

Pole Elektrostatyczne Dwóch Ładunków

Leszek Buczek, maj 2015

Zadanie domowe. Fizyka. Temat: **Stworzyć program komputerowy, który będzie rysował linie pola elektrostatycznego, jeżeli będziemy zadawali jako zmienne ładunek i odległość pomiędzy ładunkami.**

Środowisko pracy:

Dlaczego GWBasic?

- Mam już trochę programów w C/C++.
- Program jest łatwo wymienić na kod w C lub Pascal'u.
- Temat zadania jest prosty do kodowania i nie wymaga używania bogatej funkcjonalności C lub Pascal'a. Tak jak nie ma powodu wsiadać do Mercedesa aby dojechać na drugi koniec miasta - wystarczy wziąć rower.
- Interpreter działa wolniej (sprawdza każdą napotkaną instrukcję w trakcie działania programu) więc jest czas na analizę powstającej grafiki.
- Deklaracja zmiennych w GWBasic'u nie wymaga osobnego rejonu kodu. Można deklarować zmienne i przypisywać im wartości tuż przed jej użyciem.

Założenia programistyczne:

Powszechnie zakłada się, że ilość linii wychodząca z ładunku jest proporcjonalna do wielkości ładunku (ilości Coulombów). Przyjąłem więc wielkości ładunków od 1 do 5 mając na uwadze, że:

- Najmniejsza ilość linii wychodząca z ładunku to 5 (4 wydaje mi się niewystarczającą ilością).
- Wartość ładunku = 5 oznacza 25 linii dla jednego ładunku - według mnie nie powinno być ich więcej aby łatwo analizować grafikę.

Przyjąłem rzeczywistą skalę: 1 cm na ekranie monitora to rzeczywiście 1 cm w modelu.

Reszta jest tworem mojej wyobraźni.

Teoria

Pole sił elektrostatycznych rozpatruje się jako natężenie pola w danym punkcie. To właśnie natężenie pola charakteryzuje nam to pole. Pole mówi nam jaka siła będzie działać na ładunek próbny czyli taki, który nie wpływa na to pole ale odbiera jego istnienie. Z definicji pole elektrostatyczne to stosunek siły działającej na ładunek próbny, do wartości tego ładunku (symbolem natężenia pola jest E):

$$\vec{E} = \frac{\vec{F}}{q}$$

i jej zastosowanie

Przyjąłem istnienie dwóch ładunków. W tym przypadku wymagane jest dodawanie sił pochodzących od tych źródeł, superpozycja czyli znajdowanie wektora wypadkowego w danym punkcie pola. Gdy źródłem pola jest siła działająca na ładunek próbny q (o wartości obojętnej 1) od strony ładunków Q_1 i Q_2 to

$$F_1 = k * Q_1 * q / r_1^2 \quad \text{i} \quad F_2 = k * Q_2 * q / r_2^2$$

Ponieważ interesuje mnie tylko stosunek tych sił (a nie ich bezwzględna wartość), nie interesuje mnie bezwzględna długość wektorów tych sił. Silniejszy ładunek nie oznacza, aby był powodem szybszego "przemieszczania się" punktu w moim programie, kreślącym w konsekwencji linie pola. Pole jest statyczne. Wystarczy więc pominąć wartość k we wzorach, zostawiając tylko Q_1 , r_1 , Q_2 i r_2 . Wypadkowy wektor i tak 'sprasuję' równo jednocześnie w poziomie jak i w pionie i niezmiennie tak samo w ciągu całego działania programu (zmienna SKALA, ustawiona na wartość 100 w linii numer 2700).

Przedstawię wersję roboczą a nie egzekucyjną (pliku nie mającego w nazwie rozszerzenia EXE). Zaletą tego podejścia jest otwartość kodu do wglądu i edycji. Oprócz bowiem dynamicznej zmiany parametrów (podczas działania programu) możliwa jest wówczas statyczna zmiana wartości niektórych zmiennych jak np.

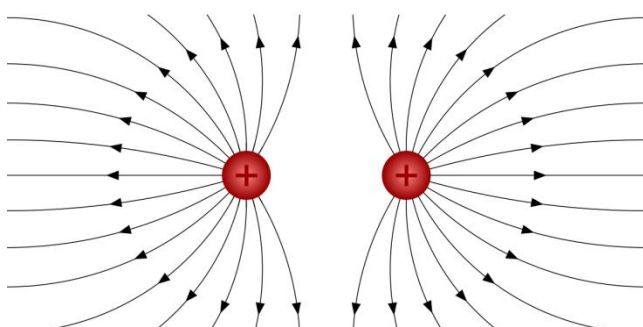
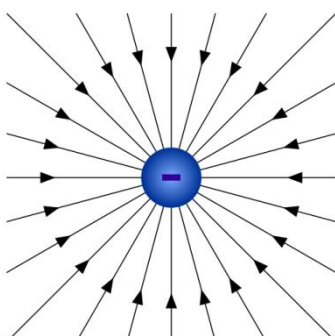
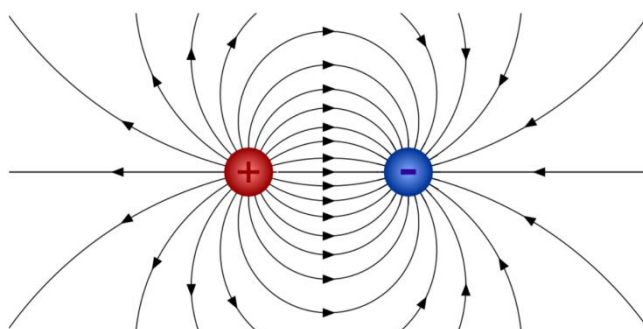
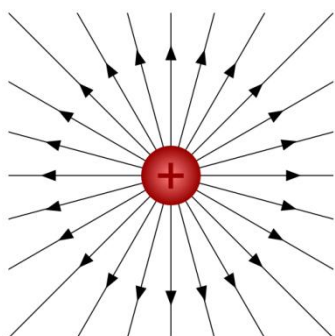
- o środka układu współrzędnych aby ukazać co się dzieje w odleglejszej strefie od pozycji ładunków - linia 170, zmienne XPQ i YPQ (współrzędne środka układu),
- o współczynnika wymiany długości z rzeczywistej na ekranową (pikselową) - linia 170, zmienna FQ (F od słowa *Factor*),
- o skali 'sprasowania' wektora wypadkowego. Im większa skala, tym linie są bardziej płynne i rysunek dokładniejszy - staje się to istotne, gdy przyjmuje się małą odległość pomiędzy ładunkami dowolnego rodzaju o różnej 'Wielkości ładunku'. W tym przypadku szybkość powstawania grafiki jest wolniejsza ale rysunek bardziej precyzyjny. W końcu teoria mówi, że najlepsze rozwiązanie znajduje się gdy odległość ta zmierza do zera. Jedynymi bowiem istotnymi parametrami są tu
 - kierunek i zwrot wektora wypadkowego
 - jednoczesne 'sprasowanie' współrzędnych tego wektora wypadkowego (i tylko jego na współrzędnych rzeczywistych) w tym samym stosunku i utrzymywanie tego stosunku od początku powstawania grafiki do jego końca - linia 2700, zmienna SKALA (zalecana wielkość od 100 do 300)

Specjalne kody ASCII mające potencjalne zastosowanie w programie (strona kodowa 852)

ą	165		ł	136		ś	152
ć	134		ń	228		ź	171
ę	169		ó	162		ż	190
+	43		-	45		0 (zero)	48

Ą	164		Ł	157		Ś	151
Ć	143		Ń	227		Ź	141
Ę	168		Ó	224		Ż	189

Oczekiwana grafika jest podobna do tej w publikacjach:



Wyszczególnione poniżej kolejno procedury zaczynają się od liczb podzielnych przez 100 w celu łatwiejszej analizy kodu a także możliwości rozszerzenia wybranej procedury bez uciekania się do wymiany numeracji linii niżej kodowanych procedur.

Nr linii	Opis procedur zaczynających się od tego numeru linii
10	Ustawienie środowiska pracy
300	Wprowadzenie danych dla ładunku pierwszego (pojawi się on później po lewej stronie ekranu)
500	Wprowadzenie danych dla ładunku drugiego (pojawi się on później po prawej stronie ekranu)
700	Wprowadzenie odległości pomiędzy dwoma ładunkami
900	Pozycjonowanie i rysowanie ładunku pierwszego - małego okręgu po lewej stronie ekranu
1000	Pozycjonowanie i rysowanie ładunku drugiego - małego okręgu po prawej stronie ekranu
1100	Ustalenie maksymalnych i minimalnych rysowanych wartości X i Y - piksele ekranu
1200	Ustawienie początkowych punktów dla wykreślanych linii tuż przy rysunkach ładunków
1300	Wyjaśnienie kodowania: sytuacja rzeczywista versus punkty ekranu
1400	Wyznaczenie pozycji punktów wokół ładunku pierwszego (po lewej stronie) na ekranie skąd będą zaczynać się linie pola elektrostatycznego - te punkty będą stabilizowane
1700	Wyznaczenie pozycji punktów wokół ładunku drugiego (po prawej stronie) na ekranie skąd będą zaczynać się linie pola elektrostatycznego - te punkty będą stabilizowane
2000	Początek procedury podającej na wyjściu współrzędne rzeczywiste X i Y. Wyjaśnienie zmiennych kodu.
2800	Wyznaczenie warunków wstępnych dla grafiki linii
3000	Rysowanie linii dla ładunku pierwszego (po lewej stronie ekranu)
4000	Rysowanie linii dla ładunku drugiego (po prawej stronie ekranu)
5000	Procedura wyjścia z programu
10000	Podprogram - wymiana wielkości rzeczywistych na punkty ekranu

Uwaga: Wprowadzając 'coś' z klawiatury do komputera, wprowadzamy łańcuchy znakowe (nie liczby, nawet jak nam się wydaje, że są to liczby) i później wymieniamy je na liczby. Podobnie jest w języku Python gdzie wprowadzone 123 to faktycznie łańcuch znakowy "123" (jeden dwa trzy). Dopiero 'rzut' (casting) na integer lub float daje liczbę.

