROCEDURA	CZAS OBLICZEŃ	KROK CAŁKOWANIA
BUSTER	3.8 <sup>s</sup>	-
RUNGE	15.3 <sup>s</sup>	10 s

Tabela 5

Obszary pamięci zajmowane przez poszczególne procedury

PROCEDURA	PAO
BUSTER	94 K
DNOR	102 K
RUNKUT	86 K
RUNGE	86 K
ILAMMOD	86 K

Jak widać z tab. 1 i tab. 2 dla równań ruchu komety Halley'a dużo lepsze wyniki czasowo daje metody zmiennokrokowe, podczas gdy z wyjątkiem metody Bulirscha-Stoera, dla równań ruchu keplerańskiego o orbicie kołowej lepsze są metody stałokrokowe. Wynika to stąd, że orbita komety Halley'a jest elipsą o dużym mimośrodku. Powoduje to, że na pewnych odcinkach można przyjąć krok całkowania wielokrotnie dłuższy niż na innych. Np. stosując metodę Runge-Kutty ze zmiennym krokiem całkowania najdłuższy krok całkowania przyjęty przez metodę miał wielkość - 772 dni, podczas gdy najkrótszy miał wartość - 1 dnia.

Dla orbity kołowej lepsze okazały się metody stałokrokowe ze względu na regularne zakrzywienie orbity. Stosując metody zmiennokrokowe dużo czasu straciłoby się na dobór optymalnego kroku całkowania przy zadanej dokładności rozwiązania.

Procedura opisująca metodę Bulirscha-Stoera okazała się lepsza ze względu na to, że oprócz doboru optymalnego kroku, dobiera także optymalny rząd aproksymacji, co istotnie skraca czas obliczeń.

Z tabel 1 i 2 widać, jak duże różnice w czasie obliczeń powstają przy stosowaniu metody Runge-Kutty ze zmiennym krokiem całkowania. Gdy optymalny krok całkowania jest prawie stały (dla orbity kołowej) traci się wiele czasu na 12-krotne wyznaczanie na każdym kroku prawych stron równania. W wypadku orbity eliptycznej czas obliczeń jest niepomiaralnie krótszy. W tym wypadku czas, jaki traci się przy dobieraniu kroku całkowania jest nieistotny wobec faktu, że dobierany krok może zmniejszać się nawet kilkaset razy.

Dla równania Bessela czas obliczeń każdą metodą na tak krótkim przedziale całkowania nie wykazuje specjalnych różnic. Jednakże i w tym wypadku widać, że lepsze okazują się metody zmiennokrokowe. Prawdopodobnie dla dłuższego przedziału całkowania różnice te byłyby większe. Wynika to z kształtu trajektorii rozwiązania.

Wyników testu dla równań o prawej stronie liniowej nie podano z tego względu, że w tym wypadku różnice w czasie są bardzo małe, a więc nie jest istotne, jaką metodę stosować.

Tabela 4 pokazuje wyniki porównania tylko metody Bulirscha-Stoera i stałokrokowej metody Runge-Kutty IV rzędu, ponieważ do tej pory tylko ta ostatnia była stosowana przy obliczeniach ruchu satelity "Interkosmos Kopernik 500". Porównywana była z metodą Bulirscha-Stoera z tego względu, że na podstawie poprzednich testów ta ostatnia okazała się najlepsza. Także czas obliczeń dla jednego okrążenia satelity "Interkosmos Kopernik 500" okazał się dla metody Bulirscha-Stoera czterokrotnie krótszy niż dla metody stałokrokowej Runge-Kutty IV rzędu.

Z powyższych tabel wynika, że najlepszą spośród wszystkich porównywanych metod jest metoda Bulirscha-Stoera. Dlatego też, jeżeli użytkownik nie ma pewności, jaki krok w metodach stałokrokowych należy dobrać dla oczekiwanej dokładności rozwiązania, warto stosować metodę Bulirscha-Stoera.

## 6. Wnioski

Na podstawie przeprowadzonej analizy można sformułować następujące wnioski konkretne:

- w równaniach ruchu, których trajektorią rozwiązania jest elipsa o dużym mimośrodku, korzystnie wyniki daje także metoda Runge-Kutty ze zmiennym krokiem całkowania; ze względu na czas obliczeń jest ona dużo lepsza od metod stałokrokowych, jak również od zmiennokrokowej metody Nordsiecka;

- do całkowania równań liniowych, jak również równań ruchu, których trajektoria rozwiązania jest elipsą o małym mimośrodku, bardziej od zmiennokrokowych nadają się metody stałokrokowe; krok, jaki należy dobrać, aby uzyskać żądaną dokładność jest prawie stały, na całym przedziale całkowania, nie ma więc sensu stosować metod, w których test na sprawdzenie dokładności rozwiązania przy danym kroku pochłania większość czasu pracy maszyny.

Ponadto z pracy niniejszej można wysnuć również wnioski natury ogólniejszej:

- poszczególne metody numerycznego całkowania równań różniczkowych zwyczajnych są związane z konkretnym zastosowaniem; zastosowanie dobrej metody do danego zagadnienia może w bardzo istotny sposób skrócić czas realizacji obliczeń przy tej samej dokładności; dlatego też celowe jest wyposażenie biblioteki modułów numerycznych przeznaczonej dla prac inżynierskich, w bogaty zestaw różnorodnych metod całkowania równań różniczkowych;
- przy doborze i opracowywaniu poszczególnych modułów należy kierować się przewidywanymi przyszłymi zastosowaniami i pod tym kątem je testować; celowe jest również rozszerzenie reportuaru testów;
- informacje o przewidzianej klasie zastosowań poparte wynikami testów, powinny znaleźć się w instrukcji korzystania z biblioteki.

#### Literatura

- [1] CLARK N.W.: A study of some numerical methods for the integration of systems of first-order ordinary differential equations. Rep. ANL-7428. Argonne, Ill.: Argonne National Laboratory 1968
- [2] HENRICI P.: Discrete variable methods in ordinary differential equations. New York: J. Wiley 1962
- [3] BULIRSCH R., STOER J.: Numerical Treatment of Ordinary Differential equations by Extrapolation Methods. Num. Math. 1966 nr 8 s. 1-13
- [4] NORDSIECK A.: On numerical integration of ordinary differential equations. Math. Comp. 1962 nr 16



mgr Izabella DOBIŃSKA  
Instytut Maszyn Matematycznych

## Analiza pewnych szeregów nieskończonych stosowanych w obliczeniach inżynierskich

### 1. Wprowadzenie

W obliczeniach inżynierskich często występują zadania, w których najistotniejsze jest rozwiązanie równania algebraicznego lub przestępnego, przy czym wobec braku rozwiązania dokładnego najczęściej stosowane są metody przybliżone. Rozwiązanie równań algebraicznych nie sprawia większych trudności, jest wiele metod ich przybliżonego rozwiązywania, np. metoda Newtona [1], [2]. Równania przestępne na ogół mogą być rozwiązane tylko w przybliżeniu ze względu na rodzaj wyrażeń w nich występujących. W niektórych wypadkach rozwiązywanie równań przestępnych sprowadza się do równań algebraicznych. Metoda przybliżona będzie polegała na aproksymowaniu niektórych wyrażeń występujących w tych równaniach przez wielkości prostsze, np. przez wielomiany, a następnie na rozwiązaniu nowo powstałego równania algebraicznego. Aproksymacja musi być taka, aby rozwiązanie to było także rozwiązaniem danego równania przestępnego, przy czym podstawowe znaczenie będzie tu miała dokładność obliczeń aproksymacyjnych.

Szczególnie jasno widać to w sytuacji, gdy wyrażeniami występującymi w równaniu przestępnym są szeregi nieskończone, ale takie, których sprowadzenie do postaci algebraicznej sprawia trudności lub jest niemożliwe. Wtedy bowiem, wiedząc o aproksymowanym szeregu tylko tyle, że ma on wartość skończoną (tzn. że jest zbieżny), trudno jest ocenić dokładnie, ile początkowych wyrazów szeregu należy brać praktycznie pod uwagę dla określenia tej wartości, aby reszta szeregu była nieistotna ze względu na zadaną dokładność.

W takich wypadkach należy znaleźć sposób określania potrzebnej liczby wyrazów. Dla danego szeregu będzie ona zależała od dokładności obliczeń. W tym celu mogą być wykorzystane pewne szczególne własności szeregu lub jego wyrazów. Zdarza się, że w rozpatrywanym zagadnieniu liczba ta nie przekracza pewnego górnego pułapu przy określonym rzędzie dokładności (co nie oznacza, że musi być zawsze taka sama). Może to wynikać ze wspólnych własności różnych szeregów, występujących w równaniach związanych z tym zagadnieniem itp.

W celu zilustrowania metody określania liczby znaczących wyrazów szeregu będą przeanalizowane przypadki równania przestępnego, zawierającego szereg nieskończony naprzemienny, tzn. szereg postaci (1) taki, że ciąg wyrazów  $u_k$  maleje od pewnego wyrazu, jego granicą jest 0 oraz  $u_k > 0$  dla wszystkich  $k \geq 1$ .

