WYDZIAŁ GEODEZJI I KARTOGRAFII

ZAKŁAD GEODEZJI I ASTRONOMII GEODEZYJNEJ

# Wprowadzenie do geometrii obliczeniowej

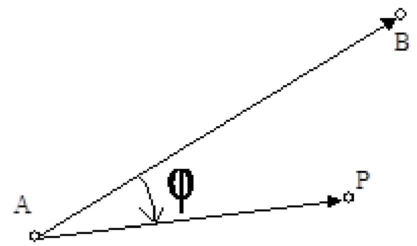
dr hab. inż. Waldemar Izdebski, prof. PW

Warszawa, 2022

# SPIS TREŚCI

# 1 WYZNACZENIE POŁOŻENIA PUNKTU WZGLĘDEM ODCINKA

Zadanie polega na napisaniu programu, który będzie określał położenie punktu względem odcinka tzn. sprawdzenie, po której stronie danego odcinka leży punkt o zadanych współrzędnych XY. Ilustrację zadania przedstawia rys. 1.




Rysunek 1. Ilustracja zadania położenia punktu względem odcinka

## 1.1 ROZWIĄZANIE OPARTE NA ZNAKU WYZNACZNIKA $\boldsymbol{det(A,B,P)}$

Jednym ze sposobów rozwiązania postawionego zadania jest obliczenie wyznacznika postaci:

$$det(A,B,P) = \begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_P & Y_P & 1 \end{vmatrix}$$

Znak wyznacznika $det(A,B,P)$ jest równy znakowi sinusa kąta nachylenia wektora $\overrightarrow{AP}$ do wektora $\overrightarrow{AB}$. Otrzymane wartości wyznacznika maja więc następującą interpretację:

- jeżeli $det(A.B,P) > 0$, to punkt $P$ położony jest po prawej stronie odcinka $AB$,
- jeżeli $det(A.B,P) < 0$, to punkt $P$ położony jest po lewej stronie odcinka $AB$,
- jeżeli $det(A.B,P) = 0$, to punkty $A$, $B$ i $P$ są współliniowe.

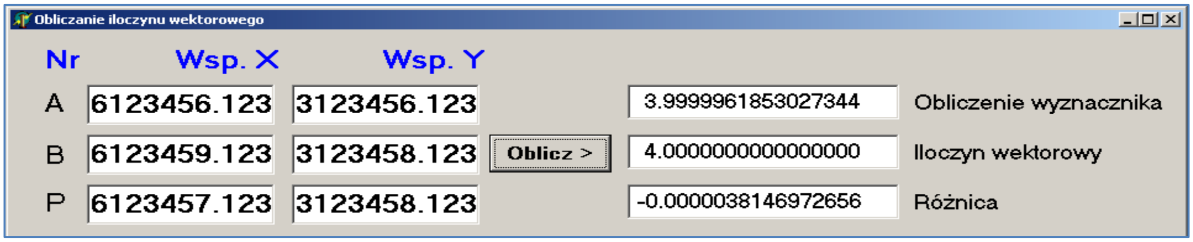
Wyznacznik możemy obliczyć np. metodą Sarussa, która daje następujące rozwinięcie:

$$det(A,B,P) = X_AY_B + X_BY_P + X_PY_A - X_PY_B - X_AY_P - X_BY_A$$

## 1.2 ROZWIĄZANIE OPARTE NA ILOCZYNIE WEKTOROWY

Innym sposobem określenia położenia punktu względem odcinka, bardziej stabilnym pod względem numerycznym, jest wykorzystanie iloczynu wektorowego. Zapisując iloczyn wektorowy do postawionego zadania, pamiętając, że punkty leżą na płaszczyźnie otrzymujemy:

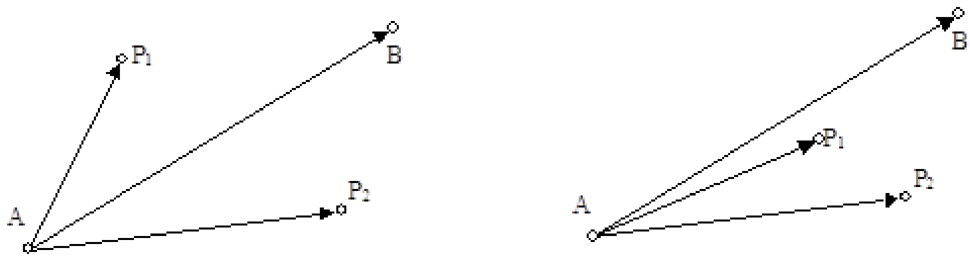
$$\begin{vmatrix} i & j & k \\ \Delta X_{AB} & \Delta Y_{AB} & 0 \\ \Delta X_{AP} & \Delta Y_{AP} & 0 \end{vmatrix} = [0,\ 0,\ \Delta X_{AB}\Delta Y_{AP} - \Delta X_{AP}\Delta Y_{AB}]$$




Rysunek 2. Różnica między dwoma metodami wyznaczenia położenia punktu względem odcinka

Jak wspomniano, drugi sposób obliczeń jest znacznie stabilniejszy numerycznie gdyż w obliczeniach wykonuje się działania na różnicach współrzędnych zamiast na ich pełnych wartościach, które dodatkowo podlegają mnożeniu. Ilustracje otrzymanych wyników przedstawiono na rys. 2. Widzimy, że uzyskana wartość bezwzględna rozbieżności z obydwu metod wynosi w przybliżeniu 0.000004, co w pewnych sytuacjach może powodować uzyskanie błędnego wyniku.

Warto pamiętać, że iloczyn wektorowy $AB \times AP$, ma również interpretację, jako podwójne pole trójkąta utworzonego przez punkty $A$, $B$, $P$.




Rysunek 3. Położenie dwóch punktów względem odcinka

### 1.2.1 Położenie dwóch punktów względem odcinka

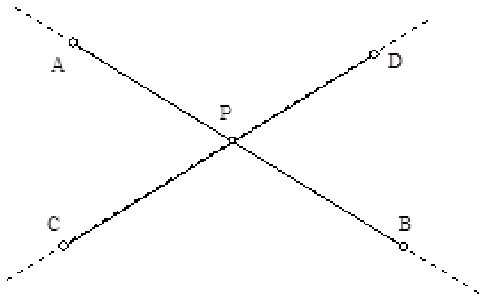
Często w zagadnieniach geometrycznych mamy do zbadania czy dane dwa punkty leżą po tej samej stronie odcinka (rys. 3). Wykorzystując opisany sposób sprawdzenia położenia punktu względem odcinka, wyznaczamy iloczyny skalarne i porównujemy ich znaki. Jeśli znaki są takie same oznacza to, że punkty leżą po tej samej stronie, w przeciwnym wypadku leżą po różnych stronach odcinka.