Opis programu		
Ustawienie środowiska pracy Pole elektrostatyczne na płaszczyźnie co najwyżej dwóch ładunków. Uwagi: Ekran znakowy 80x25, gdzie 1,1 to górna lewa strefa ekranu. Piksel (1,1) to górna lewa strefa ekranu.		
100	KEY OFF	Usuń widok klawiszy funkcyjnych w dole ekranu; Ustaw tryb ekranu. SCREEN 9 jest najlepszy do grafiki (640x350 pikseli); Wyczyść ekran monitora Zadeklaruj użycie tablic 25-elementowych dla:
110	SCREEN 9	
120	CLS	
130	DIM X1P(25),Y1P(25),	

	X2P(25),Y2P(25)	<p>X1P(25) - rzeczywiste współrzędne X punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku pierwszego;</p> <p>Y1P(25) - rzeczywiste współrzędne Y punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku pierwszego;</p> <p>X2P(25) - rzeczywiste współrzędne X punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku drugiego;</p> <p>Y2P(25) - rzeczywiste współrzędne Y punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku drugiego;</p> <p>Zadeklaruj użycie tablic 25-elementowych dla:</p> <p>XX1P(25) - pikselowe (ekranowe) współrzędne X punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku pierwszego;</p> <p>YY1P(25) - pikselowe (ekranowe) współrzędne Y punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku pierwszego;</p> <p>XX2P(25) - pikselowe (ekranowe) współrzędne X punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku drugiego;</p> <p>YY2P(25) - pikselowe (ekranowe) współrzędne Y punktów ekranu, od których zaczynają się linie, tuż przy rysunku ładunku drugiego;</p> <p>Współrzędne środka układu do którego będzie odnosiła się grafika. 1.2 to współczynnik korygujący zwężenie obrazu koła (bez 1.2 koło byłoby przedstawione jako owal);</p> <p>FQ - ilość pikseli na jednostkę rzeczywistą (tu 1 cm);</p> <p>Ustaw kolor czcionki na biały (15), zostawiając tło czarne (0);</p> <p>Ustaw kursor na 12-tym miejscu w pierwszej linii;</p> <p>Pisz tekst- CHR\$(162) to ó;</p> <p>Zamień kolor czcionki na fioletowy (7) pozostawiając tło czarne (0);</p> <p>Ustaw kursor w drugiej linii i kolumnie 25-ej;</p> <p>Pisz tekst ;</p>
140	DIM XX1P(25),YY1P(25), XX2P(25),YY2P(25)	
170	XPQ=320:YPQ=210*1.2: FQ=26	
180	COLOR 15,0: LOCATE 1,28: PRINT "Leszek Buczek, 3 maja 2015";	
190	COLOR 7,0; LOCATE 2,26: PRINT "Linie pola elektrostatycznego";	
300	Wprowadzenie z klawiatury danych dla ładunku pierwszego (symbol ładunku pojawi się po lewej stronie ekranu)	
310	F=0: COLOR 15,0: LOCATE 4,10: PRINT "Pierwszy " CHR\$(136) "adunek [+,-,0] ": LOCATE 4,35: PRINT CHR\$(95): ZNAK1\$=INPUT\$(1)	<p>RODZAJ ŁADUNKU</p> <p>Flaga - 0 znaczy wpisano z klawiatury poprawną wartość; czcionka biała</p> <p>Ustaw kursor w czwartej linii, kolumnie 10; pisz tekst;</p> <p>Ustaw kursor w czwartej linii, kolumnie 35; pisz znak podkreślenia; czekaj na znak z klawiatury - oczekiwanym znakiem jest + albo - albo 0</p> <p>Jeżeli brak znaku (przyciśnięto ENTER) to zmusz ponownie do wpisania znaku</p> <p>Jeżeli nie jest to znak z zakresu +,-,0 to</p> <p>zaznacz to flagą F</p> <p>Jeżeli wpisano niepoprawną wartość to</p> <p>Ustaw kursor w piątej linii, kolumnie 10; wypisz ten znak i ostrzeżenie</p>
320	IF ZNAK1\$=CHR\$(13) THEN GOTO 310	
330	IF ZNAK1\$<>CHR\$(43) AND ZNAK1\$<>CHR\$(45) AND ZNAK1\$<>CHR\$(48) THEN F=1	
340	IF F=1 THEN LOCATE 5,10: PRINT ZNAK1\$ " ? Wprowad" CHR\$(171) " jeden z	

<pre> trzech znak" CHR\$(162) "w: + albo - albo 0" 350 IF F=1 THEN LOCATE 4,34: PRINT SPC(2): GOTO 310 ELSE LOCATE 5,10: PRINT SPC(52): LOCATE 4,35: PRINT ZNAK1\$ 360 IF ZNAK1\$=CHR\$(43) THEN KOL10=4 ELSE IF ZNAK1\$=CHR\$(45) THEN KOL10=1 ELSE KOL10=7 370 CIRCLE(300,48),4,KOL10 380 PAINT (300,48),KOL10 390 IF ZNAK1\$=CHR\$(48) THEN L1=0: GOTO 510 400 F=0: LOCATE 4,46: PRINT "Warto" CHR\$(152) CHR\$(134) SPC(1) CHR\$(136) "adunku, Q [1-5] ": LOCATE 4,71: PRINT CHR\$(95): LAD1\$=INPUT\$(1) 410 IF LAD1\$=CHR\$(13) THEN GOTO 400 420 IF LAD1\$<>"1" AND LAD1\$<>"2" AND LAD1\$<>"3" AND LAD1\$<>"4" AND LAD1\$<>"5" THEN F=1 430 IF F=1 THEN LOCATE 5,10: PRINT LAD1\$ " ? Wprowad" CHR\$(171) " warto" CHR\$(152) CHR\$(134) SPC(1) CHR\$(136) "adunku jako jedn" CHR\$(165) " z cyfr: 1, 2, 3, 4 lub 5" 440 IF F=1 THEN LOCATE 4,70: PRINT SPC(2): GOTO 400 ELSE LOCATE 5,10: PRINT SPC(68): LOCATE 4,71: PRINT LAD1\$ </pre>	<p>Jeżeli wpisano niepoprawną wartość to usuń wpisany znak i powtórz procedurę wpisania znaku</p> <p>Jeżeli wpisano poprawną wartość to skasuj ostrzeżenie (jeżeli takie było) wypisz na ekranie znak wprowadzony z klawiatury</p> <p>Jeżeli wpisanym znakiem jest '+' to deklaruj użycie koloru czerwonego</p> <p>W przeciwnym razie Jeżeli wpisanym znakiem jest '-' to deklaruj użycie koloru niebieskiego</p> <p>Jeżeli wpisanym znakiem jest '0' to deklaruj użycie koloru cyjanowego</p> <p>Wykreśl okrąg o promieniu 4 danym kolorem o środku w współrzędnych pikselowych (300,48)</p> <p>Maluj ten okrąg deklarowanym kolorem</p> <p>Jeżeli wpisanym znakiem jest '0' to deklaruj zerową ilość linii wychodzących od tego ładunku</p> <p>Ignoruj procedurę wprowadzenia 'Wartości ładunku'</p> <p>WARTOŚĆ ŁADUNKU</p> <p>Flaga - 0 znaczy wpisano z klawiatury poprawną wartość</p> <p>Tu flaga użyta także po to aby rozbić zdanie z 'IF' bo przekracza dopuszczalną ilość 250 znaków</p> <p>Pisz tekst na pozycji współrzędnych znakowych (4,46)</p> <p>Na pozycji (4,71) pisz podkreślenie i czekaj na cyfrę z klawiatury (będzie to znak)</p> <p>Jeżeli brak znaku (przyciśnięto ENTER) to zmusz ponownie do wpisania znaku</p> <p>Jeżeli znak nie jest z przedziału od '1' do '5' (to jeszcze nie cyfra !)</p> <p>zmień wartość flagi wskazującej na błąd</p> <p>Jeżeli wprowadzono błędne dane to od pozycji (5,10) wypisz ten znak i ostrzeżenie</p> <p>Jeżeli wprowadzono błędne dane to dodatkowo wyczyść znak z ekranu powtórz procedurę wpisania 'wartości ładunku'</p> <p>Jeżeli znak jest akceptowalny to wyczyść ostrzeżenie (jeżeli takie było) napisz znak jaki wprowadzono (będzie to znak od '1' do '5' -jeszcze nie liczba</p>
---	--