Problem ten wystąpił m.in. przy opracowaniu systemu GT przeznaczonego do obliczeń geometrii tras komunikacyjnych w płaszczyźnie poziomej. Zastosowanie w tym systemie proponowanej i opisanej niżej metody pozwoliło na zoptymalizowanie obliczeń niektórych parametrów krzywej.

### 2. Przykład analizy szeregów nieskończonych naprzemiennych

Ważmy pod uwagę jedno z równań podstawowych opisujących klotoidę, która występuje m.in. w projektowaniu tras krzywoliniowych,

$$H = \sum_{k=1}^{\infty} (-1)^{k+1} u_k, \quad (1)$$

gdzie

$$u_k = \frac{L}{(2k-1)! (4k-1)4k} \left( \frac{L}{2R} \right)^{2k-1} \quad (2)$$

$L$  - jest długością łuku kłotojdy,  $R$  - promieniem kola, w które ona przechodzi, a  $H$  - odsunięciem łuku kołowego od stycznej w początkowym punkcie kłotojdy [2].

W zależności od wartości, które w równaniu (1) są dane, występują tu dwa rodzaje problemów. Jeśli dane są wielkości  $L$  i  $R$ , to problem rozwiązania równania względem  $H$  sprowadza się do obliczenia sumy nieskończonej z pewną dokładnością  $\xi$ . Sprawa przedstawia się nieco inaczej, gdy  $H$  jest jedną z danych, a szukana wielkość znajduje się po prawej stronie równania (1).

Aby dalsze rozważania miały sens, trzeba sprawdzić, czy szereg (1) jest zbieżny. Z nierówności trójkąta (tzn. wartość bezwzględna sumy liczb rzeczywistych nie przekracza sumy ich wartości bezwzględnych) oraz z tego, że

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} < +\infty,$$

gdzie  $e = 2,7182\dots$  oraz  $x$  jest dowolną liczbą rzeczywistą, dostaje się

$$\begin{aligned} \left| \sum_{k=1}^{\infty} (-1)^{k+1} H_k \right| &< \sum_{k=1}^{\infty} \left| (-1)^{k+1} H_k \right| = \sum_{k=1}^{\infty} H_k = \\ &= \sum_{k=1}^{\infty} \frac{L}{(2k-1)!} \frac{L}{(4k-1)!} \left(\frac{L}{2R}\right)^{2k-1} < \\ &< \frac{L}{3 \cdot 4} \sum_{k=1}^{\infty} \frac{L}{(2k-1)!} \left(\frac{L}{2R}\right)^{2k-1} < \\ &< \frac{L}{12} \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{L}{2R}\right)^n = \frac{L}{12} e^{\frac{L}{2R}} < +\infty. \end{aligned} \tag{3}$$

Zatem szereg (1) ma wartość skończoną dla dowolnych  $L$  i  $R$ .

Ze zbieżności szeregu (1) wynika, że granica ciągu wyrazów  $H_k$  wynosi 0 [1]. Ponieważ wielkości  $L$  i  $R$  są dodatnie, to także  $H_k > 0$  dla wszystkich  $k \geq 1$ .

a) Problem obliczenia wartości szeregu nieskończonego zbieżnego (1)

Niech dane będą wielkości  $L$  i  $R$  oraz zadana wartość  $\xi$ . Aby obliczyć wartość  $H$ , korzysta się ze wzoru (1). Będzie ona sumą pewnej liczby wyrazów początkowych szeregu (1).

Następujący fakt wymaga jeszcze wyjaśnienia: czy istnieje liczba naturalna  $n$ , możliwie najmniejsza, taka, że ciąg  $\{H_k\}_{k \geq n}$  jest malejący, tzn.

$$\frac{H_{k+1}}{H_k} < 1 \tag{4}$$

dla wszystkich  $k \geq n$ . Warunek (4) po podstawieniach ze wzoru (2) jest równoważny warunkowi

$$\left(\frac{L}{2R}\right)^2 < a_k, \tag{5}$$

gdzie

$$a_k = \frac{(2k+1)(2k+2)(4k+3)}{4k-1} \tag{6}$$

Można sprawdzić, że ciąg  $\{a_k\}_{k \geq 1}$  rośnie do  $+\infty$ . Stąd i z faktu, że po lewej stronie nierówności (5) znajduje się wartość liczbowa (gdyż  $L$  i  $R$  są dane), wynika, że istnieje taka najmniejsza liczba naturalna  $n$ , że

$$\left(\frac{L}{2R}\right)^2 < a_n < a_{n+1} < a_{n+2} < \dots \quad (7)$$

Zatem warunek (5), a także równoważny mu warunek (4), jest spełniony dla wszystkich  $k > n$ . Oznacza to, że ciąg  $\{H_k\}_{k > n}$  jest malejący, gdzie liczbę naturalną  $n$  wyznacza się jako najmniejszą liczbę  $k$ , spełniającą nierówność (5).

A więc do wyznaczenia wartości szeregu (1) potrzeba napewno  $n$  początkowych wyrazów. Od  $n$ -tego wyrazu począwszy, wyrazy szeregu co do wartości bezwzględnej maleją. Ponieważ szereg (1) jest naprzemienny, więc zachodzi dla niego nierówność

$$|H - S_k| < H_{k+1}, \quad (8)$$

gdzie  $S_k$  jest sumą  $k$  początkowych wyrazów szeregu (podana w [1]). Jeśli teraz wźmie się  $k > n$ , to na mocy (4) otrzyma się

$$|H - S_k| < H_{k+1} < H_k \quad (9)$$

Stąd widać, że na to, by reszta szeregu (1) była dostatecznie mała ze względu na przyjętą dokładność, tzn. aby

$$|H - S_1| < \xi, \quad (10)$$

wystarczy w tym szeregu uwzględnić taką liczbę wyrazów  $l$  ( $l > n$ ), możliwie najmniejszą, aby

$$H_l < \xi \quad (11)$$

Wówczas można przyjąć  $H = S_1$  z dokładnością do  $\xi$ .

b) Problem rozwiązania równania (1) względem  $L$

Niech będą dane wielkości  $R$  i  $H$  oraz wartość  $\xi$ . W celu znalezienia wartości  $L$  należy rozwiązać równanie (1). Szereg (1) staje się więc szeregiem funkcyjnym zmiennej  $L$ .

Począwszy od  $m = 1$ , szereg ten będzie aproksymowany przez wielomian, będący sumą  $m$  początkowych wyrazów tego szeregu, w następujący sposób. Znajduje się pierwiastek  $L_m$  równania

$$H = \sum_{k=1}^m (-1)^{k+1} H_k(L), \quad (12)$$

gdzie  $H_k(L)$  oznacza zależność funkcyjną  $H_k$  od  $L$ . Można to zrobić za pomocą metody Newtona, znajdując pierwiastki równania

$$f(X) = 0, \quad (13)$$

w którym funkcja  $f(X)$  musi być taka, aby  $f'(X)$  i  $f''(X)$  były ciągłe i różne od 0 w pewnym przedziale, zawierającym pierwiastek. Niech  $X^{(1)}$  będzie pewnym punktem z tego przedziału. Kolejne przybliżenia szukanego pierwiastka oblicza się na podstawie wzoru

$$X^{(k)} = X^{(k-1)} - \frac{f(X^{(k-1)})}{f'(X^{(k-1)})}. \quad (14)$$

Proces kończy się, gdy  $|f(X^{(k)})| < \xi$ , i wtedy  $X = X^{(k)}$  jest pierwiastkiem równania (13) z dokładnością do  $\xi$  [1], [3].

Przy rozwiązywaniu równania (12) funkcje  $f$  i  $f'$  określa się

$$f(L) = \sum_{k=1}^m (-1)^{k+1} \frac{L}{(2k-1)! (4k-1) 4k} \left(\frac{L}{2R}\right)^{2k-1} - H \quad (15)$$

$$f'(L) = \sum_{k=1}^m (-1)^{k+1} \frac{1}{(2k-1)! (4k-1) 2} \left(\frac{L}{2R}\right)^{2k-1} \quad (16)$$

Funkcje  $f'(L)$  i  $f''(L)$  jako wielomiany spełniają potrzebne założenia. Za  $L^{(1)}$  podstawia się wartość pierwiastka równania (12) dla  $m=1$ , czyli

$$L^{(1)} = \sqrt{24RH} \quad (17)$$

Następnie, gdy znana jest już wartość pierwiastka  $L_m$  równania (12), rozpatruje się szereg

$$\sum_{k=1}^{\infty} (-1)^{k+1} H_k(L_m), \quad (18)$$

gdzie  $H_k(L_m)$  oznacza wartość wyrazu  $H_k$  dla  $L = L_m$ . Z rozważań znajdujących się w punkcie 2a) wynika, że dla  $L = L_m$  istnieje taka najmniejsza liczba naturalna  $n$ , że ciąg wyrazów  $\{H_k(L_m)\}_{k \geq n}$  jest malejący. Jeśli  $n > m$ , to szereg (1) aproksymuje się przez  $m+1$  pierwszych wyrazów i postępuje podobnie. Jeśli zaś  $n \leq m$ , to z szeregiem (18) postępuje się jak w punkcie 2a), tzn. bierze się pod uwagę taką ilość  $l$  ( $l \geq n$ ) początkowych wyrazów tego szeregu, możliwie najmniejszą, że

$$H_l(L_m) < \varepsilon \quad (19)$$

Wówczas, jeśli  $l > m$ , to postępowanie zaczyna się dla  $m+1$ ; natomiast, gdy  $l \leq m$ , to analogicznie do nierówności (10):

$$|H - S_1(L_m)| < \varepsilon \quad (20)$$

Stąd wynika, że

$$\sum_{k=1}^{\infty} (-1)^{k+1} H_k(L_m) = H = S_1(L_m) \quad (21)$$

z dokładnością do  $\varepsilon$ . A więc aproksymację szeregu (1) przez  $m$  pierwszych wyrazów można uznać za wystarczającą. Pierwiastkiem równania (1) będzie  $L = L_m$ .

