



Ruch cząstki naładowanej w polu magnetycznym

Wstęp

Na podstawie wielu doświadczeń stwierdzono, że wartość siły \vec{F} działającej na ładunek q , który porusza się z prędkością \vec{v}_\perp składową w kierunku prostopadłym do linii danego pola magnetycznego, jest proporcjonalna do iloczynu qv_\perp :

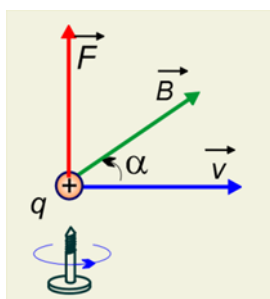
$$F = qv_\perp$$

Oczywiście, ten sam ładunek q , poruszający się z taką samą prędkością prostopadle do linii pola w innym polu magnetycznym, dozna działania siły \vec{F} o innej wartości.

Wartość siły magnetycznej \vec{F} działającej na jednostkę (qv_\perp) informuje nas jak „silne” jest pole magnetyczne. Nazywamy ją **wartością wektora \vec{B} indukcji pola magnetycznego**. Przyjmujemy zatem dla pola magnetycznego

$$B = \frac{F}{|q|v_\perp}$$

gdzie v_\perp oznacza wartość prędkości składowej w kierunku prostopadłym do linii pola. Wektor \vec{B} w danym punkcie pola ma kierunek i zwrot zgodny z linią pola magnetycznego, przy czym wektory \vec{v} , \vec{B} , \vec{F} stanowią układ prawoskrętny (zgodny z regułą śruby prawej – patrz rys. 1). Taki zwrot siły \vec{F} otrzymamy, gdy ładunek jest dodatni. W przypadku ładunku ujemnego (np. elektronu) zwrot siły magnetycznej jest przeciwny.



Rysunek 1. Układ trzech wektorów \vec{v} , \vec{B} , \vec{F} – prawoskrętny, zgodny z regułą śruby prawej

Jednostką indukcji pola magnetycznego jest **tesla** (1 T):

$$1 \text{ T} = \frac{1 \text{ N}}{1 \text{ C} \cdot 1 \frac{\text{m}}{\text{s}}} = 1 \frac{\text{kg}}{\text{A} \cdot \text{s}^2}$$

Warto zwrócić uwagę, że jest to dość duża jednostka. Ziemskie pole magnetyczne ma indukcję rzędu 10^{-5} T, typowe magnesy sztabkowe wytwarzają w pobliżu biegunów pole o indukcji rzędu kilku- kilkudziesięciu mT. Najsilniejsze pola magnetyczne, które potrafimy wytworzyć w laboratoriach, dochodzą do 10^3 T.

Gdy prędkość ładunku jest prostopadła do wektora \vec{B} , wartość siły magnetycznej zapiszemy następująco:

$$F = qv_\perp B$$

Gdy prędkość jest skierowana do wektora indukcji \vec{B} pod pewnym kątem $\alpha \neq \frac{\pi}{2}$, wtedy wektor prędkości można rozłożyć na dwie składowe: poprzeczną $v_\perp = v \cdot \sin \alpha$ i podłużną

$v_{\parallel} = v \cdot \cos\alpha$. Siła magnetyczna przy ruchu z prędkością podłużną jest równa zero. Zatem w wypadku ruchu ładunku w dowolnym kierunku siła magnetyczna zależy tylko **od składowej poprzecznej prędkości**:

$$F = qv_{\perp}B = qvB\sin\alpha$$

Powyższy wzór wyraża wartość siły magnetycznej w ogólnym przypadku. Nosi ona nazwę **siły Lorentza**. Siłę Lorentza można wyrazić za pomocą iloczynu wektorowego:

$$\vec{F} = q \cdot (\vec{v} \times \vec{B})$$

Siła ta pełni rolę siły dośrodkowej:

$$qvB = \frac{mv^2}{r}$$

gdzie m to masa cząstki, q jej ładunek, v prędkość, r to promień toru ruchu, B – indukcja pola magnetycznego.

Przekształcając,

$$\frac{v}{r} = \frac{qB}{m}$$

v/r odpowiada prędkości kątowej ω , stąd **częstość cyklotronowa**:

$$\omega = \frac{qB}{m}$$

Jak działa model ruchu cząstki naładowanej w polu magnetycznym w Excelu

W arkuszu *wykres i parametry* umieszczony został wykres przedstawiający tor cząstki naładowanej w układzie współrzędnych (R, Z) (rys. 2).

W komórkach kolumn A i B znajdują się parametry wejściowe do sterowania wykresem. Nie wszystkie będziemy zmieniać – część z nich przyjmie w naszej symulacji ustalone wartości.

W komórce B5 znajduje się masa cząstki m wyrażona w masach protonu,

w B6 – ładunek q ,

B7 – wartość indukcji pola magnetycznego B (u nas może przyjmować wartości z przedziału od 1,25 do 5 T, do regulacji tego parametru zastosujemy pasek przewijania),

B9 – v_{x0} prędkość początkowa w kierunku osi OX (może przyjmować wartości od 1 do 5 cm/ns – takie są parametry graniczne paska przewijania, który tu zastosowaliśmy),

B10 – v_{y0} prędkość początkowa w kierunku osi OY (v_{y0} w cm/ns),

B11 – v_{z0} prędkość początkowa w kierunku osi OZ (v_{z0} może przyjmować wartości od 0,1 do 1 cm/ns)