450	L1=VAL(LAD1\$)	Zamień znak na liczbę
500	Wprowadzenie z klawiatury danych dla ładunku drugiego (symbol ładunku pojawi się po prawej stronie ekranu)	
510	F=0: LOCATE 5,10: PRINT "Drugi " CHR\$(136) "adunek [+,-,0] ": LOCATE 5,35: PRINT CHR\$(95): ZNAK2\$=INPUT\$(1)	RODZAJ ŁADUNKU Flaga - 0 znaczy wpisano z klawiatury poprawną wartość Ustaw kursor w piątej linii, kolumnie 10; pisz tekst; Ustaw kursor w piątej linii, kolumnie 35; pisz znak podkreślenia; czekaj na znak z klawiatury - oczekiwanym znakiem jest + albo - albo 0 Jeżeli brak znaku (przyciśnięto ENTER) to zmusz ponownie do wpisania znaku Jeżeli nie jest to znak z zakresu +,-,0 to zaznacz to flagą F Jeżeli wpisano niepoprawną wartość to Ustaw kursor w szóstej linii, kolumnie 10; wypisz ten znak i ostrzeżenie
520	IF ZNAK2\$=CHR\$(13) THEN GOTO 510	usuń wpisany znak i powtórz procedurę wpisania znaku
530	IF ZNAK2\$<>CHR\$(43) AND ZNAK2\$<>CHR\$(45) AND ZNAK2\$<>CHR\$(48) THEN F=1	Jeżeli wpisano poprawną wartość to skasuj ostrzeżenie (jeżeli takie było) wypisz na ekranie znak wprowadzony z klawiatury
540	IF F=1 THEN LOCATE 6,10: PRINT ZNAK2\$ " ? Wprowad" CHR\$(171) " jeden z trzech znak" CHR\$(162) "w: + albo - albo 0": LOCATE 5,34: PRINT SPC(2): GOTO 510 ELSE LOCATE 6,10: PRINT SPC(52): LOCATE 5,35: PRINT ZNAK2\$	Jeżeli wpisanym znakiem jest '+' to deklaruj użycie koloru czerwonego
560	IF ZNAK2\$=CHR\$(43) THEN KOL2O=4 ELSE IF ZNAK2\$=CHR\$(45) THEN KOL2O=1 ELSE KOL2O=7	W przeciwnym razie Jeżeli wpisanym znakiem jest '-' to deklaruj użycie koloru niebieskiego
570	CIRCLE(300,62),4,KOL2O	Jeżeli wpisanym znakiem jest '0' to deklaruj użycie koloru cyjanowego Wykreśl okrąg o promieniu 4 danym kolorem o środku w współrzędnych pikselowych (300,62)
580	PAINT (300,62),KOL2O	Maluj ten okrąg deklarowanym kolorem
590	IF ZNAK2\$=CHR\$(48) THEN L2=0: GOTO 710	Jeżeli wpisanym znakiem jest '0' to deklaruj zerową ilość linii wychodzących od tego ładunku Ignoruj procedurę wprowadzenia 'Wartości ładunku'
600	F=0: LOCATE 5,46: PRINT "Warto" CHR\$(152) CHR\$(134) SPC(1) CHR\$(136) "adunku, Q [1-5] ": LOCATE 5,71: PRINT CHR\$(95): LAD2\$=INPUT\$(1)	WARTOŚĆ ŁADUNKU Flaga - 0 znaczy wpisano z klawiatury poprawną wartość Tu flaga użyta także po to aby rozbić zdanie z 'IF' bo przekracza dopuszczalną ilość 250 znaków Pisz tekst na pozycji współrzędnych znakowych (5,46)
610	IF LAD2\$=CHR\$(13) THEN GOTO 600	Na pozycji (5,71) pisz podkreślenie i czekaj na cyfrę z klawiatury (będzie to znak)
620	IF LAD2\$<>"1" AND LAD2\$<>"2" AND LAD2\$<>"3" AND LAD2\$<>"4" AND	Jeżeli brak znaku (przyciśnięto ENTER) to zmusz ponownie do wpisania znaku Jeżeli znak nie jest z przedziału od '1' do '5' (to jeszcze nie cyfra !)

630	<pre> LAD2\$<>"5" THEN F=1 IF F=1 THEN LOCATE 6,10: PRINT LAD2\$ " ? Wprowad" CHR\$(171) " warto" CHR\$(152) CHR\$(134) SPC(1) CHR\$(136) "adunku jako jedn" CHR\$(165) " z cyfr: 1, 2, 3, 4 lub 5" 640 IF F=1 THEN LOCATE 5,70: PRINT SPC(2): GOTO 600 ELSE LOCATE 6,10: PRINT SPC(68): LOCATE 5,71: PRINT LAD2\$ 650 L2=VAL(LAD2\$) </pre>	<p>zmień wartość flagi wskazującej na błąd</p> <p>Jeżeli wprowadzono błędne dane to od pozycji (6,10) wypisz ten znak i ostrzeżenie</p> <p>Jeżeli wprowadzono błędne dane to dodatkowo wyczyść znak z ekranu powtórz procedurę wpisania 'wartości ładunku'</p> <p>Jeżeli znak jest akceptowalny to wyczyść ostrzeżenie (jeżeli takie było) napisz znak jaki wprowadzono (będzie to znak od '1' do '5' -jeszcze nie liczba Zamień znak na liczbę</p>
700	Wprowadzenie odległości pomiędzy dwoma ładunkami	
710	<pre> LOCATE 6,18: PRINT "Odleg" CHR\$(136) "o" CHR\$(152) CHR\$(134) " pomi" CHR\$(169) "dzy " CHR\$(136) "adunkami w cm [1-14]": LOCATE 6,59: PRINT CHR\$(95) 720 P=0: I=0: ODL\$="": V\$="" 730 WHILE (V\$<>CHR\$(13) AND I<6) 740 V\$=INPUT\$(1): I=I+1: LOCATE 7,10: PRINT SPC(68) 750 LOCATE 6,58: PRINT SPC(1): LOCATE 6,58+I: PRINT V\$ CHR\$(95): F=0 760 IF V\$<>"0" AND V\$<>"1" AND V\$<>"2" AND V\$<>"3" AND V\$<>"4" AND V\$<>"5" AND V\$<>"6" AND V\$<>"7" AND V\$<>"8" AND V\$<>"9" AND V\$<>"," THEN F=1 770 IF F=1 THEN LOCATE 7,1: PRINT SPC(78): LOCATE 7,10: PRINT "Wprowad" CHR\$(171) " cyfr" CHR\$(169) " lub przecinek": I=I-1: LOCATE 6,59+I: PRINT CHR\$(95) SPC(2): GOTO 810 780 IF V\$="," THEN P=P+1: V\$="." </pre>	<p>Napisz tekst zaczynając od pozycji współrzędnych znakowych (6,18) ekranu i zapytaj o odległość pomiędzy ładunkami Na pozycji 6,59 pisz podkreślenie Inicjalizuj wartości zmiennych - liczbowych do zera, znakowych do pustego zbioru Wykonuj aż do [ENTER] albo ilości znaków większej od 6 (zaczynamy od 0)</p> <p>Akceptuj znak z klawiatury Podnieś licznik wpisanych znaków o jeden Wyczyść linię, która jest tekstem z ostrzeżeniem Wyczyść znak z pozycji (6,58) Dopisuj znaki wprowadzone z klawiatury z podkreśleniem na końcu Flaga - 0 znaczy wpisano z klawiatury poprawną wartość Jeżeli znak nie jest ani cyfrą ani przecinkiem to</p> <p>Sygnalizuj błędny (nieakceptowalny) znak Jeżeli wprowadzony znak jest nieakceptowalny to Wyczyść linię komentarzy ostrzeżeń Od pozycji (,10) pisz ostrzeżenie</p> <p>Zmniejsz licznik wprowadzonych znaków (ignoruj ostatni znak) Wyczyść zły znak i daj podkreślenie Omiń inne instrukcje w pętli WHILE dla tego znaku Jeżeli znakiem jest przecinek Podnieś licznik przecinków o jeden (inicjalnie był zerem) Zamień przecinek na kropkę dla celów obliczeniowych</p>

790	IF P>1 THEN LOCATE 7,1: PRINT SPC(78): LOCATE 7,10: PRINT "Mo" CHR\$(190) "e by" CHR\$(134) " tylko 1 przecinek w liczbie": I=I-1: P=P-1: LOCATE 6,59+I: PRINT CHR\$(95) SPC(2): GOTO 810	Jeżeli wprowadzony przecinek nie jest pierwszym przecinkiem to Wyczyść linię komentarzy ostrzeżeń Od pozycji (,10) pisz ostrzeżenie Zmniejsz licznik wprowadzonych znaków (ignoruj ostatni znak) Zmniejsz licznik wprowadzonych przecinków/znaków Ustaw kursor na miejscu wyświetlanych znaków wprowadzonych Omiń ostatnią instrukcję w pętli WHILE dla tego znaku Kompletuj zmienną znakową ODL\$ (odległość pomiędzy ładunkami)
800	ODL\$=ODL\$+V\$	While END - koniec pętli WHILE
810	WEND	Ustaw kursor znakowy na odpowiedniej pozycji i wyczyść 3 znaki spacjami
820	LOCATE 8,58+I: PRINT SPC(3)	Zamień ciąg znaków 'odległości pomiędzy ładunkami' na liczbę
830	ODLEG=VAL(ODL\$)	Sprawdź wielkość liczby -jeżeli nie jest z zakresu [1,14] to
840	IF ODLEG<1 OR ODLEG>14 THEN LOCATE 7,10: PRINT "Wprowad" CHR\$(171) " warto" CHR\$(152) CHR\$(134) " odleg" CHR\$(136) "o" CHR\$(152) "ci " CHR\$(136) "adunk" CHR\$(162) "w jako liczb" CHR\$(169) " pomi" CHR\$(169) "dzy 1 a 14"	W linii ostrzeżeń wypisz zdanie ostrzegawcze
850	IF ODLEG<1 OR ODLEG>14 THEN LOCATE 6,58: PRINT SPC(6): GOTO 710	Jeżeli nie jest z zakresu [1,14] to Wyczyść z ekranu wprowadzoną liczbę Zacznij od nowa budować 'odległość pomiędzy ładunkami'
860	LOCATE 7,1: PRINT SPC(75)	(Jeżeli liczba jest zaakceptowana) Wyczyść linię ostrzeżeń (nawet jeżeli jest pusta)
900	Ustalenie pozycji ładunku pierwszego i rysowanie ładunku pierwszego jako małego koła o promieniu 4 pikseli po lewej stronie ekranu	
920	XPOZ1=XPQ-ODLEG*FQ/2	910 '* Ustalenie pozycji x-owej pierwszego ładunku: y-kowa jest równa 0 'Pikselowy' środek ekranu, minus połowę odległości pomnożonej przez 'ilość pikseli na jednostkę' (domyślnie FQ=26; patrz linia 170)
940	CIRCLE(XPOZ1,YPQ/1.2),4,KOL10	930 '* Rysuj ładunek pierwszy o promieniu 4 pikseli (po lewej stronie ekranu) 1.2 to współczynnik korygujący zwężenie obrazu koła (bez 1.2 koło byłoby przedstawione jako owal)
950	PAINT (XPOZ1,YPQ/1.2),KOL10	Maluj pierwszy ładunek tym samym kolorem co koło (KOL10 to numer koloru)
1000	Ustalenie pozycji ładunku drugiego i rysowanie ładunku drugiego jako małego koła o promieniu 4 pikseli po prawej stronie ekranu	
1020	XPOZ2=XPQ+ODLEG*FQ/2	1010 '* Ustalenie pozycji x-owej drugiego ładunku; y-kowa jest równa 0 'Pikselowy' środek ekranu, plus połowę odległości pomnożonej przez 'ilość pikseli na jednostkę' (domyślnie FQ=26; patrz linia 170)
1040	CIRCLE(XPOZ2,YPQ/1.2),4,KOL20	1030 '* Rysuj ładunek drugi o promieniu 4 pikseli (po prawej stronie ekranu) 1.2 to współczynnik korygujący zwężenie obrazu koła (bez 1.2 koło byłoby przedstawione jako owal)
1050	PAINT (XPOZ2,YPQ/1.2),KOL20	Maluj drugi ładunek tym samym kolorem co koło (KOL10 to numer koloru)
1100	Ustalenie maksymalnych i minimalnych rysowanych wartości X i Y w pikselach ekranu	