W przypadku tym funkcja zmiennej  $L$  z równania (1), będąca szeregiem nieskończonym, była aproksymowana  $m$  razy w punktach  $L_1, \dots, L_m$  w podany wyżej sposób, przez wielomiany. Każda taka  $i$ -ta aproksymacja ( $i=1, \dots, m$ ) sprowadza się do aproksymowania szeregu liczbowego zbieżnego (w punkcie  $L=L_1$ ), jak w pierwszym przypadku.

### 3. Wyniki przeprowadzonej analizy

Zostało zrealizowanych 5 programów, obliczających niektóre wielkości klotoidy, na podstawie danych dwóch wielkości przy zadanej (zawsze tej samej) dokładności obliczeń  $\varepsilon$ , a między innymi w omówionych tutaj przypadkach w punkcie 2a) i b), przy czym w równaniach klotoidy zawsze występowały szeregi nieskończone. Programy te były opracowane pod kątem określenia liczby początkowych wyrazów szeregu nieskończonego zbieżnego, mających istotny wpływ na jego wartość.

W przypadkach omówionych w punkcie 2a) i b) przy rzędzie dokładności  $\varepsilon = 0.000001$ , liczba początkowych wyrazów szeregu (1) nie przekraczała 5. Przy tym brano by pod uwagę różne wartości danych, niekiedy dosyć skrajnie i nie stosowane w praktyce. Dla wartości danych najbardziej typowych praktycznie liczba ta była  $\leq 3$ . Gdyby wyniki, dotyczące liczby tych wyrazów wykorzystać do odpowiednich programów, w których szereg nieskończony (1) byłby zastąpiony przez aproksymują-



cy go wielomian, będący sumą 3 pierwszych wyrazów szeregu, to czas trwania obliczeń byłby krótszy. I tak na przykład dla 6 zestawów danych w przypadku omówionym w punkcie 2a) czas trwania programu skraca się z 9.60 s do 8.86 s, natomiast w przypadku omówionym w punkcie 2b) z 14.42 s do 12.44 s. (Programy były realizowane na maszynie RLAD-32). Aby jednak móc sobie pozwolić na takie skrócenie, trzeba znać z góry tę liczbę. Aby ją znaleźć, trzeba wykonać serię odpowiednich obliczeń pomocniczym programem. Ma to sens wówczas, gdy zestawów danych do obliczeń szukanych wielkości jest dużo. Jeśli natomiast ma się do czynienia z niewielką liczbą tych zestawów, szukane wielkości najlepiej obliczyć, stosując opisaną analizę. Zatem analizę taką warto przeprowadzić zarówno wtedy, gdy liczba zestawów danych jest niewielka, jak i wtedy, gdy tych obliczeń trzeba wykonać dużo. Bowiem dokładność obliczeń i poprawność wyników są najistotniejsze.

Podana metoda, w zależności od zadanej dokładności, wyznacza właściwą liczbę początkowych wyrazów szeregu nieskończonego zbieżnego, możliwie najmniejszą, których suma aproksymuje go. Można powiedzieć, że w tym sensie, z punktu widzenia efektywności obliczeń i czasu ich trwania, proces obliczeniowy jest zoptymalizowany. I to jest główną zaletą opisaną wyżej metody.

#### Literatura

- [1] FICHTENHOLZ G.M.: Rachunek różniczkowy i całkowy, tom I i II. Warszawa: PWN 1972
- [2] LIPIŃSKI M.: Klotoida. Geodezja gospodarcza, tom VI. Warszawa: PPWK 1956
- [3] RALSTON A.: Wstęp do analizy numerycznej. Warszawa: PWN 1971.



## z cyklu: Programowanie obrabiarek

mgr Lidia JESIONOWSKA

Instytut Maszyn Matematycznych

Interaktywne systemy wspomagające wytwarzanie programów  
dla obrabiarek sterowanych numerycznie (OSN)

## 1. Wstęp

Wprowadzenie obrabiarek sterowanych numerycznie (OSN) do procesu produkcyjnego było pierwszym i niezbędnym etapem pozwalającym zautomatyzować proces koncepcja-produkcja.

Obrabiarki sterowane numerycznie stosujemy najczęściej w wypadku wytwarzania małych serii produkcyjnych lub detali o skomplikowanych kształtach, których proces wytwarzania składa się z wieloetapowej obróbki na różnych stanowiskach obróbczych. Obrabiarki sterowane są za pomocą zestawów instrukcji, definiujących zarówno tor ruchu obrabiarek jak i różne niezbędne dane technologiczne np. szybkość obrotu, szybkość posuwu itd. Instrukcje te w postaci numerycznej mogą być przekazywane obrabiarce bezpośrednio z komputera, gdzie zapamiętany jest program (systemy DNC) lub za pośrednictwem nośników informacji np. taśm perforowanych lub taśm magnetycznych (systemy CNC).

Zastosowanie obrabiarek sterowanych numerycznie pociągnęło za sobą rozwój specjalizowanych języków programowania ułatwiających definiowanie opisów geometrycznych i detali technologicznych. Języki te nazywamy językami sterowania numerycznego: APT, IFAPT, EXAPT itd.

Klasyczny model przygotowania programu obróbki części obejmuje kilka etapów pracy.

W pierwszym etapie, na podstawie rysunków i danych technologicznych tworzony jest źródłowy program w języku programowania np. APT. W następnej kolejności program przetwarzany jest przez procesor tego języka. W najlepszym wypadku, do otrzymania bezbłędnego programu potrzebnych jest od 5 do 10 przejść przez procesor APT. O ile program nie zawiera już błędów formalnych, technologicznych i geometrycznych tworzony jest zbiór położeń narzędzia zwany zbiorem CLDATA. W następnym etapie na podstawie wytworzonej taśmy zbioru CLDATA tworzy się np. na ploterze rysunek kontrolny wytwarzanego detalu. Jeżeli otrzymany rysunek nie jest poprawny program wymaga dalszej weryfikacji oraz ponownego przepuszczenia przez procesor APT, w celu otrzymania poprawnego zbioru CLDATA. W wypadku, gdy rysunek otrzymany na ploterze jest poprawny, program nie wymaga korekty i można przejść do post-przetwarzania, w wyniku którego otrzymujemy tasiemkę sterującą dla konkretnej obrabiarki. Otrzymana tasiemka sterująca jest przenoszona na obrabiarkę, gdzie wykonywana jest próbna obróbka np. w drewnie. Jeżeli w tym etapie nie wykryjemy żadnych błędów możemy wykonać obróbkę w metalu.

Przy skomplikowanych kształtach detali przeznaczonych do obróbki programista napotyka na wiele kłopotów związanych z uruchomieniem i weryfikacją programów sterujących. Dlatego też ważnym krokiem w kierunku dalszej automatyzacji procesu wytwarzania detali jest zaprojektowanie odpowiedniego systemu wspomagającego wytwarzanie programów sterujących dla OSN. Systemy te mają na celu skrócenie i ułatwienie procesu tworzenia programów. W związku z tym powinny one dawać użytkownikowi różnorodne możliwości: kontroli przebiegu programu, manipulacji zbiorami, modyfikacji geometrii, wprowadzania danych technologicznych, wydawania zbiorów, analizy syntaktycznej rozkazów i ich korekty. Przetwarzanie, z uwzględnieniem wyżej wymienionych czynników jest możliwe jedynie w wypadku zaprojektowania odpowiedniego systemu, zawierającego elementy komunikacji człowiek - maszyna. Najprostszym przykładem interakcyjnego systemu wspomagającego jest program działający na zasadzie "edytora tekstowego". Daje on użytkownikowi możliwości bieżącej kontroli przebiegu programu, manipulacji zbiorami danych oraz korekty każdego wykrytego błędu. Nawet w skrajnym wypadku klody praca systemu zostaje przerwana, programista powinien uzyskać pełną informację wyjściową pozwalającą

jącą zlokalizować, zidentyfikować i skorygować błąd. Jednakże najłepiej w tego rodzaju systemach sprawdzają się środki grafiki komputerowej, zarówno ze względu na ich możliwości komunikacyjne, jak i wspomagające.