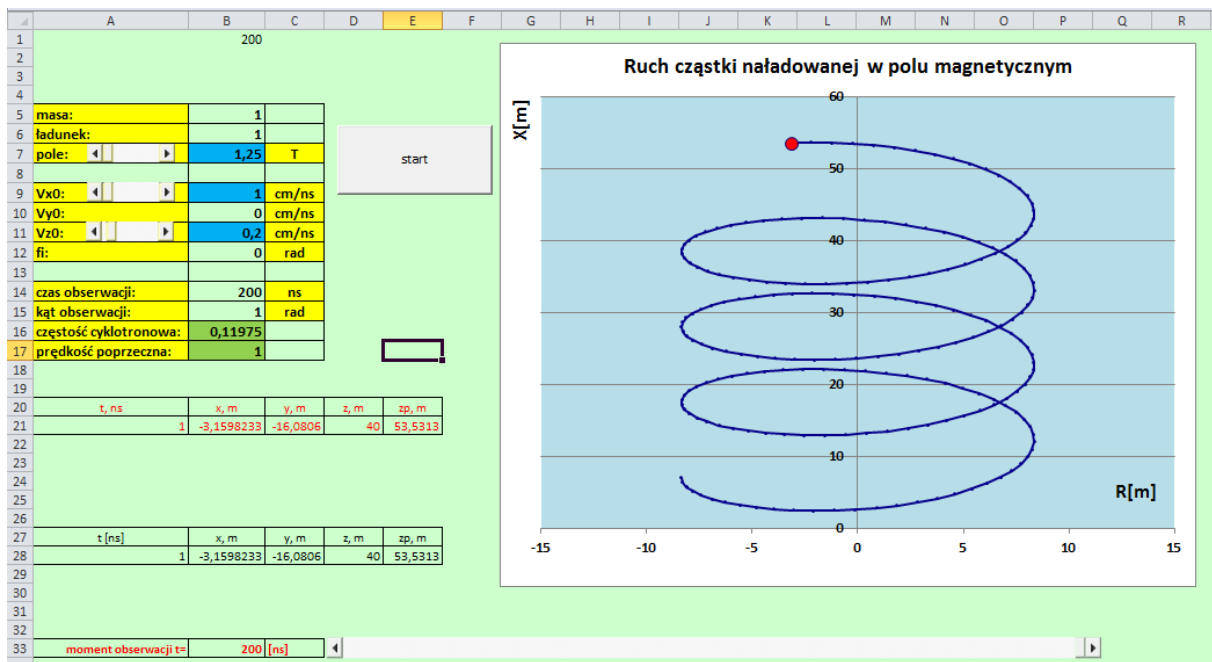
B12 – fi jest to faza początkowa φ we wzorach typu $= (-vt \cdot \cos(\omega \cdot t \cdot t_o + \varphi))/\omega$,

B14 – t_o czas obserwacji (u nas stała wartość 200 ns),

B15 – α kąt obserwacji (u nas 1 rad),

B16 – w tej komórce obliczana jest ω częstość cyklotronowa (ze wzoru $0,0958 \cdot q \cdot B/m$),

B17 – prędkość poprzeczna dana wzorem: $\sqrt{v_{x0}^2 + v_{y0}^2}$.



Rysunek 2. Wykres i parametry – model w Excelu

Wprowadź przykładowe wartości do komórek B7, B9 i B11 (mogą być takie, jak na rysunku 2), a następnie naciśnij przycisk **start**. Przyciskiem tym uruchomisz makro, które przygotowuje i przeprowadzi symulację ruchu cząstki, wyznaczając bieżące położenia – tabela wartości znajduje się w arkuszu *obliczenia*. Wartości z arkusza *obliczenia* zostaną wyświetlone na wykresie ukazując przebieg ruchu.

Makro – jest to program napisany w języku VBA (Visual Basic for Application), który wykonuje zapisane w nim instrukcje automatyzując wykonywanie pewnych czynności. Kod makra możemy tworzyć i modyfikować w specjalnie do tego celu przeznaczonym edytorze stowarzyszonym z produktami MS Office. Więcej informacji o makrach znajdziesz w *Dodatku* umieszczonym na końcu instrukcji.

Wspominaliśmy już, że całkowity czas ruchu (u nas 200 ns) znajduje się w komórce B15, natomiast w komórce B33 możesz obserwować bieżący czas trwania symulacji.

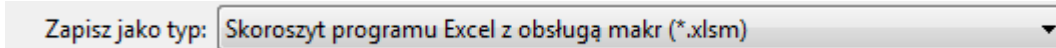
Poniżej wykresu umieściliśmy długi pasek przewijania, który pozwala na spokojne prześledzenie położenia cząstki, krok po kroku, po zakończeniu symulacji. Zmieniając czas za pomocą paska zaobserwuj jak zmienia się położenie cząstki – na wykresie będzie się przemieszczała duża czerwona kropka.

Sprawdź, że zmiana jakiegokolwiek wartości w komórkach B7, B9 lub B11 spowoduje wyzerowanie wartości w tabeli w arkuszu *obliczenia*, co będzie skutkowało również „wyzerowaniem” wykresu – nie będzie widoczny wykres składający się z granatowej linii i małych granatowych kropek.

Jeśli chcesz z powrotem zobaczyć na wykresie pełny tor cząstki (dla tych samych lub zmienionych parametrów), kliknij ponownie na przycisku **start**.

Wykonanie

Do wykonania symulacji ruchu cząstki naładowanej w polu magnetycznym potrzebny będzie pusty plik Excela. Zapisz go na dysku swojego komputera pod nazwą *pole_magnetyczne_symulacja*. Poniżej pola z nazwą pliku znajdziesz pole *Zapisz jako typ* (rys. 3). Wybierz w nim *Skoroszyt programu Excel z obsługą makr*. Wykonanie tej akcji spowoduje, że plik otrzyma rozszerzenie *.xlsm* zamiast standardowego rozszerzenia *.xlsx*.



Rysunek 3. W polu *Zapisz jako typ* wybierz „Skoroszyt programu Excel z obsługą makr”

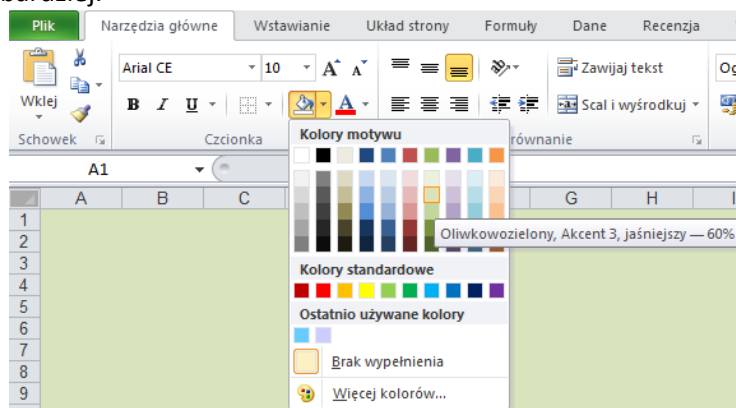
W pliku *pole_magnetyczne_symulacja.xlsm* zmień nazwę pierwszego arkusza na *wykres i parametry*, drugiemu nadaj nazwę *obliczenia*. Pozostałe arkusze możesz usunąć – nie będą nam potrzebne.