## 1.3 ĆWICZENIE 1 – WYMAGANIA

1. Program do wyznaczania może być napisany w dowolnym środowisku programistycznym.
2. Program powinien zawierać i wykorzystywać funkcje do sprawdzenia położenia, bazujące na powyższych algorytmach (wyznacznik macierzy, iloczyn wektorowy). Argumentami funkcji są współrzędne 3 punktów, a zwracana wartość jest liczbą rzeczywistą, której znak oznacza położenie punktu.
3. W interfejsie programu powinny być wykorzystanie obie funkcje. Należy zaprezentować oraz zinterpretować zwrócone przez funkcje wyniki (określić, po której stronie odcinka $AB$ znajduje się punkt $P$).
4. W interfejsie programu powinno być zawarte zabezpieczenie na okoliczność wprowadzenia błędnych danych, które powinny mieć wartości liczbowe.
5. Czas na wykonanie ćwiczenia: 2 zajęcia. Program powinien być oddany (zaprezentowany) na do końca drugich zajęć, w przypadku nieoddania wystawiana jest ocena negatywna.

# 2 WYZNACZENIE PUNKTU PRZECIĘCIA DWÓCH ODCINKÓW

Zadanie polega na napisaniu programu, który będzie wyznaczał punkt przecięcia dwóch odcinków, z których każdy jest dany przez współrzędne dwóch punktów. Ilustrację zadania przedstawia rys. 4.




Rysunek 4. Ilustracja zadania przecięcia dwóch prostych określonych dwoma punktami

## 2.1 METODA ROZWIĄZANIA ZADANIA

Z punktu widzenia wzorów rozwiązujących, to zadanie jest szczególnym przypadkiem ogólnego zadania polegającego na wyznaczeniu punktu przecięcia dwóch prostych. Do rozwiązania zadania wykorzystujemy równanie parametryczne prostej, w którym każdy punkt na prostej daje się wyrazić w funkcji punktu początkowego, końcowego i pewnego parametru rzeczywistego $t$. Parametryczne równanie prostej przechodzącej przez punkty $A$ i $B$ ma postać:

$$\begin{aligned} X &= X_A + t * \Delta X_{AB}, \\ Y &= Y_A + t * \Delta Y_{AB}, \end{aligned}$$

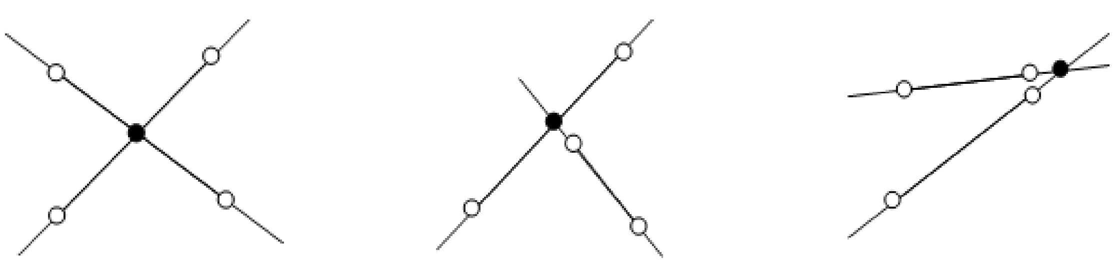
gdzie $t \in (-\infty, \infty)$. Przy czym na uwagę zasługuje fakt, że w punkcie $A$ parametr $t = 0$ natomiast w punkcie $B$ parametr $t = 1$. Rozwiązanie zadania możemy przedstawić następującymi wzorami:

$$\begin{aligned} X_P &= X_A + t_1 \Delta X_{AB}, \\ Y_P &= Y_A + t_1 \Delta Y_{AB} \end{aligned} \quad \text{lub} \quad \begin{aligned} X_P &= X_C + t_2 \Delta X_{CD}, \\ Y_P &= Y_C + t_2 \Delta Y_{CD}, \end{aligned}$$

gdzie:

$$t_1 = \frac{\Delta X_{AC} \Delta Y_{CD} - \Delta Y_{AC} \Delta X_{CD}}{\Delta X_{AB} \Delta Y_{CD} - \Delta Y_{AB} \Delta X_{CD}}, \qquad t_2 = \frac{\Delta X_{AC} \Delta Y_{AB} - \Delta Y_{AC} \Delta X_{AB}}{\Delta X_{AB} \Delta Y_{CD} - \Delta Y_{AB} \Delta X_{CD}}.$$

O ile jednak w ogólnym zadaniu wyznaczenia współrzędnych punktu przecięcia prostych wystarczające jest obliczenie parametrów parametry $t_1$ i $t_2$ a następnie współrzędnych punktu przecięcia, to w zadaniu wyznaczenia współrzędnych punkty przecięcia odcinków należy dodatkowo sprawdzić czy wyznaczony punkt przecięcia należy do obu odcinków (co jest konieczne do uznania go za rozwiązanie).