1110 MINX=40: MAXX=600: MINY=100: MAXY=320	Liczby są w pikselach gdzie punkt (0,0) jest w górnym lewym kącie ekranu monitora
1200 Ustawienie początkowych punktów dla wykreślanych linii tuż przy rysunkach ładunków. Ich ilość jest następująca: - Dla 'wartości ładunku' = 1, wyprowadzanych jest 5 linii, z tym, że pierwsza linia w linii prostej pomiędzy ładunkami jest pominięta co można zauważyć w publikacjach. - Dla innej 'wartości ładunku', wyprowadzanych jest linii w ilości 5*'wartość ładunku'. Potrzebne są zarówno wartości rzeczywiste jak i ekranowe. Potrzebne obliczenia w strefie fizyki ładunku, wykonywane są na dokładnych wartościach rzeczywistych. Ekranowe (pikselowe) wartości potrzebne są do pozycjonowania linii na ekranie (punkty końcowe minimalnych linii tworzących linię ciągłą) są ustawione w konkretnym pikselu ekranu - daje to wrażenie linii nie perfekcyjnie regularnych ale bardzo dobrze widocznych. Wszystkie obliczane tu wielkości dotyczą strefy realnej. Dopiero procedura od numeru 10000 zamienia je na punkty ekranu.	
1400 Ładunek pierwszy - Wyznaczenie pozycji punktów wokół ładunku pierwszego (po lewej stronie) na ekranie skąd będą zaczynać się linie pola elektrostatycznego - te punkty będą stabilizowane. To, że zmienne dotyczą ładunku pierwszego charakteryzuje cyfra 1 w nazwach zmiennych: L 1 , ILOSCL 1 , XPOZ 1 R, YPOZ 1 R, X 1 P, Y 1 P, XX 1 P, YY 1 P, KATL 1	
1410 IF L1=0 THEN ILOSCL1=0: GOTO 1700 '* Ładunek zerowy czyli brak ładunku 1420 XPOZ1R= -ODLEG/2: YPOZ1R=0 1430 R=.2 1440 X1P(1)= XPOZ1R + R 1450 Y1P(1)= 0 1460 X=X1P(1): Y=Y1P(1) 1470 GOSUB 10000 1480 XX1P(1)=XX: YY1P(1)=YY 1490 ILOSCL1=L1*5 1500 '*- Te linie trzeba rozlozyc rownomiernie na okregu o promieniu R 1510 KATL1=360/ILOSCL1 1520 '- Jeden stopien to 0.017453 radiana, wiec... 1530 FOR I=2 TO ILOSCL1 1540 X1P(I)= XPOZ1R + R*COS(.017453*KATL1*(I-1)) 1550 Y1P(I)= R*SIN(.017453*KATL1*(I-1)) 1560 X=X1P(I): Y=Y1P(I) 1570 GOSUB 10000 1580 XX1P(I)=XX: YY1P(I)=YY '* Wspolrzedne poczatkow linii na ekranie	Jeżeli ładunek jest neutralny to nie wychodzą od niego linie - pomiń procedurę Pozycja środka ładunku pierwszego Promień okręgu na którym zaczynają się rysunki linii (okrąg ten nie jest wyświetlany ale jego punkty - początki linii - będą zapamiętywane/tablicowane) Pozycja x-owa 1-go punktu wokół ładunku pierwszego. Pozycja ta ustawiona jest naprzeciw ładunku drugiego Pozycja y-owa 1-go punktu wokół ładunku pierwszego Przygotuj punkt do wejścia do podprogramu (wejście to X i Y; wyjście będzie XX i YY) Wywołaj podprogram zamieniający punkty rzeczywiste na punkty ekranowe Zapamiętaj pierwszy punkt ekranowy. Stąd linie nie będą wyprowadzane ! Ilość linii wychodzących z ładunku pierwszego. Tu zaczyna się pozycjonowane następnych punktów (po pierwszym, już wypozyjonowanym) rzeczywistych i ekranowych Kąt w stopniach do następnego punktu. Ustalanie pozycji następnych punktów na okręgu R. Od nich będą rysowane linie. Zapamiętanie tych ekranowych (pikselowych) wartości w tablicy co najwyżej 25- elementowej X-owa wartość realna/rzeczywista Y-owa wartość realna/rzeczywista Przygotowanie współrzędnych (X,Y) do zamiany na wielkości ekranowe Wykonanie konwersji wartości rzeczywistych na ekranowe Ztablicowanie otrzymanych wielkości - będą wykorzystane w procedurze

1590 NEXT I	zaczynającej się od numeru linii 3000
1700 Ładunek drugi - Wyznaczenie pozycji punktów wokół ładunku drugiego (po prawej stronie) na ekranie skąd będą zaczynać się linie pola elektrostatycznego - te punkty będą stabilizowane To, że zmienne dotyczą ładunku drugiego charakteryzuje cyfra 2 w nazwach zmiennych: L2 , ILOSCL2 , XPOZ2R , YPOZ2R , X2P , Y2P , XX2P , YY2P , KATL2	
1710 IF L2=0 THEN ILOSCL2=0: GOTO 2000 '* Ładunek zerowy czyli brak ładunku 1720 XPOZ2R= ODLEG/2: YPOZ2R=0 1730 R=.2 1740 X2P(1)= XPOZ2R - R 1750 Y2P(1)= 0 1760 X=X2P(1): Y=Y2P(1) 1770 GOSUB 10000 1780 XX2P(1)=XX: YY2P(1)=YY 1790 ILOSCL2=L2*5 1800 '*- Te linie trzeba rozłożyć równomiernie na okręgu o promieniu R 1810 KATL2=360/ILOSCL2 1820 '- Jeden stopień to 0.017453 radiana, więc... 1830 FOR I=2 TO ILOSCL2 1840 X2P(I)= XPOZ2R - R*COS(.017453*KATL2*(I-1)) 1850 Y2P(I)= R*SIN(.017453*KATL2*(I-1)) 1860 X=X2P(I): Y=Y2P(I) 1870 GOSUB 10000 1880 XX2P(I)=XX: YY2P(I)=YY '* Współrzędne początków linii na ekranie 1890 NEXT I	Jeżeli ładunek jest neutralny to nie wychodzą od niego linie - pomiń procedurę Pozycja środka ładunku drugiego Promień okręgu na którym zaczynają się rysunki linii (okrąg ten nie jest wyświetlany ale jego punkty - początki linii - będą zapamiętywane/tablicowane) Pozycja x-owa 1-go punktu wokół ładunku drugiego. Pozycja ta ustawiona jest naprzeciw ładunku pierwszego Pozycja y-owa 1-go punktu wokół ładunku drugiego Przygotuj punkt do wejścia do podprogramu (wejdzie to X i Y; wyjście będzie XX i YY) Wywołaj podprogram zamieniający punkty rzeczywiste na punkty ekranowe Zapamiętaj pierwszy punkt ekranowy. Stąd linie nie będą wyprowadzane ! Ilość linii wychodzących z ładunku drugiego. Tu zaczyna się pozycjonowane następnych punktów (po pierwszym, już wypozycjonowanym) rzeczywistych i ekranowych Kąt w stopniach do następnego punktu. Ustalanie pozycji następnych punktów na okręgu R. Od nich będą rysowane linie. Zapamiętanie tych ekranowych (pikselowych) wartości w tablicy co najwyżej 25- elementowej X-owa wartość realna/rzeczywista Y-owa wartość realna/rzeczywista Przygotowanie współrzędnych (X,Y) do zamiany na wielkości ekranowe Wykonanie konwersji wartości rzeczywistych na ekranowe Stabilizowanie otrzymanych wielkości - będą wykorzystane w procedurze zaczynającej się od numeru linii 4000
W tym momencie mamy wszystkie potrzebne ustawienia - teraz można podjąć się obliczeń natężenia pola (najpierw dla punktów w pobliżu ładunków) i wykreślenia linii. 2000 Początek procedury podającej na wyjściu współrzędne rzeczywiste X i Y	
Parametry użyte do uzyskania obrazu graficznego (pytajnik poniżej '?' zastępuje cyfrę 1 lub 2): J - numer punktu, od którego zaczyna się linia. Zastępuje poprzednie I Pierwszy punkt to (znak ? zastępuje cyfry 1 i 2): J=1 w przypadku, gdy drugi ładunek jest neutralny. Oznacza to, że wszystkie linie w liczbie Wartość ładunku (LAD?\$)=ILOSCL?*5 będą wykreślane tylko wtedy, gdy drugi ładunek jest neutralny. J=2 w przypadku, gdy obydwa ładunki są albo tego samego znaku, albo znaku przeciwnego do siebie. Oznacza to, że pierwsza linia dla obu ładunków nie będzie wykreślana a są one dla obu ładunków naprzeciwko siebie. Upraszcza to programowanie a jednocześnie pozostawia obraz łatwiejszy do zrozumienia idei linii pola.	