Rysunek 4. Nazwy arkuszy w pliku *pole_magnetyczne_symulacja*

Tworzenie symulacji ruchu cząstki rozpoczniemy od przygotowania danych w arkuszu *wykres i parametry*. Wykonaj w nim następujące czynności:

1. Zaznacz cały arkusz (na przykład używając kombinacji klawiszy Ctrl+A) i za pomocą ikony *Kolor wypełnienia* znajdującego się na wstążce na karcie *Narzędzia główne*, w grupie *Czcionka*, zmień kolor tła komórek na dowolny, ale jasny!, odcień koloru niebieskiego, fioletowego lub zielonego (rys. 5) – wybierz ten, który lubisz najbardziej.



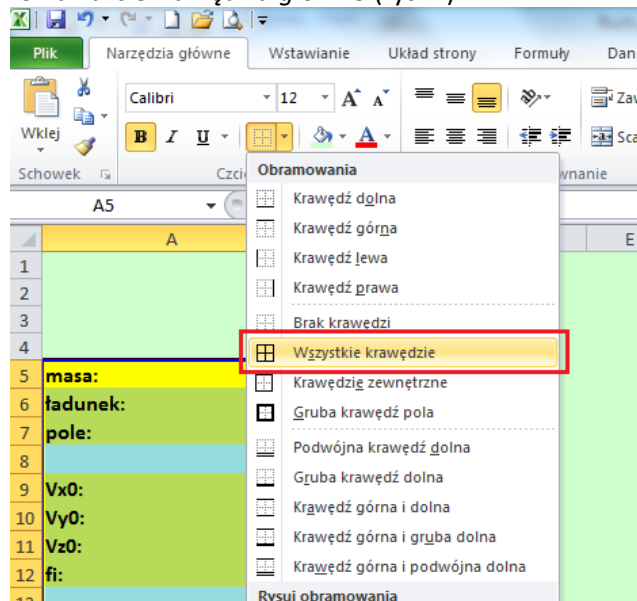
Rysunek 5. Zmiana tła komórek w arkuszu *wykres i parametry*

2. Spójrz na rysunek 2 zaprezentowany powyżej. Komórki z żółtym tłem na rysunku 2 nie zawierają żadnych obliczeń. Znajdują się w nich wyłącznie etykiety tekstowe, czyli opisy różnych parametrów niezbędnych do wykonania symulacji, jednostki. Zajmiemy się teraz ich wprowadzeniem. W tym celu postępuj następująco: do komórki A5 wpisz tekst „masa:”, do komórki A6 – „ładunek:” itd. W ten sposób wpisz do arkusza wszystkie etykiety. Tło komórek zawierających etykiety sformatuj na żółto (rys. 6). Wyśrodkuj teksty w kolumnie C.

	A	B	C	D
1				
2				
3				
4				
5	masa:			
6	ładunek:			
7	pole:		T	
8				
9	Vx0:		cm/ns	
10	Vy0:		cm/ns	
11	Vz0:		cm/ns	
12	fi:		rad	
13				
14	czas obserwacji:		ns	
15	kąt obserwacji:		rad	
16	częstość cyklotronowa:			
17	prędkość poprzeczna:			
18				

Rysunek 6. Wygląd arkusza *wykres i parametry* po wykonaniu pkt. 1 i 2 instrukcji

- Zaznacz zakres A5:C17 i włącz obramowanie postępując się ikonką *Wszystkie krawędzie* na karcie *Narzędzia główne* (rys. 7).



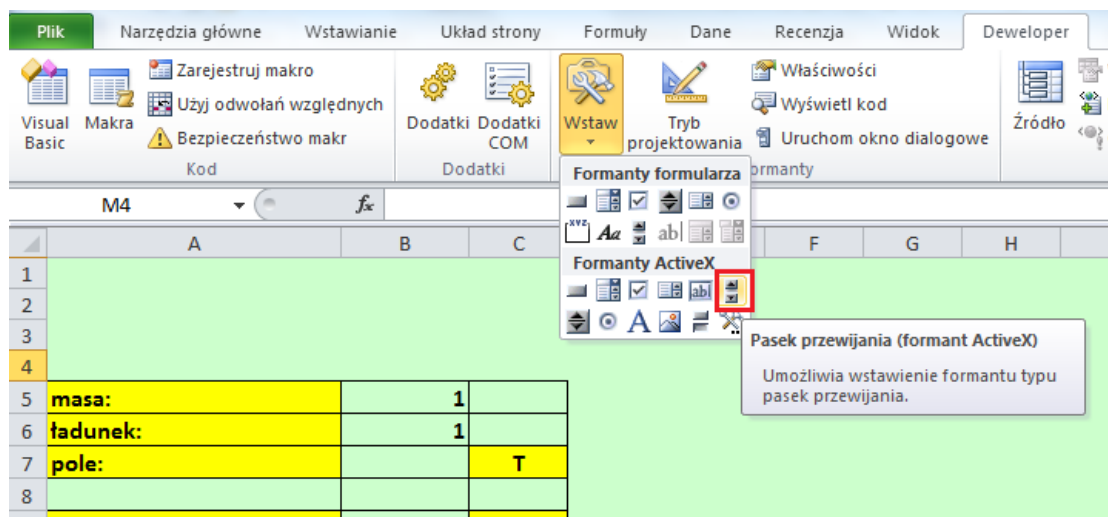
Rysunek 7. Obramowanie tabeli zawierającej parametry

- Teraz wprowadzimy stałe parametry, których, jako użytkownik, nie będziesz mógł modyfikować: do komórki B5 wprowadź wartość 1, B6: 1, B10: 0, B12: 0.
- Poniżej głównej tabelki, zawierającej parametry i formuły, wprowadź jeszcze do odpowiednich komórek dane pokazane na rys. 8. Zmień kolor czcionki, wyśrodkuj tytuły, obramuj tabelki.

	A	B	C	D	E	F
15	kąt obserwacji:	1	rad			
16	częstość cyklotronowa:					
17	prędkość poprzeczna:					
18						
19						
20	t, ns	x, m	y, m	z, m	zp, m	
21	1					
22						
23						
24						
25						
26						
27	t [ns]	x, m	y, m	z, m	zp, m	
28						
29						
30						
31						
32						
33	moment obserwacji t=	0 [ns]				
34						
35	pole	Vx	Vz			
36	25	20	2			
37						