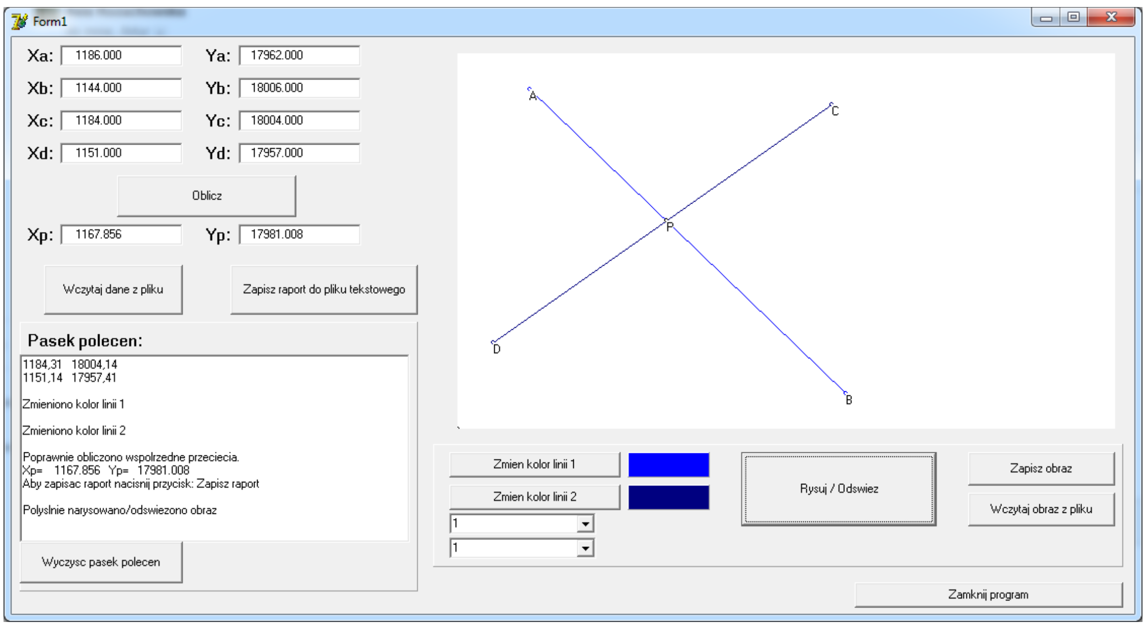
Rysunek 5. Różne położenie punktu przecięcia dwóch prostych w stosunku do odcinków

W celu sprawdzenia przynależności do odcinków możemy wykorzystać parametry $t_1$ i $t_2$, na podstawie których, można stwierdzić czy punkt przecięcia będzie należał do obu przecinanych odcinków. Ma to miejsce, jeżeli: $(0 \leq t_1 \leq 1)$ i $(0 \leq t_2 \leq 1)$.

Przykładowe dane do obliczeń przedstawiono w poniższych tabelach, a przykładowy wygląd programu realizującego wyznaczenie przecięcia przedstawiono na rys. 6.

| Zestaw 1 | | |
|---|---|---|
| Punkt | X | Y |
| A | 1186.00 | 17962.69 |
| B | 1144.74 | 18006.22 |
| C | 1184.31 | 18004.14 |
| D | 1151.14 | 17957.41 |
| P | 1168.210 | 17981.459 |

| Zestaw 2 | | |
|---|---|---|
| Punkt | X | Y |
| A | 5772348.431 | 7501363.570 |
| B | 5772443.825 | 7501363.570 |
| C | 5772452.342 | 7501466.630 |
| D | 5772488.399 | 7501471.740 |
| P | 5772501.971 | 7501529.449 |



Rysunek 6. Przykładowy wygląd programu

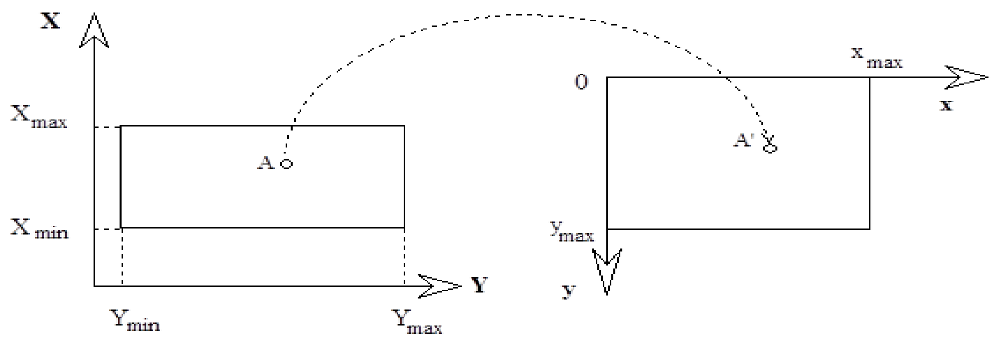
## 2.2 ĆWICZENIE 2 – WYMAGANIA DO PROGRAMU

1. Program może być napisany w dowolnym środowisku programistycznym.
2. Program powinien mieć możliwość wprowadzania danych z klawiatury jak i odczytu danych z pliku tekstowego, a także mieć możliwość zapisania wyników do pliku tekstowego.
3. Program powinien realizować prezentację graficzną analizowanych punktów i odcinków wraz z punktem ich przecięcia:
   a. pierwsza prezentacja powinna być dopasowana do widoczności wszystkich punktów,
   b. punkty na rysunku powinny być opisana swoimi oznaczeniami (A,B,C,D,P),
   c. powinna istnieć możliwość konfigurowania parametrów rysunku, tzn.: kolory, grubości i style linii, widoczność ukrycie oznaczeń punktów.
4. Czas na wykonanie ćwiczenia: 4 zajęcia. Program powinien być oddany na koniec szóstych zajęć, w przypadku nieoddania wystawiana jest ocena negatywna.

# 3 INFORMACJE PRZYDATNE PRZY PREZENTACJI GRAFICZNEJ

W większości środowisk programistycznych dla komponentów graficznych początek układu współrzędnych znajduje się w lewym górnym narożniku komponentu, oś X jest skierowana w prawo, analogicznie do układu współrzędnych matematycznych, natomiast oś Y jest skierowana w dół komponentu. Jest to istotna różnica względem układu współrzędnych matematycznych czy geodezyjnych (rys. 7). Dodatkowo, obszar wyświetlania współrzędnych ograniczony jest prostokątem o długościach boków rysunku $\boldsymbol{x_{max}}, \boldsymbol{y_{max}}$ (określających wielkość elementu).

W celu poprawnego wyświetlania prezentacji punktów o zadanych współrzędnych, rysunek (współrzędne wyświetlanych punktów) należy odpowiednio przeskalować i dopasować do wielkości komponentu, na którym jest tworzony.