Z punktu widzenia użytkownika system komunikacji graficznej zapewnia mu:

- łatwy i bezpośredni sposób wprowadzania informacji graficznej i alfanumerycznej do systemu,
- przedstawienie informacji w postaci najbardziej komunikatywnej w celu łatwej i bezpośredniej jej analizy oraz zwrotnego dynamicznego oddziaływania na nią,
- łatwy i bezpośredni sposób współpracy z systemem, zazwyczaj w języku naturalnym najbardziej zrozumiałym przez użytkownika.

Graficzny system interakcyjnego wspomagania wytwarzania programów sterujących powinien być zorganizowany jako niezależny moduł wykonujący w trybie konwersacyjnym, oprócz czysto zobrazowujących czynności, modyfikacje geometryczne, wprowadzanie danych technologicznych, wydawanie (wydruk) oraz analizę syntaktyczną instrukcji.

W systemach graficznych wspomagających wytwarzanie taśm sterujących dla OSN, ze względu na sposób definiowania geometrii części, możemy wyróżnić dwa typy systemów.

- W systemach pierwszego typu wejściowe dane geometryczne opisujemy podstawowymi elementami geometrycznymi tzn. punkt, okrąg, wektor... Aby otrzymać standard CLDATA musimy przejść przez cały translator systemu. Natomiast w systemach drugiego typu na wejściu podajemy makro-elementy zawierające oprócz opisu geometrii detalu odpowiednie opisy CLDATA. W tych systemach przez scalanie i modyfikacje geometrycznych i technologicznych opisów bibliotecznych uprzednio utworzonych detali tworzy się nowe opisy części przeznaczonych do obróbki.

Rozwój tych systemów jest poważnym krokiem naprzód w kierunku dalszej automatyzacji procesu wytwarzania taśm sterujących dla OSN.

Odrębną klasę będą stanowiły systemy wspomagające wytwarzanie programów sterujących dla bardzo skomplikowanych kształtów geometrycznych np. stosowanych w przemyśle lotniczym. Wejściem do tego typu systemów są algorytmy numeryczne wykonujące dodatkowe obliczenia potrzebne do zaprojektowania kształtu wyrobu.

Dwie pierwsze klasy systemów będą dokładniej omówione w punkcie 2 i 3 artykułu, natomiast trzecią klasę systemów omówimy nieco szerzej w zakończeniu.

## 2. Systemy działające na zasadzie "edytora tekstowego"

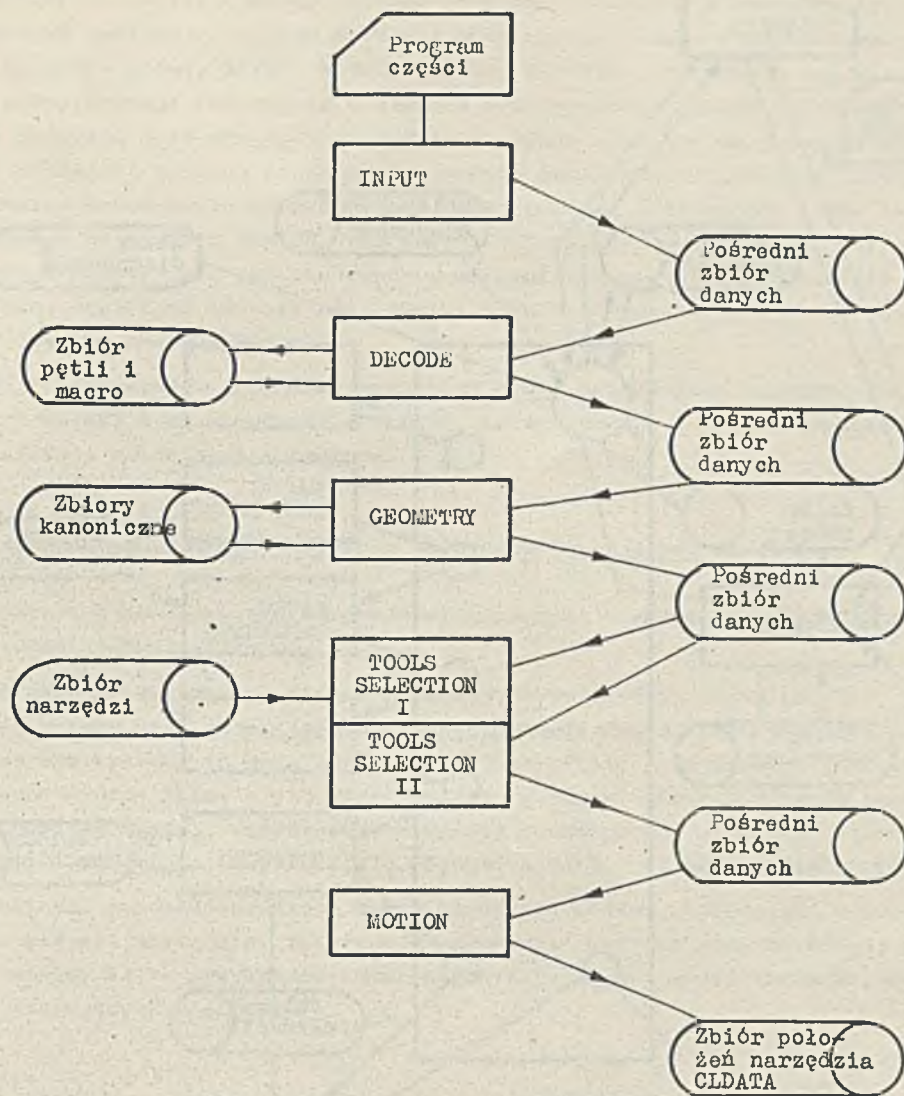
Jak już wspomniano we wstępie, system działający na zasadzie "edytora tekstowego" jest najprostszym przykładem konwersacyjnego systemu wspomagającego wytwarzanie programów sterujących dla OSN. Systemy tego typu dają użytkownikowi możliwość bieżącej kontroli przebiegu programów oraz korekty wszystkich błędów wykrytych przez procesor.

Typowym przykładem takiego systemu jest interakcyjna wersja NELAPT, opracowana w National Engineering Laboratory (East Kilbride, Glasgow, Wielka Brytania) na maszynie UNIVAC 1108. Interakcyjny NELAPT jest programem komputerowym, wytwarzającym taśmki sterujące dla OSN.

Ażby dostrzec różnice między wsadową a interakcyjną wersją NELAPT-u konieczne jest zrozumienie zasad działania oraz metod pracy systemu NELAPT.

### • Wsadowa wersja systemu NELAPT

W systemie wsadowego przetwarzania procesor NELAPT-u zawiera pięć rozdzielnych modułów. Każdy z nich przetwarza kompletny fragment programu obróbki części zaczynając się od instrukcji PARNO a kończąc się na instrukcji FINI. Rys. 1 przedstawia organizację zbiorów oraz metody pracy tej wersji systemu. System zawiera pięć podstawowych modułów INPUT, DECODE, GEOMETRY, TOOLS, SELECTION, MOTION. Moduł INPUT czyta z wejścia jedną linię (kartę) programu, wyszukuje podstawowe elementy takie, jak przerywniki i słaszo, a następnie przetwarza je w ciąg słów komputerowych i zapisuje w przejściowym zbiorze danych, który następnie jest czytany przez moduł DECODE. Po przejściu