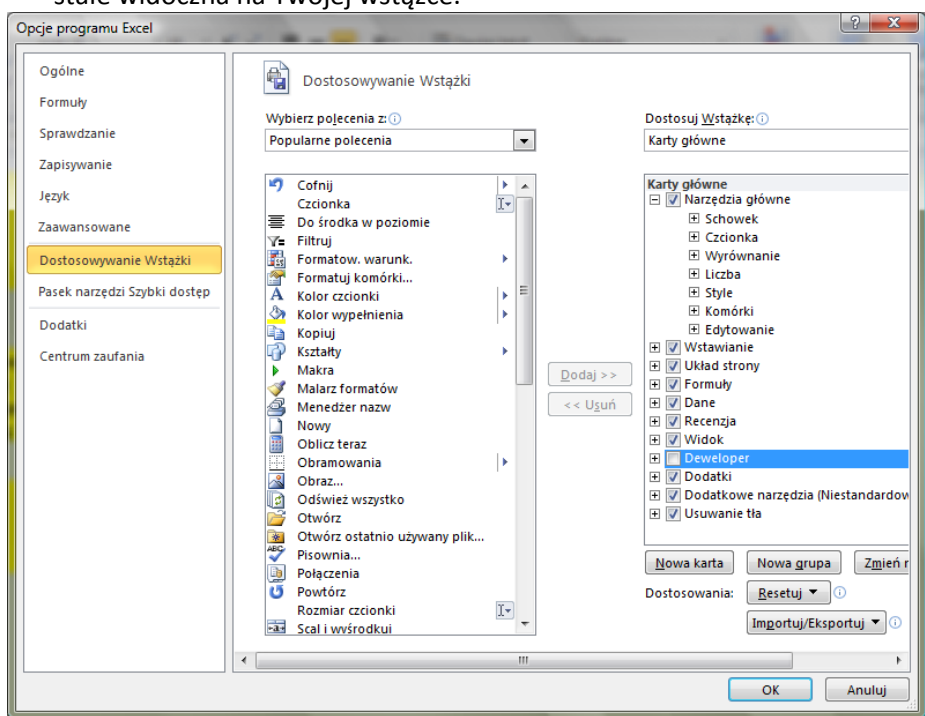
Rysunek 8. Pozostałe elementy arkusza

- Spójrz jeszcze raz na rys. 2. – poniżej wykresu znajduje się pasek przewijania. Postaraj się wstawić go mniej-więcej tym samym miejscu, co w pliku wzorcowym (np. w wierszu 33, w zakresie kolumn od D do Q). W tym celu na karcie *Developer* (jeśli nie widzisz jej na swojej wstążce, w następnym punkcie opiszemy jak ją wyświetlić), w grupie *Formanty* znajdź polecenie *Wstaw* i z wewnętrznej listy wybierz formant *Pasek przewijania (formant ActiveX)* (rys. 9). „Narysuj” pasek w odpowiednim miejscu arkusza.



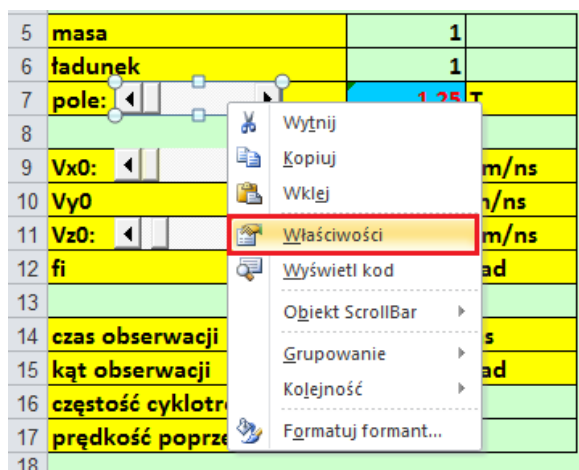
Rysunek 9. Wstawianie paska przewijania

- Uwaga:** jeżeli na wstążce nie ma karty *Deweloper*, włącz ją w następujący sposób: wybierz polecenie *Plik*, a następnie *Opcje*. W oknie *Opcje programu Excel* kliknij *Dostosowywanie wstążki*. Po prawej stronie okna znajdziesz spis wszystkich kart wstążki. Włącz wyświetlanie karty *Deweloper* (rys. 10). Od tej pory będzie ona na stałe widoczna na Twojej wstążce.



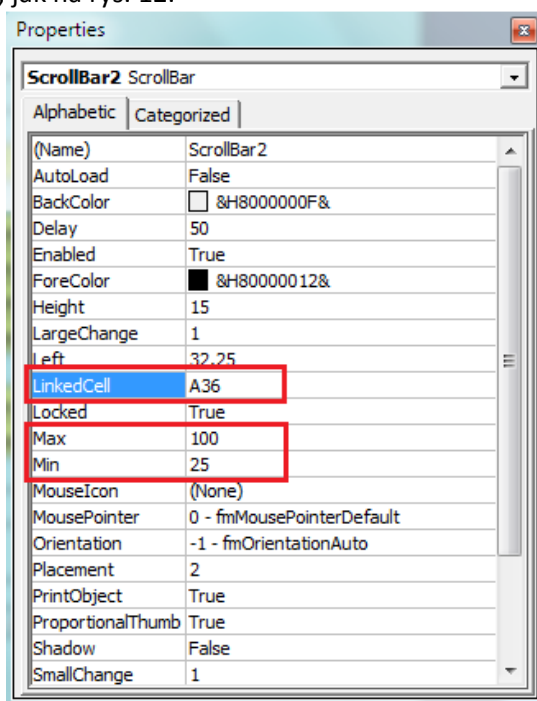
Rysunek 10. Włączanie karty *Deweloper*

- W podobny sposób „narysuj” pozostałe paski przewijania do sterowania indukcją pola magnetycznego (komórka A7), oraz składowymi prędkościami (komórki A9 i A11). Paski na razie nie działają. Zajmiemy się teraz ich aktywowaniem.
- Na karcie *Deweloper* znajdź przycisk *Tryb projektowania*. Upewnij się, że jest włączony. Następnie kliknij prawym klawiszem myszki na pasku przewijania z komórki A7 i z menu podręcznego wybierz *Właściwości* (rys. 10).



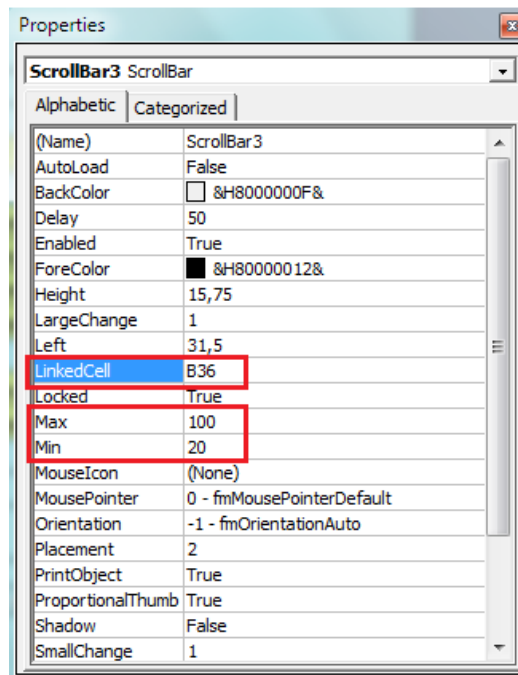
Rysunek 11. Wybór właściwości paska przewijania z komórki A7

10. W oknie właściwości (*Properties*) wprowadź wartości w polach *LinkedCell* (tzn. komórka połączona, czyli taka, w której będzie się wyświetlała wartość wybrana na pasku), *Max* (wartość maksymalna, jaką można wybrać) oraz *Min* (wartość minimalna), jak na rys. 12.