Przypadki (istotne dla celu kodowania programu):

- PRZYP=1 Pierwszy ładunek jest zerowy/neutralny. Dla tego przypadku współrzędne natężenia pola pochodzącego od tego ładunku są wyzerowane: J dla drugiego ładunku jest 1.
- PRZYP=2 Drugi ładunek jest zerowy/neutralny. Dla tego przypadku współrzędne natężenia pola pochodzącego od tego ładunku są wyzerowane: J dla pierwszego ładunku jest 1.
- PRZYP=3 Pierwszy i drugi ładunek są przeciwnych znaków. Dla tego przypadku od czerwonego ładunku (+) wychodzą jasnoczerwone linie a od niebieskiego (-) linie jasnoniebieskie.
Początkowo J=2 co oznacza pierwszą kreśloną linię nie na przeciwko ładunku drugiego. Teoretycznie siła działająca na ładunek próbny jest od + i do -. W praktyce kodowania przyjęto zwrot siły od pierwszego do drugiego ładunku obojętnie jaki by nie był pierwszy (+ czy -) i drugi ładunek. W końcu kierunek działania sił jest umowny. Wobec tego, zwrot wektora siły pochodzącej od ładunku drugiego jest przeciwny (K=-1).
- PRZYP=4 Pierwszy i drugi ładunek są takiego samego znaku. Dla tego przypadku linie pola są tego samego koloru (cyjanowy).
Początkowo J=2 co oznacza pierwszą kreśloną linię nie na przeciwko ładunku drugiego. Równie dobrze mogło być J=1.
Zwrot wektora siły pochodzącej od ładunku drugiego jest taki sam jak ładunku pierwszego: K=1.

K - zwrot wektora siły pochodzącej od ładunku drugiego:

K=1 - dla ładunku drugiego gdy on jest tego samego znaku co pierwszy

K=-1 - dla ładunku drugiego gdy on jest innego znaku co pierwszy.

Początkowe punkty wykreślanych linii wyznaczone są przez:

X?P(J) i Y?P(J) - rzeczywiste współrzędne tuż przy ładunku.

XX?P(J) i YY?P(J) - współrzędne obrazu na ekranie tuż przy ładunku.

Współrzędne następnych punktów nie są już tablicowane bo nie muszą być zapamiętywane: punkt końcowy staje się punktem początkowym.

Końcowe punkty wykreślanych linii wyznaczone są przez:

Okno graficzne dla wykreślanych linii (MINX, MAXX, MINY i MAXY):

MINX - minimalna wartość X okna graficznego w pikselach ekranu

MAXX - maksymalna wartość X okna graficznego w pikselach ekranu

MINY - minimalna wartość Y okna graficznego w pikselach ekranu

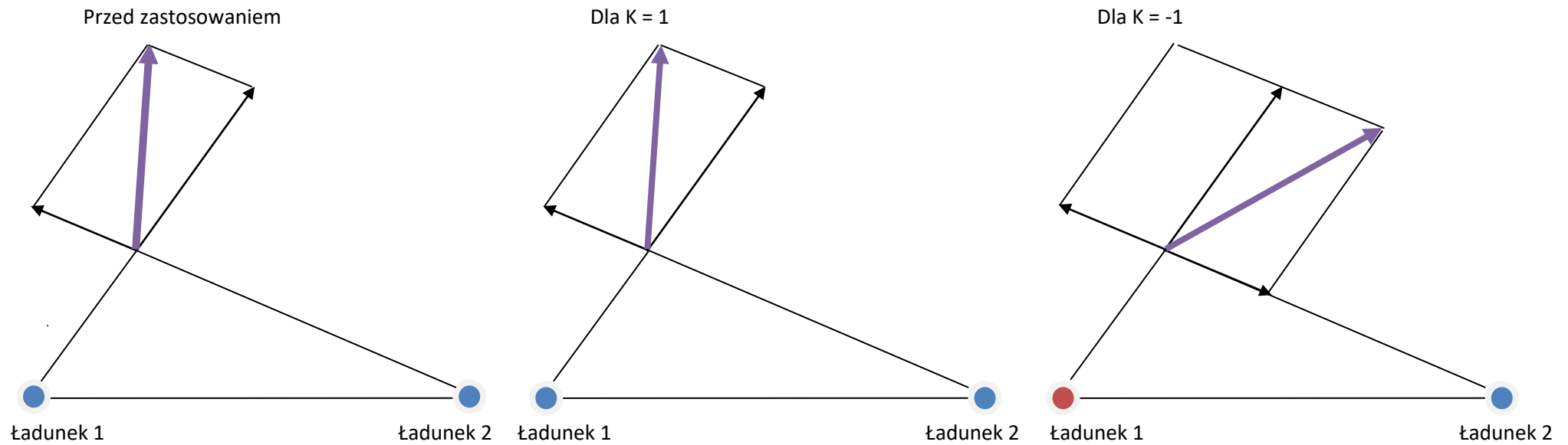
MAXY - maksymalna wartość Y okna graficznego w pikselach ekranu

Zbliżenie linii do przeciwnego ładunku na odległość nie mniejszą niż R, gdzie R jest promieniem okręgów na których zaczynają się linie przeciwnego ładunku (a więc także kończą roboczego ładunku).

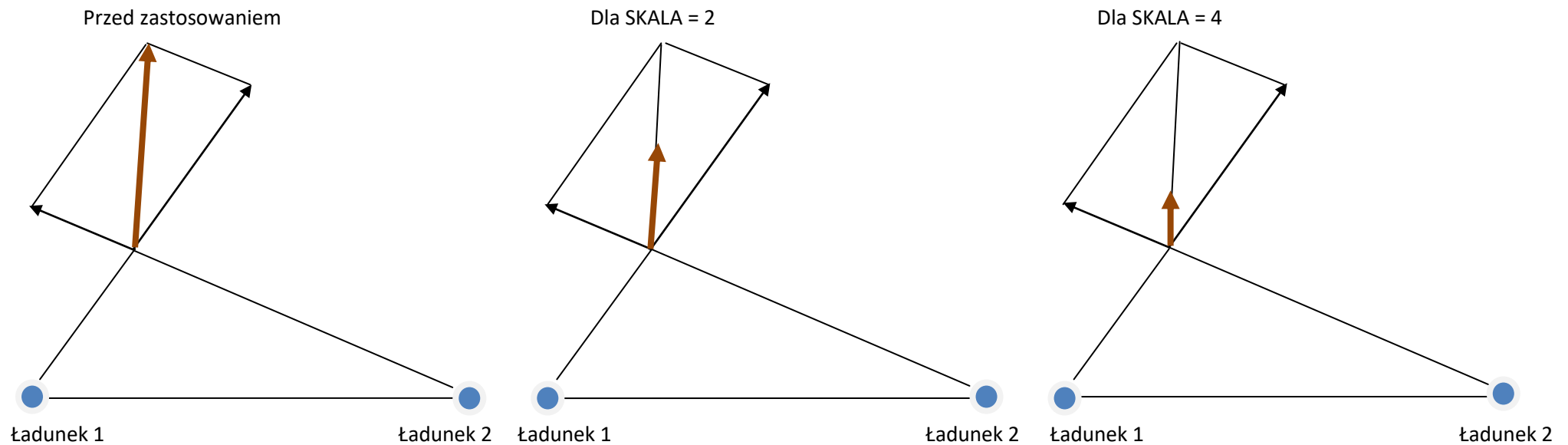
SKALA - Skala jest stała dla wszystkich współrzędnych obu sił - ustalona na bazie doświadczenia.

2700 SKALA=100 '* Im SKALA większa tym lepsza ale za bardzo spowalnia rysunek

Zobrazowanie zastosowania zmiennej zmiany zwrotu wektora (zmienna K)



Zobrazowanie zastosowania skali (zmienna SKALA)