Rysunek 7. Przekształcenie współrzędnych terenowych do współrzędnych aplikacji

Odwzorowanie współrzędnych układu terenowego w układ współrzędnych aplikacji można zrealizować za pomocą zależności:

$$x = s_x(Y - Y_{min}),$$

$$y = y_{max} - s_y(X - X_{min})$$

gdzie: $x, y$ – współrzędne analizowanego punktu w oknie aplikacji, $X, Y$ – współrzędne analizowanego punktu w wyjściowym układzie współrzędnych, $X_{min}, Y_{min}$ – minimalne wartości współrzędnych, które mają znaleźć się na tworzonym rysunku, $x_{max}, y_{max}$ – maksymalne współrzędne okna rysowania (wymiary: szerokość i wysokość), $s_x, s_y$ – współczynniki zmiany skali wyznaczone według wzorów:

$$s_x = \frac{x_{max}}{Y_{max} - Y_{min}}, \qquad s_y = \frac{y_{max}}{X_{max} - X_{min}},$$

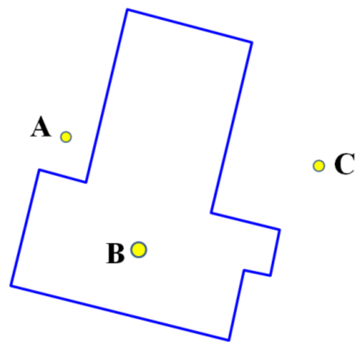
gdzie: $X_{max}, Y_{max}$ – maksymalne współrzędne terenowe.

Układ współrzędnych w komponentach jest układem dyskretnym (współrzędne mogą przyjmować tylko wartości całkowite nieujemne – nie można narysować punktu w połowie lub jednej trzeciej piksela), wynikowe współrzędne $x, y$ w układzie aplikacji, zależności **należy zatem zaokrąglić przed rozpoczęciem rysowania**.

Istotne jest proporcjonalne wyświetlanie rysunku – wskazane jest zastosowanie jednakowego współczynnika skali dla obu współrzędnych: mniejszej z wartości $\boldsymbol{s_x}, \boldsymbol{s_y}$. Skutkiem wyboru większego z nich, byłaby sytuacja, w której część przetwarzanego szkicu w układzie terenowym znalazła się poza obszarem rysowania w aplikacji.

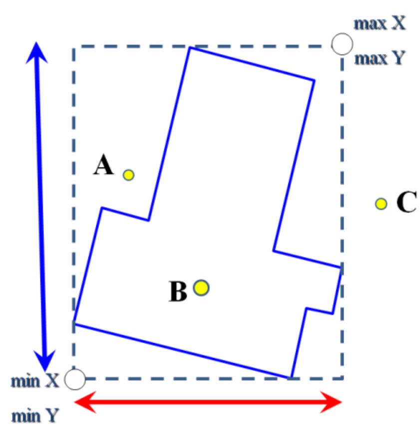
# 4 WYZNACZENIE POŁOŻENIA PUNKTU WZGLĘDEM WIELOKĄTA

Ustalanie położenia punktu względem wielokąta podobnie jak obliczanie współrzędnych punktu przecięcia dwóch odcinków, jest jedną z najczęściej wykonywanych czynności w procesie analizy danych przestrzennych. Celem rozpatrywanego zadania jest ustalenie położenia punktu o zadanych współrzędnych P(x,y) względem wielokąta **Q** określonego ciągiem punktów. Zakładamy przy tym, że wielokąt jest wielokątem zwykłym, co oznacza, że jego boki się nie przecinają.




Rysunek 8. Możliwe przypadki położenia punktu względem wielokąta

Najbardziej znanymi algorytmami sprawdzania położenia punktu względem wielokąta są: algorytm sumy kątów oraz algorytm parzystości. Wymienione algorytmy różnią się między sobą dosyć znacznie. W obydwu przypadkach jednak badanie powinno się rozpoczynać od określenia położenia względem prostokąta ograniczającego. Wystarczy bowiem, że punkt leży poza tym prostokątem aby stwierdzić, że leży również poza wielokątem.




Rysunek 9. Wstępne badanie położenia punktu wewnątrz wielokąta

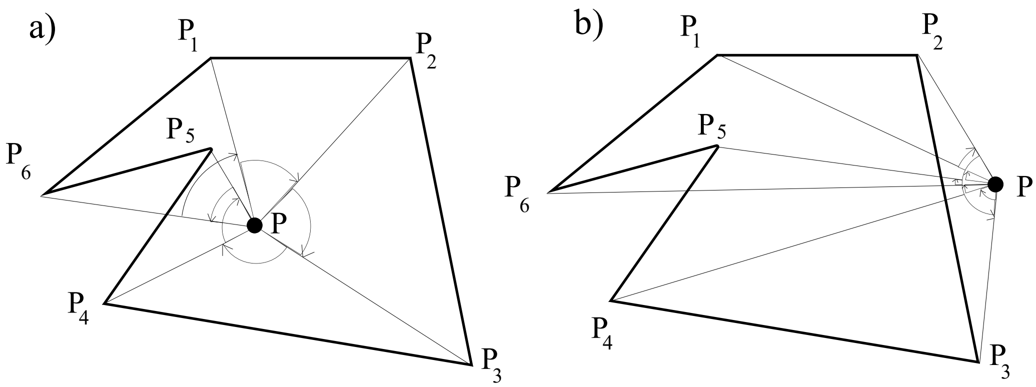
Warunkiem koniecznym, położenia punktu wewnątrz wielokąta jest jego położenie wewnątrz prostokąta ograniczającego, ale nie jest to warunek wystarczający, co widać na przykładzie punktu **A**, przedstawionego na powyższym rysunku.

## 4.1 ALGORYTM SUMY KĄTÓW

Algorytm sumy kątów wykorzystuje, do ustalenia położenia punktu względem wielokąta, sumę kątów pomiędzy półprostymi poprowadzonymi z badanego punktu **P** przez wierzchołki wielokąta **P**$_i$. Obliczaną sumę kątów możemy zapisać jako:

$$S = \sum_{i=1}^{n} \alpha_i$$

gdzie $\alpha_i$ jest kątem między półprostymi **PP**$_i$ a **PP**$_{i+1}$.