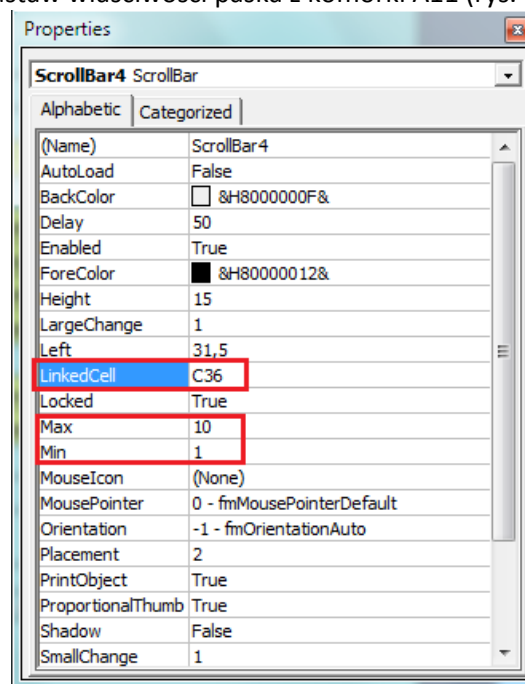
Rysunek 12. Ustawienia kluczowych wartości w oknie właściwości paska przewijania z komórki A7

11. Postępując analogicznie ustaw wartości w oknach właściwości paska z komórki A9 (rys. 13).



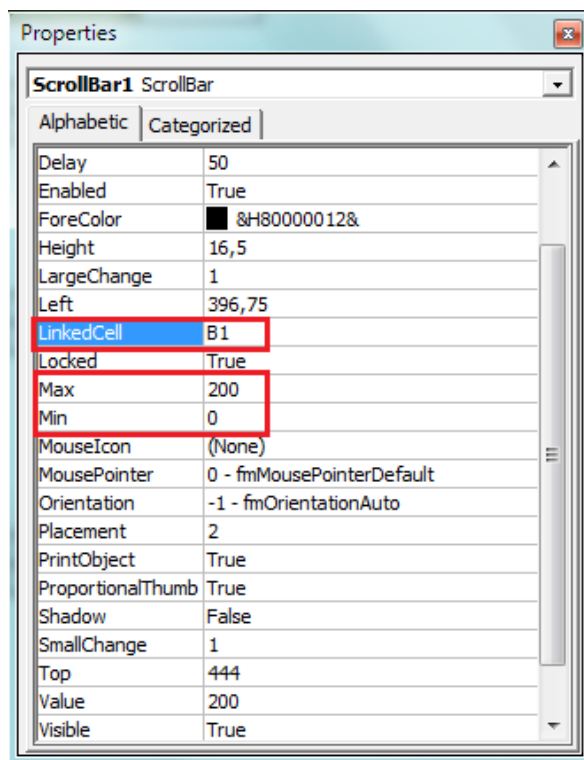
Rysunek 13. Ustawienia kluczowych wartości w oknie właściwości paska przewijania z komórki A9

12. Następnie ustaw właściwości paska z komórki A11 (rys. 14).



Rysunek 14. Ustawienia kluczowych wartości w oknie właściwości paska przewijania z komórki A11

13. Na dole naszego arkusza znajduje się jeszcze jeden pasek przewijania (wstawiony jako pierwszy). Wyświetl jego właściwości i wprowadź ustawienia jak na rys. 15.



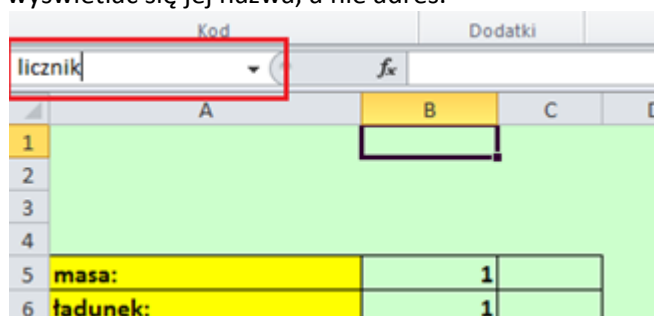
Rysunek 15. Ustawienia kluczowych wartości w oknie właściwości długiego paska przewijania znajdującego się na dole arkusza

14. Kolejnym krokiem tworzenia symulacji powinno być wprowadzenie formuł do arkusza. Najpierw jednak, dla wygody, niektórym komórkom nadamy *nazwy*.

Uwaga: Odwołanie do komórki za pomocą nadanej jej własnoręcznie *nazwy* jest alternatywą do odwołania się do niej za pomocą adresu. Będziesz mieć wybór – możesz w formułach odwoływać się do komórki za pomocą jej adresu lub za pomocą *nazwy*.

15. Komórce B1 nadamy nazwę „licznik” (w trakcie wykonywania symulacji będzie wstawiany do niej kolejny numer kroku od 0 do 200).

Uaktywnij komórkę B1 (kliknij na niej). Zwróć uwagę, że adres aktywnej komórki pojawił się w tzw. *Polu nazwy* (pole otoczone czerwoną obwódką na rysunku 16). Kliknij w *Polu nazwy* i wpisz nazwę, którą chcesz nadać komórce, na przykład „licznik”. Na koniec naciśnij **Enter**. Teraz, po uaktywnieniu komórki B1 w *Polu nazwy* będzie wyświetlać się jej nazwa, a nie adres.