2800	Wyznaczenie warunkow wstepnych dla grafiki linii	
2830	IF ZNAK1\$=CHR\$(48) THEN PRZYP=1: J=1: GOTO 4030	Jeżeli pierwszy ładunek jest zerowy (neutralny) Przygotuj pod rysowanie linii numer 1 ładunku drugiego (ona nie będzie rysowana) Skocz do linii numer 4030 (zaczynj rozważać rysowanie linii ładunku drugiego)
2840	IF ZNAK2\$=CHR\$(48) THEN PRZYP=2: J=1: GOTO 3030	Jeżeli drugi ładunek jest zerowy (neutralny) Przygotuj pod rysowanie linii numer 1 ładunku pierwszego (ona nie będzie rysow.) Skocz do linii numer 3030 (zaczynj rozważać rysowanie linii ładunku pierwszego)
2850	IF (ZNAK1\$=CHR\$(43) AND ZNAK2\$=CHR\$(45)) OR (ZNAK1\$=CHR\$(45) AND ZNAK2\$=CHR\$(43)) THEN PRZYP=3: K=-1: KOLOR=11: J=2: GOTO 3030	Jeżeli ładunki są przeciwnych znaków Zmień zwrot siły pochodzącej od ładunku drugiego Ustaw kolor na jednolity dla linii obu ładunków (tu: cyjanowy) Przygotuj pod rysowanie linii ładunku pierwszego (nie rysuj pierwszej linii) Skocz do linii numer 3030 (zaczynj rysowanie linii ładunku pierwszego)
2860	IF (ZNAK1\$=CHR\$(43) AND ZNAK2\$=CHR\$(43)) OR (ZNAK1\$=CHR\$(45) AND ZNAK2\$=CHR\$(45)) THEN PRZYP=4: K=1: J=2: GOTO 3030	Jeżeli ładunki są tego samego znaku Ne zmieniaj zwrotu siły pochodzącej od ładunku drugiego (Kolor linii będzie zależał od rodzaju ładunków) Przygotuj pod rysowanie linii ładunku pierwszego (nie rysuj pierwszej linii) Skocz do linii numer 3030 (zaczynj rysowanie linii ładunku pierwszego)
3000	Rysowanie linii dla ładunku pierwszego (po lewej stronie ekranu)	
3030	FOR M=J TO ILOSCL1	' STEP=1, domyślnie
3040	X1PL=X1P(M): Y1PL=Y1P(M)	
3050	XX=XX1P(M): YY=YY1P(M)-14	
3060	IF PRZYP<>3 THEN IF ZNAK1\$=CHR\$(43) THEN KOLOR=5 ELSE KOLOR=2	Wykonaj to co poniżej tyle razy ile linii ma być rysowane od ładunku pierwszego (ILOSCL1 - J). J=2 a więc nie rysuj pierwszej linii - tej prowadzącej w linii prostej do drugiego ładunku. Weź wartości współrzędnych rzeczywistych punktu początkowego Weź wartości współrzędnych ekranowych (w pikselach) punktu początkowego (obliczone już wcześniej) Jeżeli ładunki są przeciwnych znaków Jeżeli pierwszy ładunek jest dodatni (+) Przypisz kolor magneta dla tych linii W innym przypadku - a będzie to ładunek ujemny (-), daj liniom kolor zielony
Ustaw wartość logiczną 'rysuj' (RYSUJ jest 'prawdą' dla każdej wartości różnej od zera) albo przestań rysować (RYSUJ=0) Ustaw wartość pozycji punktu początkowego na ekranie monitora i przypisz mu kolor		
3070	RYSUJ=1	Każ programowi rysować
3080	PSET(XX,YY),KOLOR	'* Ustaw pióro i kolor na punkcie początkowym Ustaw wartość pozycji punktu początkowego na ekranie monitora i przypisz mu kolor
Rysuj jedną konkretną linię ładunku pierwszego aż do czasu gdy zmienna logiczna RYSUJ zmieni się na 0 (zero)		
3090	WHILE RYSUJ	'* RYSUJ=0 znaczy 'nie rysuj'; RYSUJ=1 znaczy 'rysuj' To co poniżej rób wiele razy. Na razie nie wiemy ile razy. Skoro RYSUJ ustawione jest

3100	IF PRZYP=2 THEN F2X=0: F2Y=0: GOTO 3270	na wartość inną niż 0 (zero), ta pętla logiczna (WHILE... - podczas gdy...) zostanie wykonana przynajmniej raz. Spodziewamy się jednak, że będzie wykonana wiele razy, aż do momentu gdy RYSUJ przyjmie wartość 0 (zero) albo wymusimy wyjście z pętli instrukcją bezwarunkową taką jak GOTO (instrukcja bez żadnego 'ale' - przeciwieństwo instrukcji logicznej dopuszczającej więcej niż jeden przypadek). Jeżeli ładunek drugi jest neutralny, pominiemy wpływ ładunku drugiego na pierwszy: Przyjmij, że współrzędne siły pochodzącej od ładunku drugiego są zerowe Skocz do linii numer 3270 pomijając wpływ ładunku drugiego na ładunek pierwszy
Obliczanie wpływu ładunku drugiego na ładunek próbny czyli teraz oblicz siłę wywieraną na punkt (ładunek próbny) przez ładunek drugi		
Obliczanie odległości pomiędzy środkiem ładunku drugiego a ładunkiem próbnym (naszym punktem roboczym) RX i RY spełniło swoją rolę i ich wartości nie są już potrzebne. Zastąpimy je nowymi (potrzebnymi teraz) wartościami RX - odległość współrzędnej X-owej pomiędzy ładunkiem drugim i próbnym RY - odległość współrzędnej Y-owej pomiędzy ładunkiem drugim i próbnym		
3160	RX=X1PL-XPOZ2R '* x-owa srodka ładunku 2-go jest zawsze dodatnia	YPOZ2R=0 - Y-owa współrzędna obu ładunków jest zerem więc jej nie odejmujemy ani ją nie dodajemy do Y1PL To twierdzenie Pitagorasa (pierwiastek z sumy kwadratów); ^ znaczy potęguj
3170	RY=Y1PL '*YPOZ2R=0 wiec RY=Y1PL a jego kwadrat zawsze nieujemny	
3180	R2=SQR(RX^2+RY^2)	
Obliczanie siły działającej na ładunek próbny ze strony ładunku drugiego. R2 nigdy nie będzie równy 0 (zeru) bo punkt ładunku próbnego nigdy nie osiągnie środka ładunku drugiego. Granicami są XPOZ2R+R, XPOZ2R-R, YPOZ2R+R i YPOZ2R-R		
3240	F2=L2/R2^2	F2 to bezwzględna długość wektora siły powodowanej ładunkiem drugim
3250	F2X=F2*RX/R2*K	F2X to współrzędna X-owa siły F2
3260	F2Y=F2*RY/R2*K	F2Y to współrzędna Y-owa siły F2
Obliczanie wpływu ładunku pierwszego na ładunek próbny czyli teraz oblicz siłę wywieraną na punkt (ładunek próbny) przez ładunek pierwszy		
Obliczanie odległości pomiędzy środkiem ładunku pierwszego a ładunkiem próbnym (naszym punktem roboczym) RX i RY spełniło swoją rolę i ich wartości nie są już potrzebne. Zastąpimy je nowymi (potrzebnymi teraz) wartościami RX - odległość współrzędnej X-owej pomiędzy ładunkiem pierwszym i próbnym RY - odległość współrzędnej Y-owej pomiędzy ładunkiem pierwszym i próbnym		
3320	RX=X1PL-XPOZ1R '* x-owa srodka ładunku 1-go jest zawsze ujemna	YPOZ1R=0 - Y-owa współrzędna obu ładunków jest zerem więc jej nie odejmujemy ani ją nie dodajemy do Y1PL To twierdzenie Pitagorasa (pierwiastek z sumy kwadratów); ^ znaczy potęguj
3330	RY=Y1PL '*YPOZ1R=0 wiec RY=Y1PL a jego kwadrat zawsze nieujemny	
3340	R1=SQR(RX^2+RY^2)	
Obliczanie siły działającej na ładunek próbny ze strony ładunku pierwszego. R1 nigdy nie będzie równy 0 (zeru) bo punkt ładunku próbnego nigdy nie osiągnie środka ładunku pierwszego. Granicami są XPOZ1R+R, XPOZ1R-R, YPOZ1R+R i YPOZ1R-R		
3400	F1=L1/R1^2	F1 to bezwzględna długość wektora siły powodowanej ładunkiem pierwszym
3410	F1X=F1*RX/R1	F1X to współrzędna X-owa siły F1
3420	F1Y=F1*RY/R1	F1Y to współrzędna Y-owa siły F1

Dodawanie do siebie współrzędnych wektorów sił: F1 i F2 i podzielenie wyniku przez skalę (SKALA) wziętą po prostu z doświadczenia (patrzenia na wynik grafiki - jeżeli linie nie są 'płynne' a nawet zachodzą na siebie, to koniecznie musimy zwiększyć wartość zmiennej SKALA; jeżeli linie są takie, jak oczekujemy, to zmniejszamy wartość zmiennej SKALA aby grafikę otrzymać szybciej (program wykonuje wtedy mniej obliczeń) Uwaga: Linie pola elektrostatycznego nie mogą zachodzić na siebie ani się przecinać !		
3440	' Dodaj do siebie wektory siły: $FX=F1X+F2X$ i $FY=F1Y+F2Y$	
3450	$FX=(F1X+F2X)/SKALA$: $FY=(F1Y+F2Y)/SKALA$ '* Im większa SKALA tym lepsza	Dodaj do siebie współrzędne X-owe obu wektorów Dodaj do siebie współrzędne Y-owe obu wektorów
Do tej pory odjęcie od siebie współrzędnych wektorów oznaczało, że były one zawieszone w punkcie o współrzędnych (0,0). Zawieś ten wektor w punkcie ładunku próbnego aby znaleźć następny punkt, w którym będzie ładunek próbny. Te wielkości są rzeczywiste (współrzędne końca wektora wypadkowego) będą początkiem następnego wektora wypadkowego początkującego następny cykl pętli WHILE. Zanim zaczniemy następny cykl pętli WHILE: a) Zawiesimy ten wektor w punkcie ładunku próbnego aby znaleźć następny punkt, w którym będzie ładunek próbny, b) Znajdziemy mu odpowiadający punkt na ekranie monitora		
3470	$X1KL=X1PL+FX$:	Współrzędna X-owa końca wektora wypadkowego zawieszonego w punkcie ładunku próbnego
	$Y1KL=Y1PL+FY$ '* To są następne punkty linii	Współrzędna Y-owa końca wektora wypadkowego zawieszonego w punkcie ładunku próbnego
3480	' Wejdz z tymi punktami do podprogramu aby znaleźć punkt na ekranie	
Sprawdź czy ta linia prowadzona od ładunku pierwszego nie przecina granicy otoczki ładunku drugiego wyznaczonej przez promień R. Jeżeli tak, to przerwij rysowanie linii. Może się zdarzyć, że linia przetnie ładunek i nieznacznie wyjdzie poza niego.		
3500	IF $X1KL < XPOZ2R+R$ AND $X1KL > XPOZ2R-R$ AND $Y1KL < YPOZ2R+R$ AND $Y1KL > YPOZ2R-R$ AND PRZYP<>2 THEN RYSUJ=0: GOTO 3550 ELSE $X=X1KL$: $Y=Y1KL$: GOSUB 10000	Jeżeli linia prowadzona od ładunku pierwszego (a nie jest to linia dla neutralnego ładunku drugiego i nie jest to PRZYP=2), wtedy: Przerwij rysowanie tej linii Pomiń (zignoruj) dalsze instrukcje pętli WHILE W przeciwnym wypadku: Przygotuj współrzędne punktu końcowego wektora wypadkowego do zamiany go na punkt ekranu Konwertuj (zamień) te współrzędne na współrzędne ekranu monitora
Sprawdź czy rysowana linia nie przekracza okna graficznego. Dlaczego linie wykraczające poza okno graficzne nie są kontynuowane tak, aby docierały do ładunku drugiego? Odpowiedź: Zdecydowałem się na to ponieważ przypadek ładunków przeciwnych znaków o zdecydowanej różnicy 'wartości ładunku' (jeden ładunek 1 a drugi 5) dawałoby taką samą ilość linii przy obu ładunkach pomimo zdecydowanej różnicy ich 'wartości ładunków'.		
3510	IF $XX < MINX$ OR $XX > MAXX$ OR $YY < MINY$ OR $YY > MAXY$	Jeżeli punkt jest poza ekranem graficznym (wykreślana linia przecina granice wyznaczone oknem graficznym):
	THEN RYSUJ=0: GOTO 3550	Przerwij rysowanie tej linii Pomiń (zignoruj) dalsze instrukcje pętli WHILE
Nadal rysuj linię a po narysowaniu tej minimalnej kreseczki zamień wartość współrzędnej rzeczywistej punktu końcowego wektora wypadkowego na początek następnego wektora wypadkowego, którego współrzędne końca obliczy następny cykl pętli WHILE		
3520	LINE -(XX,YY-14),KOLOR '* Rysuj linię do tego punktu od punktu poprzed.	Rysuj następny mały odcinek linii z góry ustalonym kolorem. Liczba 14 to wysokość