Rysunek 10. Testowanie położenia punktu względem wielokąta algorytmem sumy kątów: a) punkt wewnątrz wielokąta; b) punkt na zewnątrz wielokąta

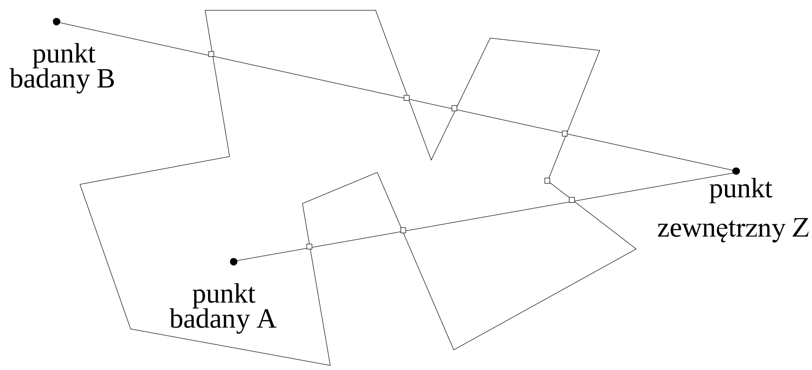
Jeśli punkt znajduje się wewnątrz wielokąta wtedy **S=360°** (rys. 10a) w przeciwnym wypadku **S=0°** (rys. 10b). Oczywiście podane wartości sumy kątów należy sprawdzać w pewnym przedziale dokładności numerycznej **±ε**. Obliczane kąty $\alpha_i$ traktujemy jako dodatnie, jeśli uporządkowanie ramion jest zgodne z ruchem wskazówek zegara i jako ujemne jeśli jest kierunek uporządkowania jest przeciwny do ruchu wskazówek zegara.

## 4.2 ALGORYTM PARZYSTOŚCI

Algorytm parzystości opiera się na założeniu, że jeżeli przyjmiemy dowolny punkt (Z) leżący na zewnątrz wielokąta i poprowadzimy od niego odcinek do punktu badanego (takiego, którego położenie względem wielokąta wyznaczamy) to zachodzą następujące przypadki:

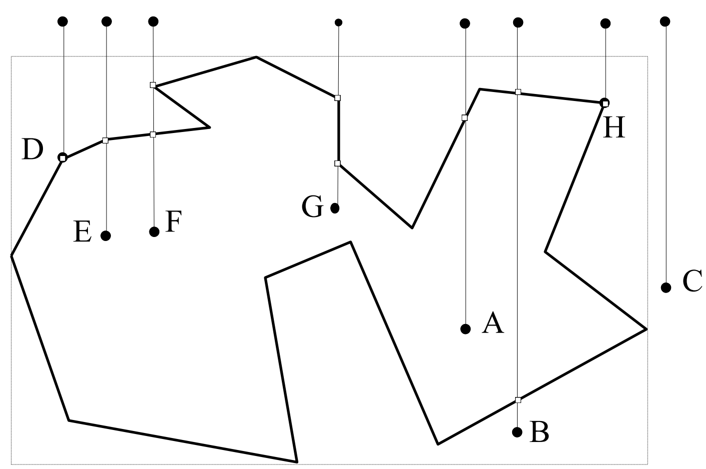
- jeżeli punkt badany (A) leży wewnątrz wielokąta, to wtedy odcinek łączący te punkty przecina boki wielokąta nieparzystą liczbę razy,
- jeżeli punkt badany (B) leży na zewnątrz wielokąta odcinek łączący punkty albo nie przecina żadnego boku, albo przecina je parzystą liczbę boków.

Ilustrację graficzna algorytmu parzystości przedstawiona jest na poniższym rysunku 11.




Rysunek 11. Ogólna zasada działania algorytmu parzystości

Praktyczne zastosowanie algorytmu będzie różniło się od przypadku ogólnego przedstawionego na rys. 11. Zalecane jest, aby badając położenie punktu przyjmować punkt o znanym położeniu (na zewnątrz wielokąta), tak aby odcinek był równoległy do jednej z osi układu współrzędnych, co znacznie skróci proces obliczania przecięć. Przyjmując zasadę, badania liczby przecięć wielokąta z odcinkiem równoległym do osi **X,** zwanym dalej odcinkiem pionowym (określonym punktem zewnętrznym położonym poza prostokątem zawierającym wielokąt i punktem badanym) mamy sytuację przedstawioną na rys. 12. W przypadku wystąpienia nieparzystej liczby przecięć stwierdzamy, że badany punkt leży wewnątrz wielokąta (punkt **A** na rys.12). Jeżeli liczba przecięć będzie parzysta (punkt **B**) lub jeśli przecięcia w ogóle nie wystąpią (punkt **C**) wtedy punkt leży na zewnątrz.

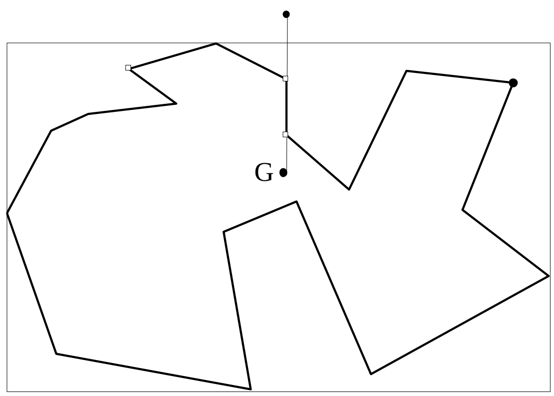



Rysunek 12. Zasada działania algorytmu parzystości z odcinkiem pionowym

Działanie algorytmu jest poprawne dla wszystkich punktów położonych wewnątrz i na zewnątrz wielokąta z wyjątkiem punktów pokrywających się z wierzchołkami wielokąta (punkt **D** i **H**) oraz w przypadku, kiedy odcinek przechodzi przez wierzchołki (punkt **E** i **F**) lub pokrywa się z jednym z boków wielokąta (punkt **G**).

W przypadku pokrywania się badanego punktu z wierzchołkiem wielokąta należy postępowanie identyczne jak w algorytmie sumy kątów, polegające na porównywaniu współrzędnych kolejnych wierzchołków ze współrzędnymi badanego punktu. Przez opisane postępowanie wyeliminujemy wszystkie przypadki analogiczne do punktu **D** i **H**.