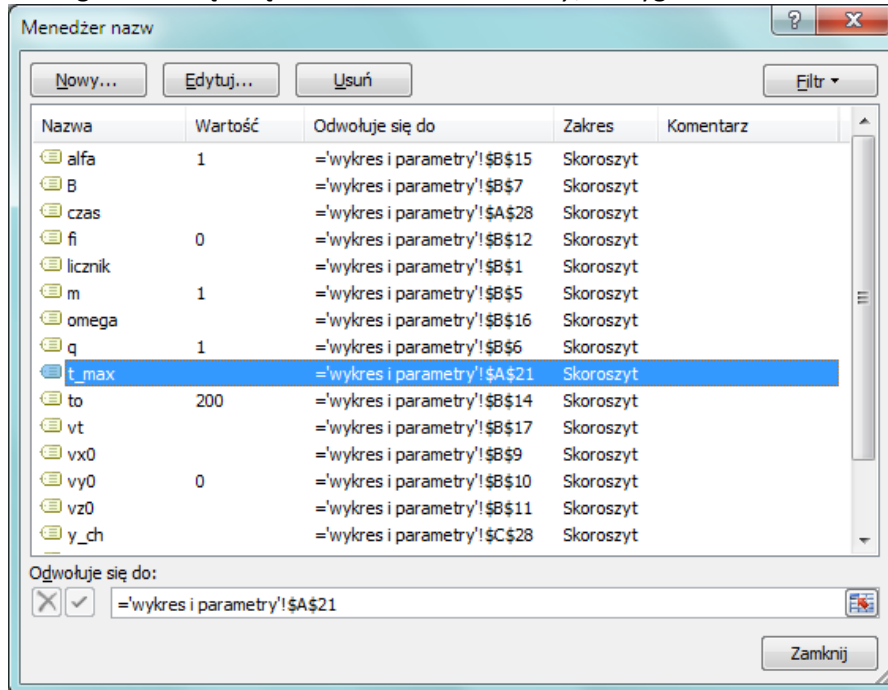
Rysunek 16. Lokalizacja *Pola nazwy*, za pomocą którego nadajemy *nazwy* komórkom

16. Zaznacz komórkę B5 i postępując w sposób opisany w poprzednim punkcie nadaj jej nazwę „m”.

Uwaga: jeśli w przyszłości zechcesz nadać komórce nazwę wielocłonową, nie używaj spacji! Zamiast niej możesz użyć znaku podkreślenia, np. *kąt_alfa*, lub wpisać nazwę w taki sposób: *KątAlfa* – bez odstępów, ale zachowując czytelność nazwy.

17. W podobny sposób nadaj nazwy pozostałym komórkom, i tak, komórkę B6 nazwij „q”, B7 – „B”, B9 – „vx0”, B10 – „vy0”, B11 – „vz0”, B12 – „fi”, B14 – „to”, B15 – „alfa”, B16 – „omega”, B17 – „vt”, dodatkowo nadamy nazwy komórkom (na razie pustym) A21 – „t_max”, A28 – „czas”, C21- „y_max”, D21 – „z_max”, C28 – „y_ch”, D28 – „z_ch”.

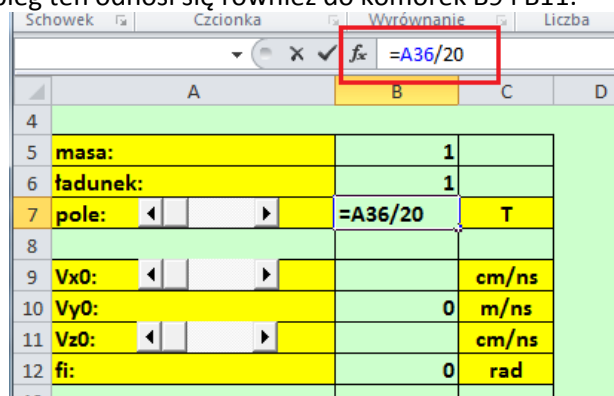
Uwaga: jeśli pomylisz się definiując nazwy, na wstążce na karcie *Formuły* w grupie *Nazwy zdefiniowane* znajdziesz ikonę *Menedżer nazw*. Otwiera ona okno z listą wszystkich nazw zdefiniowanych w bieżącym skoroszycie. Możesz w nim – „w razie czego” – usunąć błędnie zdefiniowane nazwy, skorygować ich odwołanie itp.



Rysunek 17. Okno *Menedżera nazw* z listą zdefiniowanych nazw, ich bieżące wartości (u Ciebie na tym etapie wykonywania ćwiczenia większość wartości będzie równa zero), odwołania do komórek itp.

18. Najwyższa pora zająć się obliczeniami! Wprowadź do komórki B7 formułę, jak na rysunku 18. Spójrz na swój *Pasek formuły* i sprawdź czy formuła, którą wpisałeś, jest prawidłowa. W komórce tej będzie się wyświetlała wartość z komórki A36, ale tak przeskalowana, byśmy uzyskali za pomocą paska przewijania mogli uzyskać zakres wartości od 1,25 do 5 (zmiana o 0,05).

Uwaga: musieliśmy zastosować taki zabieg, aby uzyskać ułamkowe zmiany wartości pola magnetycznego, gdyż formant *Pasek przewijania* obsługuje wyłącznie wartości całkowite! Zabieg ten odnosi się również do komórek B9 i B11.



Rysunek 18. Formuła w komórce B7

19. Postępując analogicznie, do komórki B9 wprowadź formułę jak na rys. 19.

$$f_x = B36/20$$

Rysunek 19. Formuła w komórce B9

20. Do komórki B11 wpisz natomiast:

$$f_x = C36/10$$

Rysunek 20. Formuła w komórce B11

21. Sformatuj tło komórek B7, B9 i B11 na kolor niebieski.

22. W komórce B16 obliczymy częstość cyklotronową:

$$f_x = 0,0958 * q * B / m$$

Rysunek 21. Formuła w komórce B16

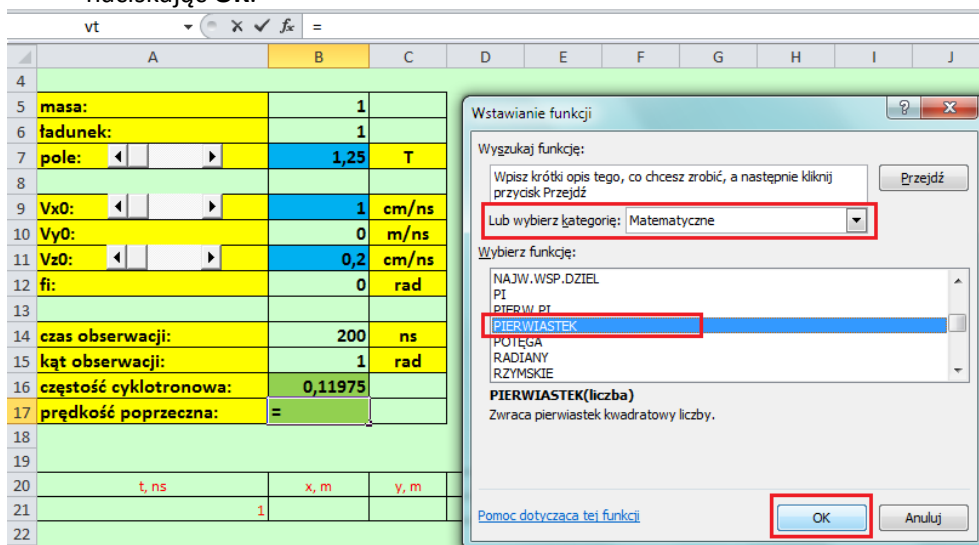
23. Sformatuj tło komórek B16 i B17 na kolor zielony.