3530	X1PL=X1KL: Y1PL=Y1KL		kursora, który podnosi okno graficzne - wprowadzone w celu korekcji obrazu. Współrzędne punktu końcowego 'starego' wektora stają się współrzędnymi punktu początkowego 'nowego' wektora
3540	WEND		Koniec pętli WHILE (WEND czyli While END) - jej początek to linia 3090
3550	NEXT M		Koniec pętli FOR - jej początek to linia 3030
4000	Rysowanie linii dla ładunku drugiego (po prawej stronie ekranu) Kod jest taki sam jak z 'Rysowanie linii ładunku pierwszego' z tym, że dotyczy ładunku drugiego (Numeracja linii - tamte linie + 1000)		
4030	FOR M=J TO ILOSCL2		Wykonaj to co poniżej tyle razy ile linii ma być rysowane od ładunku drugiego (ILOSCL2 - J). J=2 a więc nie rysuj pierwszej linii - tej prowadzącej w linii prostej do pierwszego ładunku.
4040	X2PL=X2P(M): Y2PL=Y2P(M)		Weź wartości współrzędnych rzeczywistych punktu początkowego
4050	XX=XX2P(M): YY=YY2P(M)-14		Weź wartości współrzędnych ekranowych (w pikselach) punktu początkowego (obliczone już wcześniej)
4060	IF PRZYP<>3 THEN IF ZNAK2\$=CHR\$(43) THEN KOLOR=5 ELSE KOLOR=2		Jeżeli ładunki są przeciwnych znaków Jeżeli pierwszy ładunek jest dodatni (+) Przypisz kolor magneta dla tych linii W innym przypadku - a będzie to ładunek ujemny (-), daj liniom kolor zielony
Ustaw wartość logiczną 'rysuj' (RYSUJ jest 'prawdą' dla każdej wartości różnej od zera) albo przestań rysować (RYSUJ=0) Ustaw wartość pozycji punktu początkowego na ekranie monitora i przypisz mu kolor			
4070	RYSUJ=1		Każ programowi rysować
4080	PSET(XX,YY),KOLOR	* Ustaw pióro i kolor na punkcie początkowym	Ustaw wartość pozycji punktu początkowego na ekranie monitora i przypisz mu kolor
Rysuj jedną konkretną linię ładunku pierwszego aż do czasu gdy zmienna logiczna RYSUJ zmieni się na 0 (zero)			
4090	WHILE RYSUJ	* RYSUJ=0 znaczy 'nie rysuj'; RYSUJ=1 znaczy 'rysuj'	To co poniżej rób wiele razy. Na razie nie wiemy ile razy. Skoro RYSUJ ustawione jest na wartość inną niż 0 (zero), ta pętla logiczna (WHILE... - podczas gdy...) zostanie wykonana przynajmniej raz. Spodziewamy się jednak, że będzie wykonana wiele razy, aż do momentu gdy RYSUJ przyjmie wartość 0 (zero) albo wymusimy wyjście z pętli instrukcją bezwarunkową taką jak GOTO (instrukcja bez żadnego 'ale' - przeciwieństwo instrukcji logicznej dopuszczającej więcej niż jeden przypadek).
4100	IF PRZYP=1 THEN F1X=0: F1Y=0: GOTO 4270		Jeżeli ładunek pierwszy jest neutralny, pomiń wpływ ładunku pierwszego na drugi: Przyjmij, że współrzędne siły pochodzącej od ładunku pierwszego są zerowe Skocz do linii numer 4270 pomijając wpływ ładunku pierwszego na ładunek drugi
Obliczanie wpływu ładunku pierwszego na ładunek próbny czyli teraz oblicz siłę wywieraną na punkt (ładunek próbny) przez ładunek pierwszy			
Obliczanie odległości pomiędzy środkiem ładunku pierwszego a ładunkiem próbnym (naszym punktem roboczym) RX i RY spełniło swoją rolę i ich wartości nie są już potrzebne. Zastąpimy je nowymi (potrzebnymi teraz) wartościami RX - odległość współrzędnej X-owej pomiędzy ładunkiem pierwszym i próbnym RY - odległość współrzędnej Y-owej pomiędzy ładunkiem pierwszym i próbnym			

4160	RX=X2PL-XPOZ1R '* x-owa srodka ladunku 1-go jest zawsze ujemna	YPOZ1R=0 - Y-owa współrzędna obu ładunków jest zerem więc jej nie odejmujemy ani ją nie dodajemy do Y2PL To twierdzenie Pitagorasa (pierwiastek z sumy kwadratów); ^ znaczy potęguj
4170	RY=Y2PL '*YPOZ1R=0 wiec RY=Y2PL a jego kwadrat zawsze nieujemny	
4180	R1=SQR(RX^2+RY^2)	
Obliczanie siły działającej na ładunek próbny ze strony ładunku pierwszego. R1 nigdy nie będzie równy 0 (zeru) bo punkt ładunku próbnego nigdy nie osiągnie środka ładunku pierwszego. Granicami są XPOZ1R+R, XPOZ1R-R, YPOZ1R+R i YPOZ1R-R		
4240	F1=L1/R1^2 '* F1 to dlugosc wektora siły powodowanej ladunkiem 1-ym	F1 to bezwzględna długość wektora siły powodowanej ładunkiem pierwszym
4250	F1X=F1*RX/R1*K '* F1X to wspolrzedna x-owa siły F1	F1X to współrzędna X-owa siły F2
4260	F1Y=F1*RY/R1*K '* F1Y to wspolrzedna y-kowa siły F1	F1Y to współrzędna Y-owa siły F2
Obliczanie wpływu ładunku drugiego na ładunek próbny czyli teraz oblicz siłę wywieraną na punkt (ładunek próbny) przez ładunek drugi		
Obliczanie odległości pomiędzy środkiem ładunku drugiego a ładunkiem próbnym (naszym punktem roboczym) RX i RY spełniło swoją rolę i ich wartości nie są już potrzebne. Zastąpimy je nowymi (potrzebnymi teraz) wartościami RX - odległość współrzędnej X-owej pomiędzy ładunkiem drugim i próbnym RY - odległość współrzędnej Y-owej pomiędzy ładunkiem drugim i próbnym		
4320	RX=X2PL-XPOZ2R '* x-owa srodka ladunku 2-go jest zawsze dodatnia	YPOZ2R=0 - Y-owa współrzędna obu ładunków jest zerem więc jej nie odejmujemy ani ją nie dodajemy do Y2PL To twierdzenie Pitagorasa (pierwiastek z sumy kwadratów); ^ znaczy potęguj
4330	RY=Y2PL '*YPOZ2R=0 wiec RY=Y2PL a jego kwadrat zawsze nieujemny	
4340	R2=SQR(RX^2+RY^2)	
Obliczanie siły działającej na ładunek próbny ze strony ładunku drugiego. R2 nigdy nie będzie równy 0 (zeru) bo punkt ładunku próbnego nigdy nie osiągnie środka ładunku drugiego. Granicami są XPOZ2R+R, XPOZ2R-R, YPOZ2R+R i YPOZ2R-R		
4400	F2=L2/R2^2	F2 to bezwzględna długość wektora siły powodowanej ładunkiem drugim
4410	F2X=F2*RX/R2	F2X to współrzędna X-owa siły F1
4420	F2Y=F2*RY/R2	F2Y to współrzędna Y-owa siły F1
Dodawanie do siebie współrzędnych wektorów sił: F1 i F2 i podzielenie wyniku przez skalę (SKALA) wziętą po prostu z doświadczenia (patrzenia na wynik grafiki - jeżeli linie nie są 'płynne' a nawet zachodzą na siebie, to koniecznie musimy zwiększyć wartość zmiennej SKALA; jeżeli linie są takie, jak oczekujemy, to zmniejszamy wartość zmiennej SKALA aby grafikę otrzymać szybciej (program wykonuje wtedy mniej obliczeń) Uwaga: Linie pola elektrostatycznego nie mogą zachodzić na siebie ani się przecinać !		
4440	' Dodaj do siebie wektory siły: FX=F1X+F2X i FY=F1Y+F2Y i ...	
4450	FX=(F1X+F2X)/SKALA: FY=(F1Y+F2Y)/SKALA '* Im większa SKALA tym lepsza	Dodaj do siebie współrzędne X-owe obu wektorów Dodaj do siebie współrzędne Y-owe obu wektorów
Do tej pory odjęcie od siebie współrzędnych wektorów oznaczało, że były one zawieszone w punkcie o współrzędnych (0,0). Zawieś ten wektor w punkcie ładunku próbnego aby znaleźć następny punkt, w którym będzie ładunek próbny. Te wielkości są rzeczywiste (współrzędne końca wektora wypadkowego) będą początkiem następnego wektora wypadkowego początkującego następny cykl pętli WHILE. Zanim zaczniemy następny cykl pętli WHILE: a) Zawiesimy ten wektor w punkcie ładunku próbnego aby znaleźć następny punkt, w którym będzie ładunek próbny, b) Znajdziemy mu odpowiadający punkt na ekranie monitora		
4470	X2KL=X2PL+FX:	Współrzędna X-owa końca wektora wypadkowego zawieszonego w punkcie ładunku