Opisany powyżej przypadek jest jednak tylko jednym z przypadków szczególnych występujących w niniejszym algorytmie. Generalnie przypadki takie mają miejsce, kiedy odcinek pionowy poprowadzony przez badany punkt przechodzi przez wierzchołek wielokąta. Pojawia się wtedy problem, czy przecięcia liczyć, jako jednokrotne czy dwukrotne. W przypadku punktu **E,** aby algorytm dał poprawny wynik należy przecięcie potraktować, jako jednokrotne, natomiast w przypadku punktu **F** należy przecięcie traktować, jako dwukrotne. Problem rozwiązujemy przez przyjęcie następującej zasady. Jeżeli oba przecinane odcinki położone są po tej samej stronie odcinka pionowego, wtedy przecięcia traktujemy, jako dwukrotne, w przeciwnym wypadku, jako jednokrotne. Pozostaje jednak jeszcze inny szczególny przypadek, kiedy odcinek pionowy przechodzi przez jeden z boków wielokąta (punkt **G**).

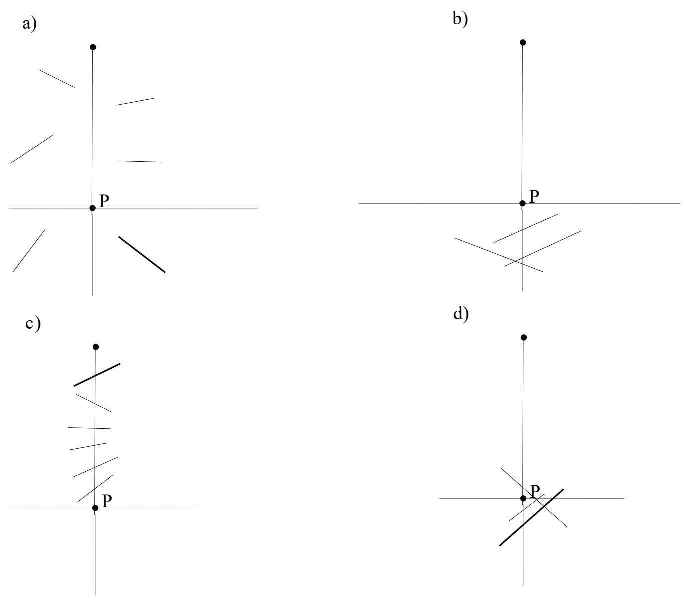



Rysunek 13. Analiza przypadku szczególnego

W takiej sytuacji mamy również, przy punkcie położonym wewnątrz, parzystą liczbę przecięć, a powinniśmy otrzymać liczbę nieparzystą. Rozwiązanie takiego przypadku jest możliwe, jeśli potraktujemy taki odcinek, jako pseudowierzchołek i zastosujemy do niego opisany wyżej sposób ustalania liczby przecięć. Wprowadzone dodatkowe sprawdzenia sprawiają również, że algorytm staje się odporny na przypadek, kiedy badany punkt pokrywa się z wierzchołkiem wielokąta. Jest tak w przypadku, kiedy oba wychodzące z tego wierzchołka boki leżą po różnych stronach odcinka testującego, gdyż wtedy przecięcie traktowane jest, jako jednokrotne. W przypadku, kiedy będzie inaczej algorytm daje zły wynik ponieważ przecięcie traktowane jest jako dwukrotne, a punkt przecież leży wewnątrz. Dlatego też również w tym algorytmie nie można uniknąć bezpośredniego sprawdzania, czy punkt nie pokrywa się z wierzchołkiem wielokąta.

Na zakończenie należy zwrócić uwagę na fakt, że istotna jest w nim jedynie liczba występujących przecięć, a nie same wartości współrzędnych punktów przecięcia. Aby ustalić liczbę przecięć nie zawsze musimy korzystać ze stwierdzania ich istnienia w drodze wyznaczania punktu przecięcia dwóch odcinków. Są takie przypadki położenia boków wielokąta, w których istnienie lub brak przecięcia można stwierdzić badając jedynie proste warunki. Jest to znacznie korzystniejsze czasowo niż wykonywanie obliczeń. Sposób postępowania w niniejszym algorytmie, pozwalający uniknąć obliczania niektórych przecięć może być następujący:

a) jeśli oba punkty boku leżą w po tej samej stronie odcinka pionowego wtedy bok nie przecina się z tym odcinkiem (boki na rys. 13a) i można przejść do analizy następnego boku, w przeciwnym wypadku sprawdzić następny warunek przedstawiony w punkcie b,
b) jeśli oba punkty boku leżą poniżej punktu, którego położenie badamy wtedy bok również nie przecina się odcinkiem pionowym (rys. 13b) i można przejść do analizy następnego boku, w przeciwnym wypadku należy przejść do punktu c,
c) jeśli oba punkty leżą powyżej punktu sprawdzanego wtedy stwierdzamy istnienie przecięcia bez obliczań (boki na rys. 13c) i przechodzimy do analizy następnego boku, w przeciwnym wypadku przechodzimy do punktu d,
d) konieczne jest stwierdzenie istnienie przecięcia przez obliczenie (boki na rys. 13d).




Rysunek 14. Ilustracja szczególnych położeń boków wielokąta w stosunku do odcinka pionowego

Jak widzimy przedstawiony sposób postępowania zdecydowanie ogranicza liczbę boków wielokąta, dla których w drodze obliczeń należy sprawdzać istnienie przecięcia z odcinkiem pionowym. Konieczność taka będzie występowała jedynie dla boków o położeniu schematycznie przedstawionym na rys. 13d.

## 4.3 REALIZACJA ZADANIA

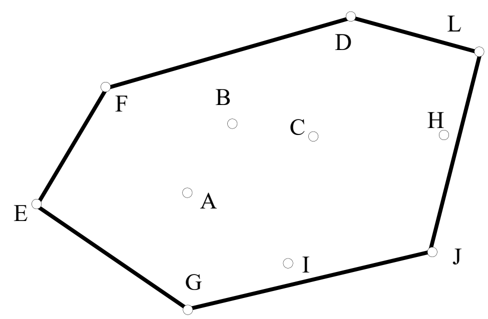
Należy przygotować program komputerowy w dowolnym środowisku programowania z zaimplementowanym jednym z algorytmów. Program powinien umożliwiać:

1. wczytanie wielokąta z pliku tekstowego,
2. przedstawić graficznie wczytany wielokąt,
3. sprawdzać położenie podawanych z klawiatury punktów i wizualizować je graficznie na tle wielokąta,
4. na podstawie pliku punktów z sprawdzić, ile punktów znajduje się wewnątrz wielokąta,
5. powinna istnieć możliwość konfigurowania parametrów rysunku, tzn.: kolory, grubości i style linii,

Przykładowe dane do obliczeń znajdują sie na stronie www.izdebski.edu.pl. Czas na wykonanie ćwiczenia: 3 zajęcia. Program powinien być oddany na koniec dziewiątych zajęć, w przypadku nieoddania wystawiana jest ocena negatywna.