24. Do komórki B17 wprowadź formułę na prędkość poprzeczną (rys. 22). Zapis ^ oznacza podnoszenie do kwadratu, natomiast PIERWIASTEK() oznacza funkcję arkuszową obliczającą pierwiastek kwadratowy.

$$f_x = \text{PIERWIASTEK}(v_x0^2 + v_y0^2)$$

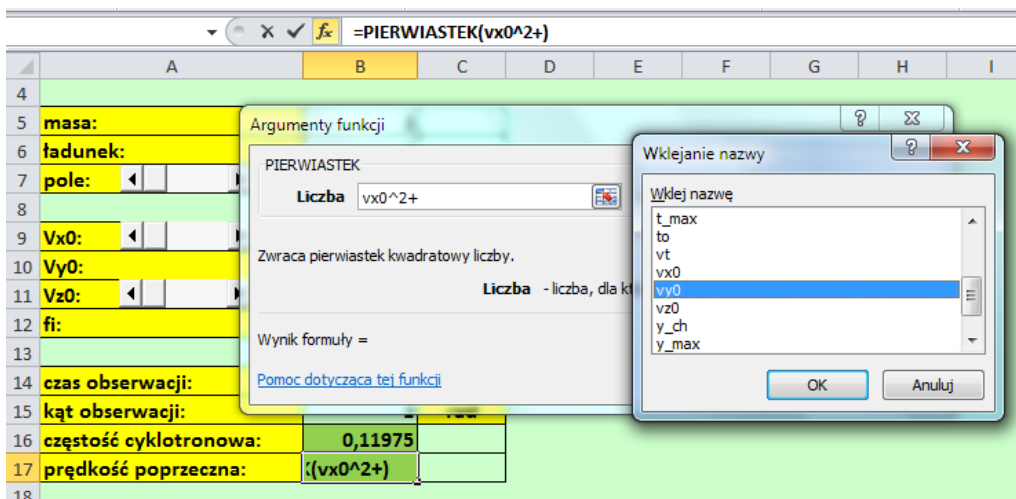
Rysunek 22. Formuła w komórce B17

25. **Uwaga:** Funkcję PIERWIASTEK możesz wpisać ręcznie lub wstawić ją korzystając z narzędzi do wstawiania funkcji. Przybliżymy tutaj ten drugi sposób. Aby przygotować formułę z rysunku 22, zaznacz komórkę B17, a następnie kliknij na ikonie *Wstaw funkcję* f_x , którą znajdziesz z lewej strony *Paska formuły*. W oknie *Wstawianie funkcji* wybierz kategorię *Matematyczne*. Otrzymasz spis funkcji należących do tej kategorii. Znajdź i wybierz funkcję PIERWIASTEK(), a następnie w oknie funkcji wprowadź odpowiednie argumenty. Gotową formułę zatwierdź naciskając **OK**.



Rysunek 23. Tworzenie formuły z użyciem okna *Wstawianie funkcji*

26. Tworząc formułę, możesz adresy komórek (lub nazwy) wpisywać ręcznie lub klikać na odpowiednich komórkach – wówczas ich adresy (nazwy) w formule pojawią się automatycznie. W przypadku nazw możesz alternatywnie skorzystać z okna *Wklejanie nazwy*, które zawiera spis wszystkich nazw występujących w skoroszybie. Okno *Wklejanie nazwy* wywołasz naciskając klawisz funkcyjny **F3**.



Rysunek 24. Tworzenie formuły z użyciem okna *Wklejanie nazwy*

27. Kontynuując wprowadzanie niezbędnych formuł do arkusza *wykres i parametry* wprowadź do komórki B21 formułę jak na rys. 25. COS() jest funkcją arkuszową obliczającą kosinus zadanego kąta, pozostałe argumenty to parametry z „nazwanych” komórek arkusza – wpisz je „z ręki” lub wstaw posługując się oknem *Wklejanie nazwy*.

Uwaga: W wierszu 21. wyznaczmy maksymalne wartości parametrów, które w dalszej części ćwiczenia posłużą do wyskalowania wykresu.

$$f_x = (-vt * \text{COS}(\text{omega} * t_{\text{max}} * to + fi)) / \text{omega}$$

Rysunek 25. Formuła w komórce B21

28. Postępując podobnie, jak w poprzednim punkcie, wprowadź formułę do komórki C21 (SIN() jest funkcją arkuszową pozwalającą obliczyć sinus kąta):

$$f_x = (-vx0 + vt * \text{SIN}(\text{omega} * t_{\text{max}} * to + fi)) / \text{omega}$$

Rysunek 26. Formuła w komórce C21

29. A następnie do komórki D21:

$$f_x = vz0 * t_{\text{max}} * to$$

Rysunek 27. Formuła w komórce D21

30. Oraz E21:

$$f_x = z_{\text{max}} - y_{\text{max}} * \text{SIN}(\text{alfa})$$

Rysunek 28. Formuła w komórce E21

31. Wprowadzimy teraz formuły do komórek w wierszu 28. W komórkach wiersza 21. znajdują się bieżące wartości toru cząstki, które w dalszej części ćwiczenia zostaną przedstawione na wykresie. Do komórki A28 wpisz:

$$f_x = \text{licznik} / 200$$

Rysunek 29. Formuła w komórce A28

32. Do komórki B28:

$$f_x = (-vt * \text{COS}(\text{omega} * \text{czas} * to + fi)) / \text{omega}$$

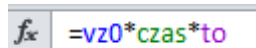
Rysunek 30. Formuła w komórce B28

33. Do komórki C28:

$$f_x = (-vx0 + vt * \text{SIN}(\text{omega} * \text{czas} * to + fi)) / \text{omega}$$