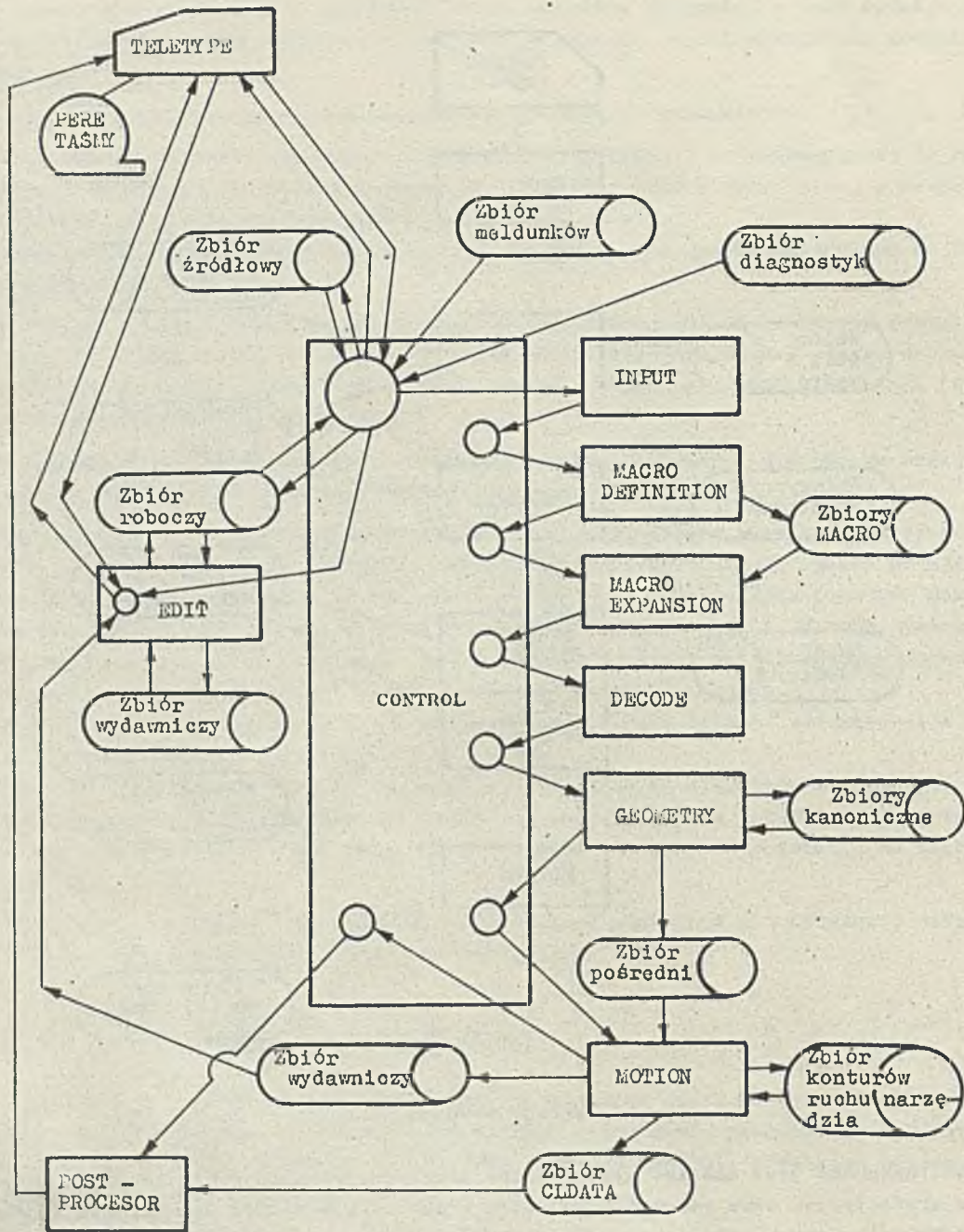


Rys. 1. Wersja wsadowa NELAPT

przez ten moduł dokonana jest analiza syntaktyczna programu. Ponadto w module DECODE wykonywane są obliczenia arytmetyczne oraz rozwijane są pętle i instrukcje MACRO. Standardowe wyjście z modułu DECODE jest zbiorem rekordów, przy czym każdy rekord zawiera jedną logiczną operację programu. Jeżeli w module DECODE będą wykryte błędy - przetwarzanie zostaje zakończone, w przeciwnym wypadku utworzony zbiór jest zbiorem wejściowym dla modułu GEOMETRY, w którym wykonywane są niezbędne obliczenia geometryczne oraz wytwarzane są opisy geometryczne elementów w postaci kanonicznej. Jeżeli moduł GEOMETRY wykryje błąd - praca systemu zostaje zakończona. Moduł TOOLS SELECTION uzupełnia zbiór wyjściowy z modułu GEOMETRY niezbędnymi informacjami o narzędziach, które pobiera z niezależnego pozasystemowego zbioru TOOL LIBRARY. Moduł MOTION wykonuje obliczenia opisujące ruch narzędzia i tworzy zbiór zwany CLDATA, który następnie jest zbiorem wejściowym dla odpowiedniego post-procesora.

● Konwersacyjna wersja systemu NELAPT

Konwersacyjny system NELAPT korzysta z tych samych modułów co wersja wsadowa. Podstawowe różnice zachodzą w sposobie organizacji przetwarzania. Na rys. 2 przedstawiono ścieżki komunikacyjne w interaktywnej wersji NELAPT-u. Jedyne zbiory sekwencyjne, które zachowano w tej wersji to te,



Rys. 2. Konwersacyjna wersja systemu NELAPT

które znajdują się między modułami GEOMETRY i MOTION oraz zbiór CLDATA. Ponadto system rozszerzono o dodatkowy moduł CONTROL sterujący pracą całego systemu. Moduł ten zawiera obszary buforowe PAO, które używane są do przekazywania informacji między poszczególnymi modułami oraz zbiorami zewnętrznymi. Dodatkowo, oprócz modułu CONTROL dodano moduł wydawniczy EDIT oraz z modułu DECODE wydzielono dwa podmoduły MACRO DEFINITION i MACRO EXPANSION. Poza tym system uzupełniony jest dwoma zbiorami o swobodnym dostępie. Zbiory te dotyczą diagnostyki i przekazywania meldunków, a przechowywane są na bębnie maszyny UNIVAC 1108.

System konwersacyjny spełnia trzy podstawowe funkcje: zarządza zbiorami danych, przetwarza program części oraz wyprowadza go na zewnątrz. Moduł CONTROL przydziela zbiory, czyta rozkazy programu źródłowego prosto z dialogowego urządzenia wejściowego albo z uprzednio utworzonego zbioru.

Ponadto moduł CONTROL zapamiętuje skompletowany program źródłowy w nowym zbiorze, a w wypadku wprowadzania poprawek zastępuje program źródłowy nową wersją. Rozkazy pochodzące z dowolnego źródła są analizowane przez moduły INPUT, MACRO, DECODE, GEOMETRY. Wykryte błędy są sygnalizowane do modułu CONTROL, który drukuje informację o błędach oraz opuszcza błędne informacje. Jeżeli źródłem wprowadzania programu jest urządzenie wejściowe system oczekuje na wprowadzenie poprawnej instrukcji. Jeżeli natomiast źródłem programu jest zbiór programowy sterowanie jest przekazywane do modułu EDIT. Sygnałem zakończenia interakcyjnej fazy programu jest rozkaz FINI. Faza przetwarzania zbioru uzyskanego na wyjściu MOTION oraz zbioru CLDATA może odbywać się przy współdziałaniu człowieka. Program źródłowy jest przechowywany w maszynie od momentu wprowadzania go do systemu, dlatego też w każdej chwili na żądanie użytkownika może być wyprowadzony na zewnątrz. Rys. 3 pokazuje uproszczony diagram pracy systemu.

Opisana powyżej interakcyjna wersja systemu może być z powodzeniem zaadaptowana we wszystkich systemach opartych na APT i na standardzie CLDATA, ze względu na to, że organizacja modułów przetwarzania i organizacja zbiorów jest podobna.

### 3. Systemy graficzno wspomagające wytwarzanie programów sterujących dla OSN

W tego typu systemach do komputera wprowadzamy geometrię przedmiotu, a następnie uzupełniamy ją niezbędnymi danymi technologicznymi.

Zazwyczaj interakcyjne systemy graficzne zawierają dwie części: część geometryczną pozwalającą uzyskać rysunek wyrobu oraz część technologiczną pozwalającą operatorowi określić dane do obróbki części. Poza tym systemy te dają użytkownikowi możliwość modyfikacji i wprowadzania poprawek. Definicje geometryczne określa się przez punkty, styczne, przekształcenia geometryczne (przesunięcia, obroty, zmiany skali). Zastosowane elementy geometryczne wykreśla się na ekranie urządzenia graficznego, a następnie uzupełnia się różnymi niezbędnymi parametrami numerycznymi.

Część technologiczna pozwala określić: drogę narzędzia wzdłuż powierzchni uprzednio utworzonych, jak również kształt narzędzia, tolerancję szybkości oraz maszynę, na której przedmiot będzie obrabiany. Omówimy kilka przykładów interakcyjnych systemów graficznych wspomagających tworzenie programów sterujących dla OSN.