# 5 WYZNACZENIE OTOCZKI WYPUKŁEJ ZBIORU PUNKTÓW

Określenie otoczki wypukłej zbioru punktów jest jednym z podstawowych zadań geometrii obliczeniowej. Zadanie polega na znalezieniu najmniejszego wielokąta **P** ograniczającego zbiór punktów **Q** takiego, że każdy z punktów leży na brzegu **P**, albo w jego wnętrzu. Wypukłą otoczkę zbioru oznaczamy przez **CH(Q)** (skrót od angielskiego convex hull). Odwołując się do intuicji, zbiór punktów możemy wyobrazić sobie jako gwoździe wystające z deski. Na gwoździe zakładamy gumkę, która przybierze kształt wypukłej otoczki. Jest wiele algorytmów wyznaczania otoczki wypukłej. Poniżej przedstawiono dwa tj.: algorytm Grahama (1972) oraz Jarvisa (1973). Cechą wspólną algorytmów jest zastosowanie metoda zwanej **„zamiataniem obrotowym"**, polegającym na przetwarzaniu wierzchołków w kolejności występowania kątów, jakie tworzą z ustalonym punktem i przechodzącym przez niego osią.

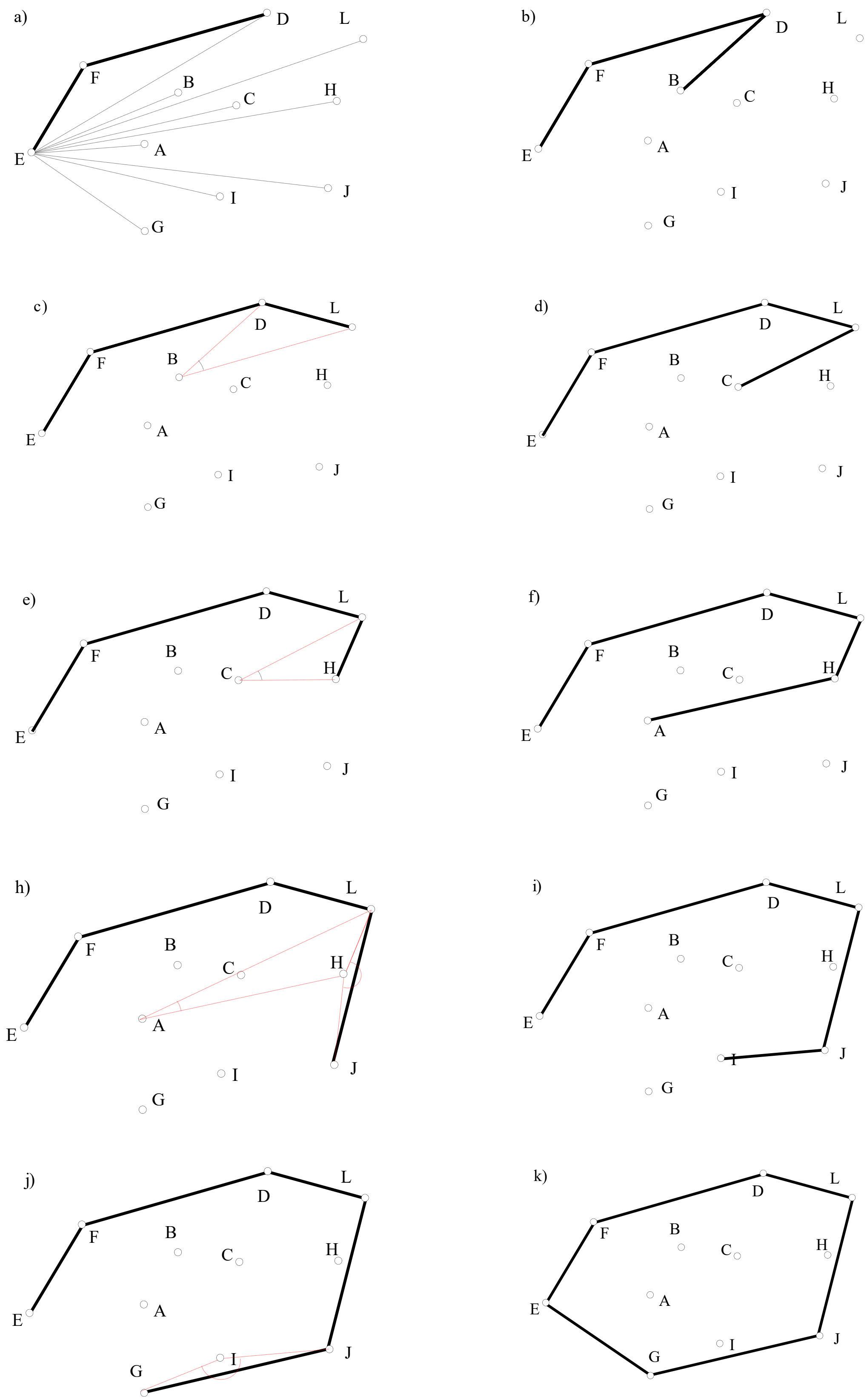



Rysunek 15. Obraz otoczki wypukłej P zbioru punktów Q

## 5.1 ALGORYTM GRAHAMA

W algorytmie Grahama otoczka jest wyznaczana z wykorzystaniem stosu **S**, który zawiera kandydatów na wierzchołki otoczki. Każdy punkt będący kandydatem do punktu otoczki jest raz wkładany na stos, a jeśli w dalszej analizie okazuje się, że nie spełnia odpowiednich warunków jest z niego zdejmowany. Po zakończeniu działania algorytmu stos zawiera wyłącznie wierzchołki **CH(Q)**. Kolejność czynności wykonywane w ramach algorytmu Grahama jest następująca:

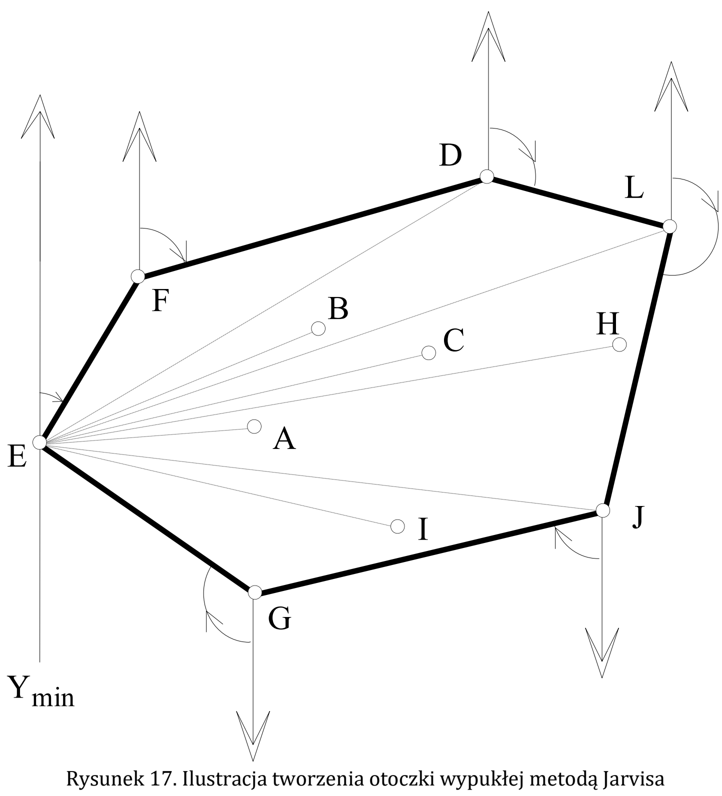
1. ze zbioru **Q** (zawierającego **n** punktów) wyznaczamy punkt **Po** o minimalnym **Y**, jeśli takich punktów jest więcej przyjmujemy punkt o największym **X**,
2. sortujemy punkty ze względu na azymuty promieni łączących punkt **Po** z kolejnymi punktami zbioru Q, jeśli więcej niż jeden promień posiada identyczną wartość azymutu usuwamy wszystkie punkty oprócz położonego najdalej od **Po** (najdłuższym promieniu),
3. w rezultacie tej operacji pozostaje do rozważania **m** punktów (kandydatów), z tego ograniczonego zbioru punktów do otoczki na pewno należy punkt ostatni czyli **Pm,**
4. zerujemy stos **S** zapamiętujący punkty otoczki i wprowadzamy na stos punktu **Po** oraz **P1** i **P2**,
5. od **j** w zakresie od **3** do **m** przeglądamy kolejne punkty w nawiązaniu do ostatnich dwóch punktów stosu **S** sprawdzając kierunek skrętu (wykorzystujemy własność iloczynu wektorowego), do chwili kiedy skręt w ostatnim wierzchołku na stosie do punktu **Pj** nie jest skrętem w prawo wtedy eliminujemy taki wierzchołek ze stosu dopiero przy skręcie w prawo dodajemy punkt **Pj** do stosu,
6. po przejrzeniu wszystkich punktów stos zawiera otoczkę wypukłą zbioru **Q**.




Rysunek 16. Ilustracja tworzenia otoczki wypukłej metodą Grahama

## 5.2 ALGORYTM JARVISA

W algorytmie Jarvisa wypukła otoczka zbioru punktów **Q** jest wyznaczana metodą zwaną owijaniem. Ujmując rzecz intuicyjne, algorytm Jarvisa naśladuje ciasne owijanie taśmy wokół punktów zbioru **Q**. Zaczynamy od zamocowania końca taśmy do położonego najbardziej z lewej strony punktu zbioru **Q**. Punkt ten oznaczamy jako **Po**. Punkt ten jest identyczny z punktem **Po** z algorytmu Grahama. Napinamy taśmę podnosząc ją do góry a następnie przesuwamy w prawo dopóki nie natrafimy na jakiś punkt. Punkt ten także musi należeć do otoczki. Tak więc trzymając cały czas napiętą taśmę wędrujemy naokoło zbioru, dopóki nie trafimy z powrotem do punktu początkowego. Formalnie rzecz biorąc postępowanie polega na tym, że zamiast napinania taśmy jako kolejny punkt otoczki bierzemy taki, który z jej punktem ostatnim daje promień wodzący o najmniejszej wartości azymutu. Postępowanie takie prowadzimy do chwili kiedy nie dotrzemy do wierzchołka położonego najbardziej z prawej strony. Po osiągnięciu tego punktu mamy skonstruowany łańcuch górny otoczki. Łańcuch dolny konstruujemy podobnie lecz zamiast wartości azymutu wykorzystujemy zawsze najmniejszą wartość kata utworzonego z ujemna półosią X.




Rysunek 17. Ilustracja tworzenia otoczki wypukłej metodą Jarvisa

W algorytmie niniejszym można zrezygnować z tworzenia osobnych łańcuchów górnego i dolnego ale wtedy należy operować wartością azymutu w zakresie od 0 do $2\pi$ co zmusza nas do jawnego obliczania jego wartości. Przy zastosowaniu podziału na łańcuchy możemy korzystać jedynie z iloczynu wektorowego zamiast z wartości azymutu.

## 5.3 REALIZACJA ZADANIA

Należy przygotować program komputerowy w dowolnym środowisku programowania z zaimplementowanym jednym z algorytmów. Program powinien umożliwiać:

1. wczytanie zbioru punktów z pliku tekstowego,
2. przedstawić graficznie wczytane punkty,
3. na podstawie zbioru punktów zbudować otoczkę wypukłą
4. powinna istnieć możliwość konfigurowania parametrów rysunku, tzn.: kolory, grubości i style linii.

Przykładowe dane do obliczeń znajdują się na stronie www.izdebski.edu.pl. Czas na wykonanie ćwiczenia: 3 zajęcia. Program powinien być oddany na koniec czternastych zajęć, w przypadku nieoddania wystawiana jest ocena negatywna.