4480	Y2KL=Y2PL+FY '* To sa nastepne punkty linii ' Wejdz z tymi punktami do podprogramu aby znalezc punkt na ekranie	próbnego Współrzędna Y-owa końca wektora wypadkowego zawieszzonego w punkcie ładunku próbnego
Sprawdź czy ta linia prowadzona od ładunku drugiego nie przecina granicy otoczki ładunku pierwszego wyznaczonej przez promień R. Jeżeli tak, to przerwij rysowanie linii. Może się zdarzyć, że linia przetnie ładunek i nieznacznie wyjdzie poza niego.		
4500	IF X2KL<XPOZ1R+R AND X2KL>XPOZ1R-R AND Y2KL<YPOZ1R+R AND Y2KL>YPOZ1R-R AND PRZYP<>1 THEN RYSUJ=0: GOTO 4550 ELSE X=X2KL: Y=Y2KL: GOSUB 10000	Jeżeli linia prowadzona od ładunku drugiego (a nie jest to linia dla neutralnego ładunku pierwszego i nie jest to PRZYP=1), wtedy: Przerwij rysowanie tej linii Pomiń (zignoruj) dalsze instrukcje pętli WHILE W przeciwnym wypadku: Przygotuj współrzędne punktu końcowego wektora wypadkowego do zamiany go na punkt ekranu Konwertuj (zamień) te współrzędne na współrzędne ekranu monitora
Sprawdź czy rysowana linia nie przekracza okna graficznego. Dlaczego linie wykraczające poza okno graficzne nie są kontynuowane tak, aby docierały do ładunku drugiego? Odpowiedź: Zdecydowałem się na to ponieważ przypadek ładunków przeciwnych znaków o zdecydowanej różnicy 'wartości ładunku' (jeden ładunek 1 a drugi 5) dawałoby taką samą ilość linii przy obu ładunkach pomimo zdecydowanej różnicy ich 'wartości ładunków'.		
4510	IF XX<MINX OR XX>MAXX OR YY<MINY OR YY>MAXY THEN RYSUJ=0: GOTO 4550	Jeżeli punkt jest poza ekranem graficznym (wykreślana linia przecina granice wyznaczone oknem graficznym): Przerwij rysowanie tej linii Pomiń (zignoruj) dalsze instrukcje pętli WHILE
Nadal rysuj linię a po narysowaniu tej minimalnej kresczki zamień wartość współrzędnej rzeczywistej punktu końcowego wektora wypadkowego na początek następnego wektora wypadkowego, którego współrzędne końca obliczy następny cykl pętli WHILE		
4520	LINE -(XX,YY-14),KOLOR '* Rysuj linie do tego punktu od punktu poprzed.	Rysuj następny mały odcinek linii z góry ustalonym kolorem. Liczba 14 to wysokość kursora, który podnosi okno graficzne - wprowadzone w celu korekcji obrazu. Współrzędne punktu końcowego 'starego' wektora stają się współrzędnymi punktu początkowego 'nowego' wektora Koniec pętli WHILE (WEND czyli While END) - jej początek to linia 4090 Koniec pętli FOR - jej początek to linia 4030
4530	X2PL=X2KL: Y2PL=Y2KL	
4540	WEND	
4550	NEXT M	
5000	Procedura wyjścia z programu	
5010	LOCATE 25,18: PRINT "Naci" CHR\$(152) "nij dowolny klawisz aby wyj" CHR\$(152) CHR\$(134) " z programu";	Ustaw kursor tekstowy na pozycji (25,18) ekranu (ostatnia linia na ekranie); Pisz tekst;
5020	S\$=INKEY\$: IF S\$="" THEN 5020	Czekaj na przyciśnięcie jakiegokolwiek klawisza;
5030	CLS: END	Wyczyść ekran monitora; Zakończ program

10000 Podprogram - wymiana wielkości rzeczywistych na punkty ekranu		
10030	XQ=X: YQ=Y	<p>Zamień X i Y na wielkości robocze podprogramu; Dostosuj współrzędną rzeczywistą X-ową punktu do współrzędnych X-owych ekranowych Dostosuj współrzędną rzeczywistą Y-ową punktu do współrzędnych Y-owych ekranowych Instrukcja CINT odpowiada za znalezienie najbliższego piksela (aktywnego punktu ekranowego) aby obraz naszego punktu nie był rozmazany; Zamień obliczone przez podprogram wielkości na te, które wyjdą z podprogramu do programu głównego; Zakończ podprogram i skocz do miejsca tuż po linii skąd wywołany został podprogram (instrukcją GOTO 10000). Wszystkie zmienne w GWBasic'u są traktowane jak globalne stąd wartości wszystkich zmiennych podprogramu są widoczne w programie głównym (RETURN to powrót do programu głównego i nie 'zwraca' do programu głównego)</p>
10040	XXQ=CINT(XQ*FQ)+XPQ '*XX-pikselowa współrzędna X punktu na ekranie	
10050	YYQ=CINT(-YQ*FQ)+YPQ '*YY-pikselowa współrzędna Y punktu na ekranie ' FQ - jednostka obu osi układu w pikselach	
10060	XX=XXQ: YY=YYQ/1.2 '* 1.2 - współczynnik korygujący zniekształcenie obrazu	
10070	RETURN	

Warunki testowe						
Ładunek 1		Ładunek 2		Odległość ładunkw	Skala	Rezultat testu
Rodzaj	Wartość	Rodzaj	Wartość			
0		0		od 1 do 14, w tym liczby nie będące całkowitymi	od 1 do 200	Taki jak oczekiwany
0		+ od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 60 daje niesatysfakcjonujące rezultaty
0		- od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 60 daje niesatysfakcjonujące rezultaty
+ od 1 do 5		0		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 60 daje niesatysfakcjonujące rezultaty
- od 1 do 5		0		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 60 daje niesatysfakcjonujące rezultaty
+ od 1 do 5		- od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 150 daje niesatysfakcjonujące rezultaty, tym bardziej nie do akceptacji im odległości pomiędzy ładunkami maleją. Użycie SKALA=200 daje dobre rezultaty.
- od 1 do 5		+ od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 150 daje niesatysfakcjonujące rezultaty, tym bardziej nie do akceptacji im odległości pomiędzy ładunkami maleją. Użycie SKALA=200 daje dobre rezultaty.
+ od 1 do 5		+ od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 100 daje niesatysfakcjonujące rezultaty, tym bardziej nie do akceptacji im odległości pomiędzy ładunkami maleją i tym bardziej gdy zwiększa się różnica pomiędzy wartościami ładunków. Użycie SKALA=200 daje dobre rezultaty.
- od 1 do 5		- od 1 do 5		od 1 do 14, w tym liczby nie będące całkowitymi	od 60 do 200	Użycie skali poniżej 100 daje niesatysfakcjonujące rezultaty, tym bardziej nie do akceptacji im odległości pomiędzy ładunkami maleją i tym bardziej gdy zwiększa się różnica pomiędzy wartościami ładunków. Użycie SKALA=200 daje dobre rezultaty.

Warunek testu	Specyfika danych	Rezultat
Niepoprawne wprowadzenie ładunku	a; ++; +0; 3; abc.d	Odpowiedź programu jest poprawna: Wyświetlanie wiadomości, usuwanie niepoprawnych danych, usunięcie wiadomości w przypadku wprowadzenia poprawnych danych.
Niepoprawne wprowadzenie wielkości ładunku	1.3; 0.5; aaa; 3.4; 4,2	Odpowiedź programu jest poprawna: Wyświetlanie wiadomości, usuwanie niepoprawnych danych, usunięcie wiadomości w przypadku wprowadzenia poprawnych danych.
Niepoprawne wprowadzenie odległości ładunków	0.9; aaa; 4A; 4,6; 14.1; 100; -5	Odpowiedź programu jest poprawna: Wyświetlanie wiadomości, usuwanie niepoprawnych danych, usunięcie wiadomości w przypadku wprowadzenia poprawnych danych.
Zmiana środka układu współrzędnych, XPQ, YPQ	Dla obu: od 100 do 400 (inicjalnie XPQ=320:YPQ=219)	Użycie wielkości pozaekranowych przerywa działanie programu. Jakakolwiek wielkość wstępna punktów ładunku poza oknem graficznym niekontrolowanie przerywa działanie programu.
Zmiana skali, SKALA	od 1 do 500 (inicjalnie SKALA=150)	Użycie skali poniżej 50 nie daje dobrych rezultatów. Zalecana skala to od 150 do 300. Użycie wyższych wartości skali (SKALA>300) jest zbyt duże i wyraźnie spowalnia działanie programu.
Zmiana jednostki długości na osi układu współrzędnych FQ	od 1 do 50 (inicjalnie FQ=26)	Użycie FQ<10 nie ma sensu ze względu na rozmiar okręgów, będących początkami linii. FQ>38 powoduje niekontrolowane przerwanie działania programu. FQ w granicach 15-30 jest OK.