#### • System WZL

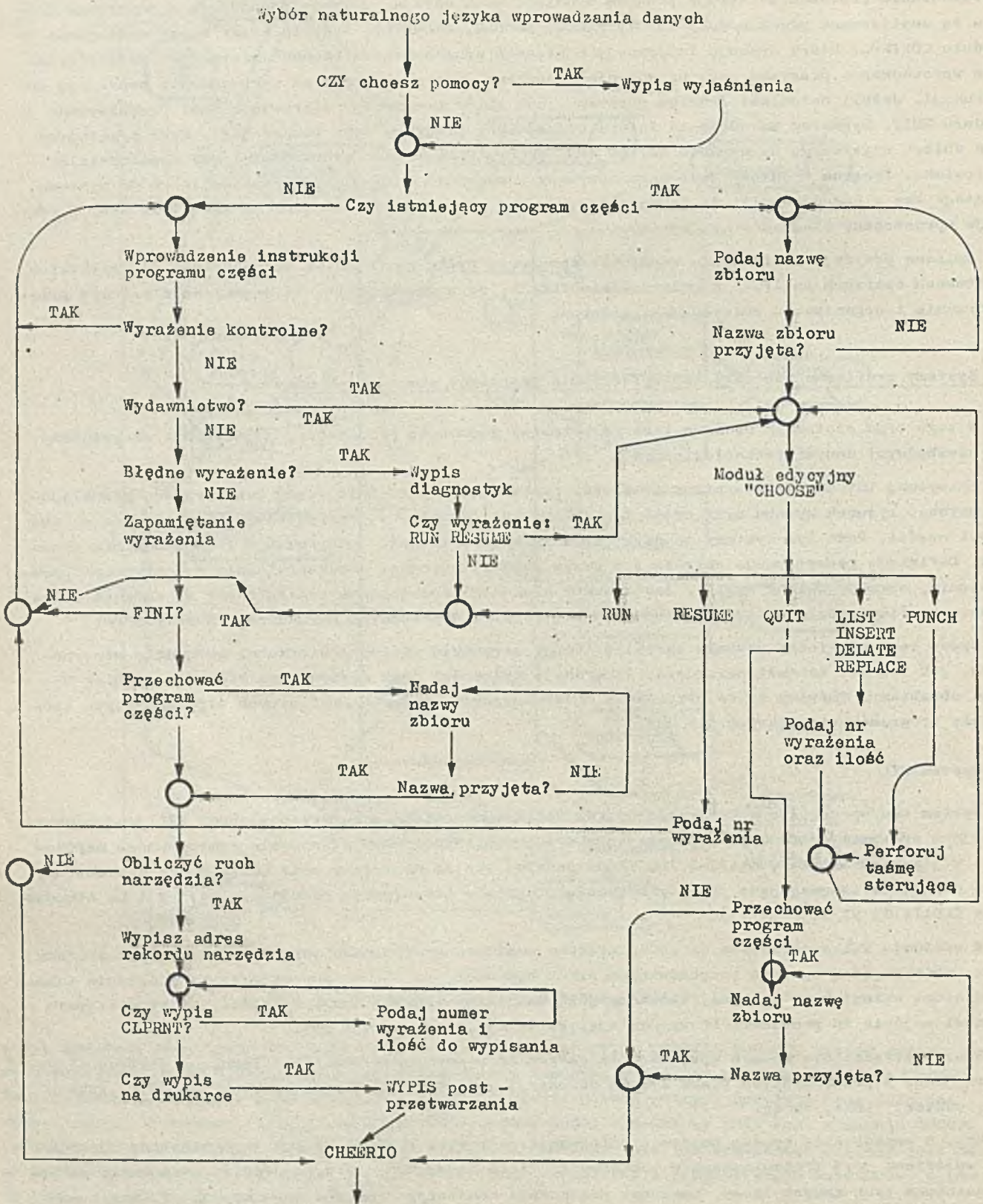
System ten opracowano w Laboratorium für Werkzeugmaschinen und Betriebslehre. WZL jest konwersyjnym systemem graficznym wspomagającym wytwarzanie programów sterowania numerycznego napisanych w językach EXAPT-1, EXAPT-2. Omówiony poniżej system dialogowy może być również zastosowany jako narzędzie wspomagające przy wytwarzaniu programów sterujących bazujących na APT i na standardzie CLDATA TC 97/SC5/WG1/139.

W systemie WZL programista, za pośrednictwem monitora graficznego, odpowiada na pytania systemu oraz wybiera jedną spośród proponowanych przez komputer odpowiedzi. Na podstawie takiego dialogu człowieka z maszyną, tworzony jest kompletny program obróbki części, który następnie stanowi wejście do procesora NC wytwarzającego taśmę sterującą dla OSN.

Rys. 4 przedstawia sposób wprowadzania danych za pomocą urządzenia graficznego. Użytkownik wybiera jedną z proponowanych przez system definicji geometrycznych podstawowych elementów takich jak: wektor, linia, okrąg.

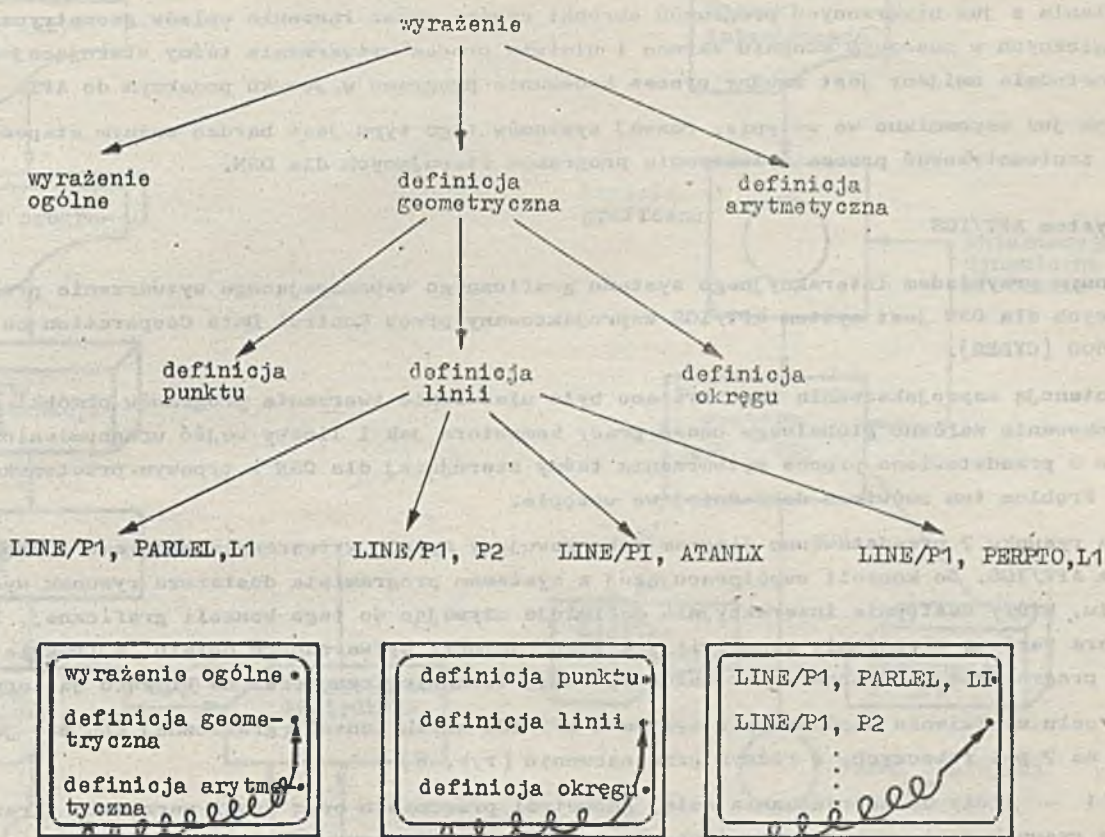
Rys. 5 przedstawia sposób kontroli informacji za pomocą systemu WZL. Z jednej strony informacja wejściowa jest transmitowana z monitora graficznego do komputera, z drugiej natomiast, za pośrednictwem tych samych łączy, komputer przekazuje monitorowi graficznemu informacje, które sam wytworzył.

System WZL daje możliwości symulacji obróbki. Użytkownik za pomocą pióra świetlnego może wybrać dowolny element rysunku wykreślonego na monitorze ekranowym oraz odpowiedni fragment opisu technologicznego. Następnie może wprowadzić pewne modyfikacje zarówno do opisu geometrycznego, jak i technologicznego części.

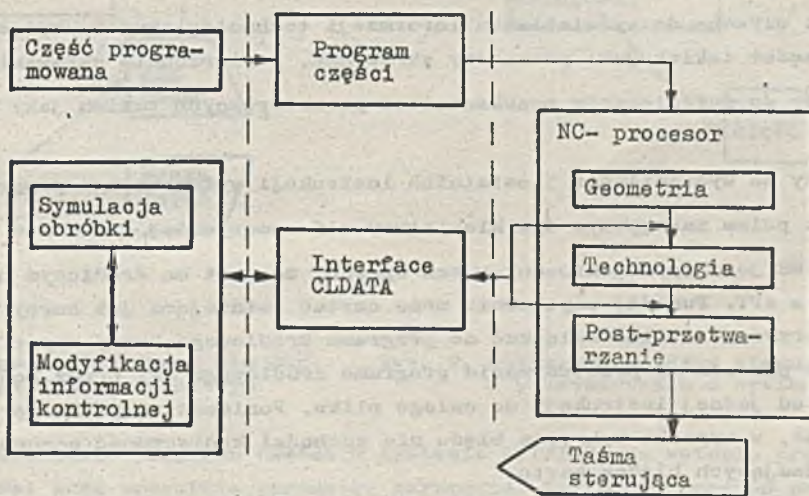


Rys. 3. Diagram pracy systemu NELAP1 w wersji konwersacyjnej





Rys. 4.



Rys. 5.

System opracowany w WZL przechowuje łącznie: informacje geometryczne i technologiczne oraz informacje o organizacji obróbki dla obiektów geometrycznych opisanych płaszczyznami i powierzchniami cylindrycznymi. Przy opracowaniu programu nowej części programista może korzystać z danych uprzednio wytworzonych, które są dostępne dla użytkownika. Zestandaryzowane elementy mogą być wybierane, a następnie łączone ze sobą za pomocą instrukcji klawiatury alfanumerycznej i funkcyjnej. Ponadto przez wprowadzenie odpowiednich danych alfanumerycznych można uzyskać dokładne umiejscowienie na ekranie konsoli graficznej poszczególnych części składających się na opis geometryczny

bądź technologiczny wytwarzanego detalu. Na podstawie opisu części, wytwarza się automatycznie EXAPT-owski program, który następnie jest przetwarzany w sposób uprzednio opisany. Możliwość korzystania z już utworzonych programów obróbki części przez łączenie opisów geometrycznych i technologicznych w znacznym stopniu skraca i ułatwia proces wytwarzania taśmy sterującej dla OSN. W tej metodzie omijany jest znużony proces kodowania programu w języku podobnym do APT.

Jak już wspomniano we wstępie, rozwój systemów tego typu jest bardzo ważnym etapem pozwalającym zautomatyzować proces wytwarzania programów sterujących dla OSN.

#### ● System APT/IGS

Innym przykładem interakcyjnego systemu graficznego wspomagającego wytwarzanie programów sterujących dla OSN jest system APT/IGS zaprojektowany przez Control Data Cooperation na maszynie CDC 600 (CYBER).

Intencją zaprojektowania tego systemu było ułatwienie tworzenia programów obróbki części oraz zredukowanie zarówno globalnego czasu pracy komputera jak i liczby wejść uruchomionych. Na rysunku 6 przedstawiono proces wytwarzania taśmy sterującej dla OSN w typowym przetwarzaniu wsadowym. Problem ten omówiono dokładniej we wstępie.

Na rysunku 7 przedstawiono diagram zobrazujący sposób wytwarzania programu obróbki części przez APT/IGS. Do konsoli współpracującej z systemem programista dostarcza rysunek wytwarzanego detalu, który następnie interakcyjnie definiuje używając do tego konsoli graficznej. Proces ten zawiera zarówno określenie geometrii, jak i technologii wytwarzanego detalu. W trakcie pracy systemu programista może obserwować każdą operację technologiczną oraz na bieżąco ją korygować.

W celu ułatwienia współpracy z systemem APT/IGS ekran konsoli graficznej CDC 27<sup>4</sup> jest podzielony na 7 pól roboczych, o różnym przeznaczeniu (rys. 8):

POLE 1 - służy do zobrazowania całej geometrii przedmiotu oraz drogi narzędzia skrawającego, ponadto rysunek jest uzupełniony niezbędnymi opisami alfanumerycznymi;

POLE 2 oraz POLE 3 - służy do wyświetlania stałych i zmiennych tekstów wyjaśniających użytkownikowi zasady współpracy z systemem APT/IGS;

POLE 4 - jest używane do wyświetlania informacji technologicznych użytecznych w trakcie tworzenia programu części takich jak: parametry skrawania, powierzchnia skrawania itp.;

POLE 5 - służy do definiowania przekształceń geometrycznych takich jak: obrót, przesunięcie i skalowanie;

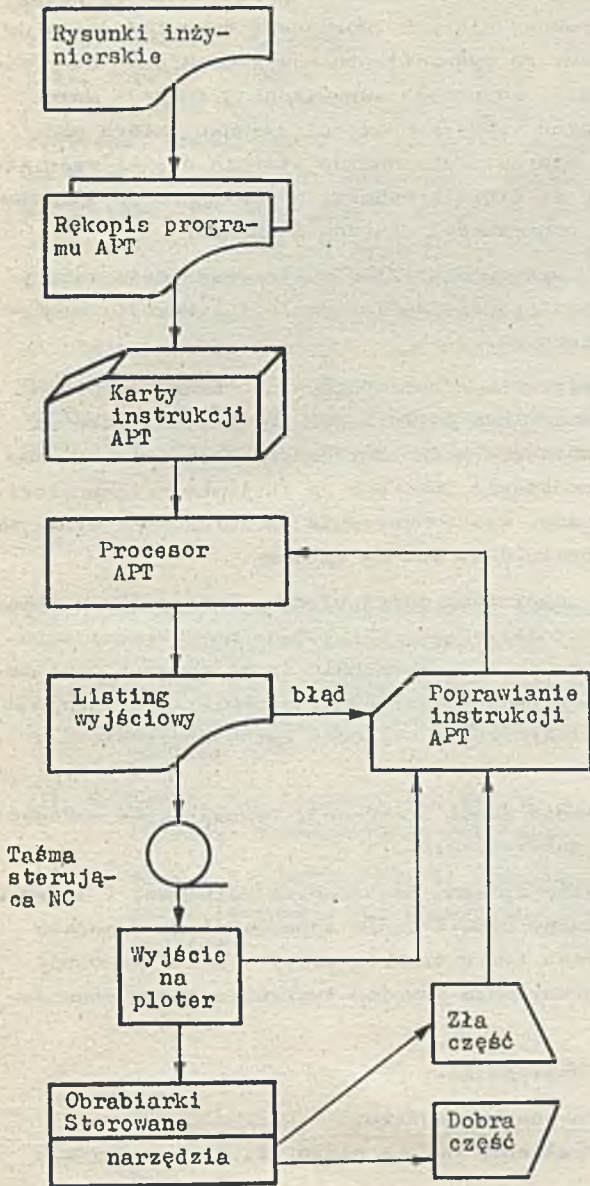
POLE 6 - służy do wyświetlania 5 ostatnich instrukcji wytworzonego programu APT;

POLE 7 - jest polem zastępczym dla klawiatury alfanumerycznej, o ile nie jest ona stosowana.

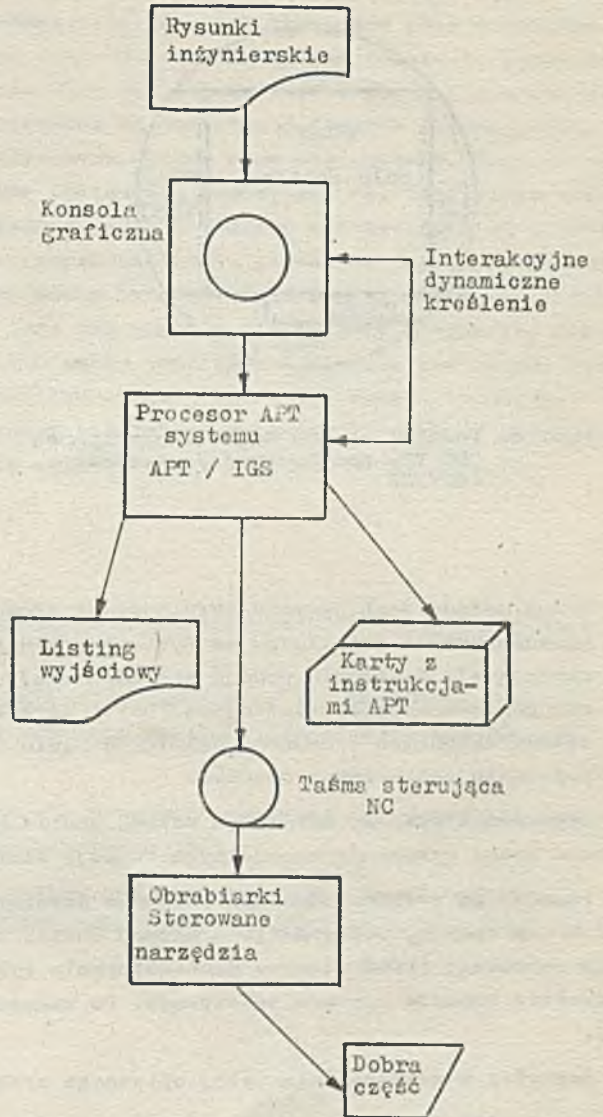
System APT/IGS jest zaprojektowany w ten sposób, że jest on źródłowym pierwotnym wyjściem dla kart programów w APT. Ponadto użytkownik może czytać istniejące już karty programu źródłowego, interakcyjnie tworzyć nowe albo dołączać do programu źródłowego karty wyperforowane poza systemem. Ponadto APT/IGS pozwala na przetwarzanie programu źródłowego APT przez podział na dowolne fragmenty zawierające od jednej instrukcji do całego pliku. Ponieważ wszystkie przetworzone instrukcje są przechowywane, w wypadku wykrycia błędu nie zachodzi konieczność ponownego przetwarzania instrukcji poprzedzających błędną kartę.

Dodatkowo zbiory danych umożliwiają programiście dostęp do informacji użytecznych przy tworzeniu programu obróbki części: np. zbiorów postaci kanonicznych elementów geometrycznych.

APT/IGS ma własności geometryczne takie jak APT-3, przy czym rozszerzony jest o geometrię trójwymiarową (3D). Każda instrukcja technologiczna natychmiast jest zobrazowana na ekranie urządzenia graficznego. Rzeczywistość interakcyjność systemu przejawia się w tym, że jest bezpośredni dostęp do wszystkich informacji przechowywanych w systemie. Ponadto programista uwolniony jest od sekwencyjności w pisaniu programów. Przejawia się to na przykład w tym, że podczas definiowania linii przechodzącej przez dwa punkty, z których jeden nie jest jeszcze zdefiniowany użytkownik może przerwać definicję linii i przejść do zdefiniowania odpowiedniego punktu, a następnie ponownie wrócić do definicji linii.

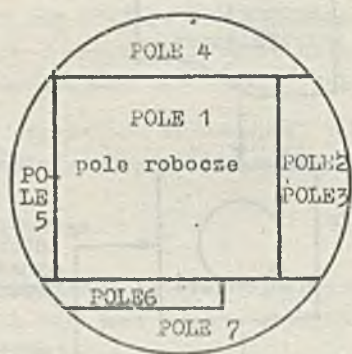


Rys. 6. Wytwarzanie taśmy sterującej dla OSN w typowym przetwarzaniu wsadowym



Rys. 7. Wytwarzanie taśmy sterującej dla OSN w przetwarzaniu w systemie APT IGS

Rozpatrzmy przykład tworzenia programu części w systemie APT/IGS. Na wstępie programista wprowadza identyfikator części oraz wszystkie parametry narzędzia. Następnie spośród proponowanych przez system definicji wybiera np. definicje geometryczne. Załóżmy na przykład, że wybraną definicją była definicja punktu. Punkt może być zdefiniowany wszystkimi możliwymi sposobami dostępnymi w geometrii APT-3. Przypuśćmy, że użytkownik chce zdefiniować punkt jako przecięcia linii. Linie te mogą być wybrane spośród linii wykreślonych na POLU 1 lub też przez zdefiniowanie ich alfanumerycznych opisów. Załóżmy, że linie definiujemy drugim sposobem, wtedy na POLU 1 wyświetlają się dwie linie. W każdym wypadku pojawiają się meldunki, podpowiadające jaka akcja musi być następnie podjęta, przy czym jest to opcjonalne działanie systemu. Jeżeli wykryty zostanie jakikolwiek błąd, informacja ta w formie migotania jest wyświetlana na ekranie. Przed podjęciem jakiegokolwiek dalszego działania musi być wykonana procedura błędowa. W ten sposób korzystając z objaśnień systemu oraz z możliwości wpisywania danych definiuje się cały kształt części. W momencie kiedy



Rys. 8. Podział ekranu konsoli graficznej CDC 274 podłączonej do systemu APT/IGS

kształt części jest zdefiniowany wywołuje się funkcje segmentowania i ograniczania rysunku, za pomocą których tworzy się bardziej czytelny obraz. Po wyświetleniu wszystkich punktów przecięcia, za pomocą odpowiedniej funkcji można wskazać wszystkie części rysunku, które mają być starte. Wskazane do starcia części wyświetlane są linią kreskową, a następnie po sprawdzeniu poprawności usuwane z ekranu.

W systemie APT/IGS zastosowano dwie metody interakcyjnego definiowania instrukcji technologicznych.

- Pierwsza metoda polega na wskazaniu piórem świetlnym powierzchni obróbczych, albo na wpisaniu alfanumerycznym parametrów. Na tej podstawie tworzone są instrukcje technologiczne, które następnie są sprawdzane przez zobrazowanie ich na ekranie.
- Druga metoda jest jeszcze łatwiejsza i stosując ją popełniamy mniej błędów. Polega ona na ulokowaniu krzyża świetlnego za wybraną powierzchnią ograniczającą CS lub na powierzchni ograniczającej. Następnie podaje się informację, że poprzednia powierzchnia ograniczająca jest teraz powierzchnią poprzedzającą (DS) i określa się następną powierzchnię ograniczającą. Obydwie metody są bardzo wydajne i eliminują wiele błędów w określaniu kierunku ruchu narzędzia i w podawaniu powierzchni obróbki.

Użytkownik systemu APT/IGS w każdej chwili może zmienić punkt obserwacji rysunku albo dokonać obrotu przez wykonanie specjalnych funkcji sterowania obrazowaniem.

Ponadto na rysunek może być nałożona kwadratowa siatka opisana oznaczonymi kolumnami i wierszami. W ten sposób, programista w każdej chwili może zażądać wyświetlenia odpowiedniego kwadratu pola roboczego (POLE 1) oraz przoskalowania rysunku. Poza tym w trakcie pracy APT/IGS wykonuje wszystkie funkcje edytora tekstowego. Po zakończeniu pracy może wywołać tworzenie taśmy sterującej.

Korzyści z zastosowania wyżej opisanego systemu są następujące:

- system APT/IGS eliminuje tabulogramy i karty perforowane na wejściu,
- eliminuje kreślenia "of-line" rysunków, to znaczy kreślenia np. na ploterze, które odbywają się poza systemem,
- znacznie skraca czas programowania części do obróbki,
- zmniejsza liczbę wejść na maszynę oraz umożliwia tworzenie wydajniejszych programów części,
- daje duże możliwości optymalizacji i korekty istniejących już programów obróbki detalu.

#### 4. Podsumowanie

Przeprowadzając analizę systemów przedstawionych w tym artykule zwracano szczególną uwagę na problem w jakim miejscu i w jaki sposób należy przeprowadzać kontrolę poprawności programów sterujących, a mianowicie: po zdefiniowaniu geometrii, wytworzeniu zbioru CLDATA i po uzyskaniu taśmy sterującej OSN. We wszystkich opisanych wyżej systemach funkcje graficzne, o których wspomniano we wstępie, nie są całkowicie spełnione, ponieważ systemy te do komunikacji człowiek - maszyna wymagają współpracy wykwalifikowanego operatora. Dlatego też rozbudowa graficznych systemów wspomagających powinna iść w kierunku tworzenia innych metod konwersacji tzn. w kierunku tworzenia języków zbliżonych do języka naturalnego, jakim posługują się ich użytkownicy. W chwili obecnej operator obsługujący tego rodzaju systemy wspomagania powinien biegle znać instrukcje programowania w języku sterowania numerycznego np. APT. W WZL podjęto próby rozwiązania tego problemu

dając użytkownikowi możliwość składania rysunku wytwarzanego detalu z części, które uprzednio były zaprogramowane. Należy jednak pamiętać, że narzędzie to nie jest użyteczne przy projektowaniu części bardziej złożonych np. części mechanicznych stosowanych w przemyśle lotniczym (profile śmigła, łopatki turbin). Kształty takich obiektów nie dają się przedstawić w postaci geometrycznych elementarnych funkcji. Do rozwiązania problemu potrzebne są dokładne definicje matematyczne, zazwyczaj bezpośrednio wynikające z własności fizycznych, które chce się uzyskać. Dlatego też do tworzenia kształtów tego typu potrzebne są pewne obliczenia numeryczne np. obliczenia wytrzymałościowe. W innych wypadkach powierzchnie wynikające np. z kryteriów estetycznych są określone jedynie przez pewną liczbę punktów, dlatego też do zaprojektowania kształtów tego typu należy dysponować wieloma algorytmami interpolacyjnymi, aby można było określić punkty pośrednie i różne parametry (styczne, koła ściśle styczne itp.). Z tego też względu dobrze jest dysponować dużą liczbą programów, których możliwości są bardziej lub mniej rozległe. Z drugiej zaś strony tworzenie systemów bardzo rozbudowanych i w pełni zautomatyzowanych jest rzeczą bardzo kosztowną i zazwyczaj koszty te są wyższe od przewidywanych. Dlatego też należy zwiększyć wysiłki w celu proponowania odbiorcom systemów modułowych łatwych do adaptowania i prostych w użytkowaniu.

#### Literatura

- [1] CRESTIN J.P.: L'utilisation de terminaux graphiques la mise au point de programmes de commande numerique. Automatisme 1974 t. XIX nr 4
- [2] WILKINSON D.G.: Conversational NELAPT, an interactive version of the NELAPT - processor. W: Computer languages for numerical control: APT/IGS. North-Holland Publishing Company: Amsterdam 1973
- [3] CREMERIUS J.: State of the art in NC graphics. W: Computer languages for numerical control: APT/IGS. North-Holland Publishing Company: Amsterdam 1973
- [4] LACOSTE J.P., ROTHEMBERG R.: Computer aided design and computer aided manufacturing. W: Computer languages for numerical control APT/IGS. North-Holland Publishing Company: Amsterdam 1973




BRANŻOWY OŚRODEK INFORMACJI NAUKOWEJ TECHNICZNEJ I EKONOMICZNEJ  
INSTYTUTU MASZYN MATEMATYCZNYCH  
02-078 Warszawa, ul. Krzywickiego 34, tel. 21-84-41 w. 391


BOINTE udziela informacji  
z zakresu techniki komputerowej

BOINTE wydaje


# informacja ekspresowa

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

# przegląd dokumentacyjny

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

# biuletyn informacyjny

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

Materiały konferencyjne, szkoleniowe, prospekty

BOINTE gromadzi

wydawnictwa zwarte, czasopisma krajowe i zagraniczne, katalogi i prospekty, sprawozdania z prac naukowo-badawczych oraz inne materiały informacyjne

BOINTE wykonuje usługi reprodukcyjne i poligraficzne

fotokopie, mikrofilmy, kserokopie z zakresu posiadanych zbiorów

#### WARUNKI PRENUMERATY

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

Cena prenumeraty rocznej zł 840.

Jednostki gospodarki uspołecznionej, instytucje i organizacje społeczno-polityczne składają zamówienia w miejscowych Oddziałach RSW "Prasa-Książka-Ruch".

Zakłady pracy i instytucje w miejscowościach, w których nie ma Oddziałów RSW oraz prenumeratorzy indywidualni, zamawiają prenumeratę w urzędach pocztowych lub u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę, która jest o 50% droższa od prenumeraty krajowej, przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO nr 1531-71 - w terminie podanym dla prenumeraty krajowej.