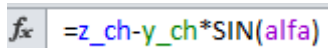
Rysunek 31. Formuła w komórce C28

34. Do komórki D28:


$$f_x \quad =vz0*czas*to$$

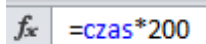
Rysunek 32. Formuła w komórce D28

35. Do komórki E28:


$$f_x \quad =z_ch-y_ch*SIN(alfa)$$

Rysunek 33. Formuła w komórce E28

36. Wreszcie, w komórce B33, powinna się znaleźć następująca formuła – będzie się tu wyświetlać bieżący czas symulacji (od wartości 0 do to):

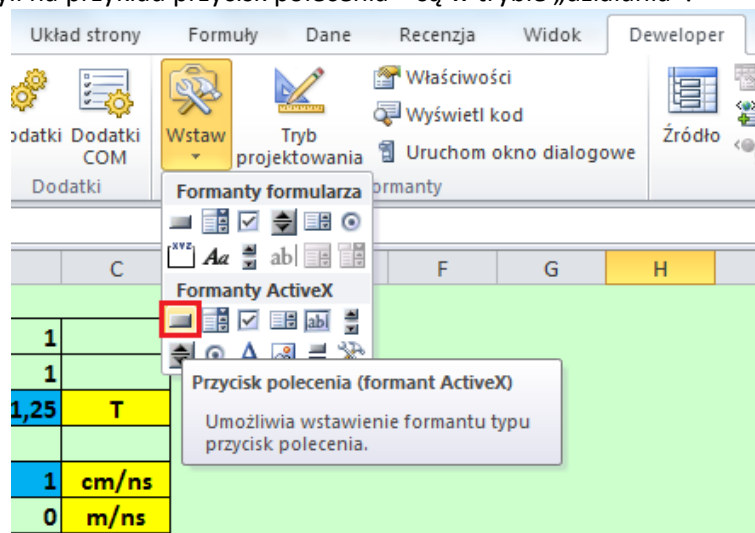

$$f_x \quad =czas*200$$

Rysunek 34. Formuła w komórce B33

37. Do tej pory zajmowaliśmy się wyłącznie arkuszem *wykres i parametry*. Teraz przygotowujemy dane w arkuszu *obliczenia*, w zakresie komórek A1:B201, które będą bazą do wykonania wykresu obrazującego ruch ciała w rzucie pionowym. Dane w arkuszu *obliczenia* będą wyliczane za pomocą makra (dla parametrów wprowadzonych przez użytkownika w arkuszu *wykres i parametry*).

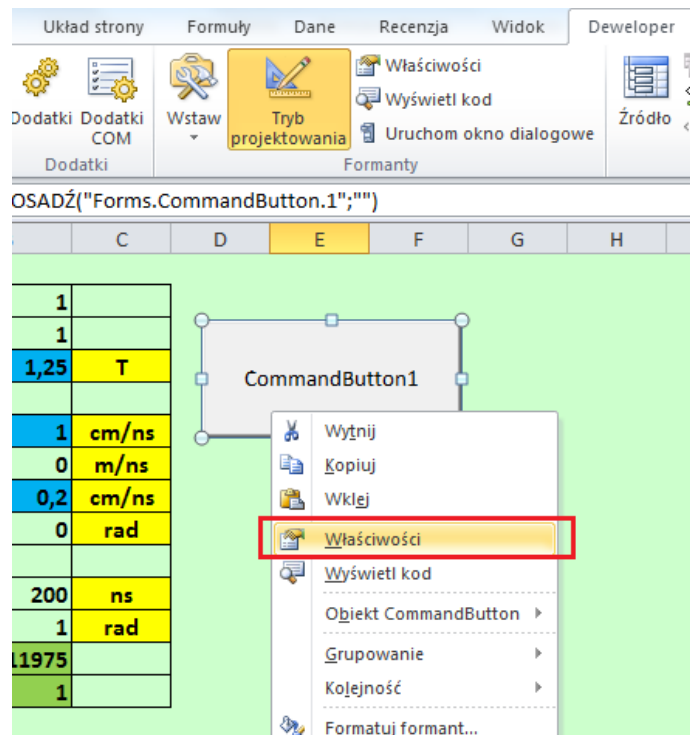
Przygotowania rozpoczniemy od wstawienia przycisku w arkuszu *wykres i parametry*, który będzie uruchamiał makro. Aby wstawić przycisk przejdź na kartę *Deweloper*, w sekcji *Formanty* kliknij na ikonie *Wstaw* i z wewnętrznej listy, z grupy *Formanty ActiveX*, wybierz *Przycisk polecenia* (rys. 35). „Narysuj” przycisk obok komórek z parametrami do sterowania symulacją, gdzieś w zakresie wierszy 6-9 i kolumn D-F. Zwróć uwagę czy na karcie *Deweloper* jest włączony przycisk *Tryb projektowania*. Jeśli nie – włącz go!

Uwaga: Kiedy przycisk *Tryb projektowania* na karcie *Deweloper* jest włączony możesz bez trudu modyfikować właściwości formantów, kiedy jest wyłączony – formanty, czyli na przykład przycisk polecenia – są w trybie „działania”!



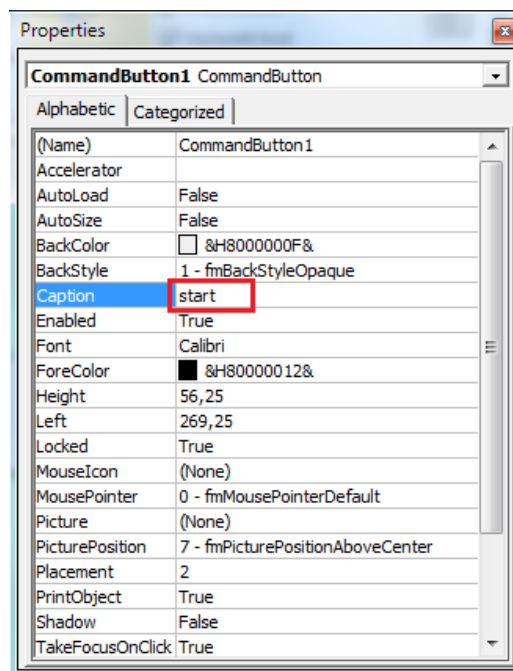
Rysunek 35. Wstawianie *Przycisku polecenia*, który będzie uruchamiał makro

38. Kliknij prawym klawiszem myszki na wstawionym przycisku i z menu podręcznego wybierz *Właściwości* (rys. 36).



Rysunek 36. Menu podręczne Przycisku polecenia

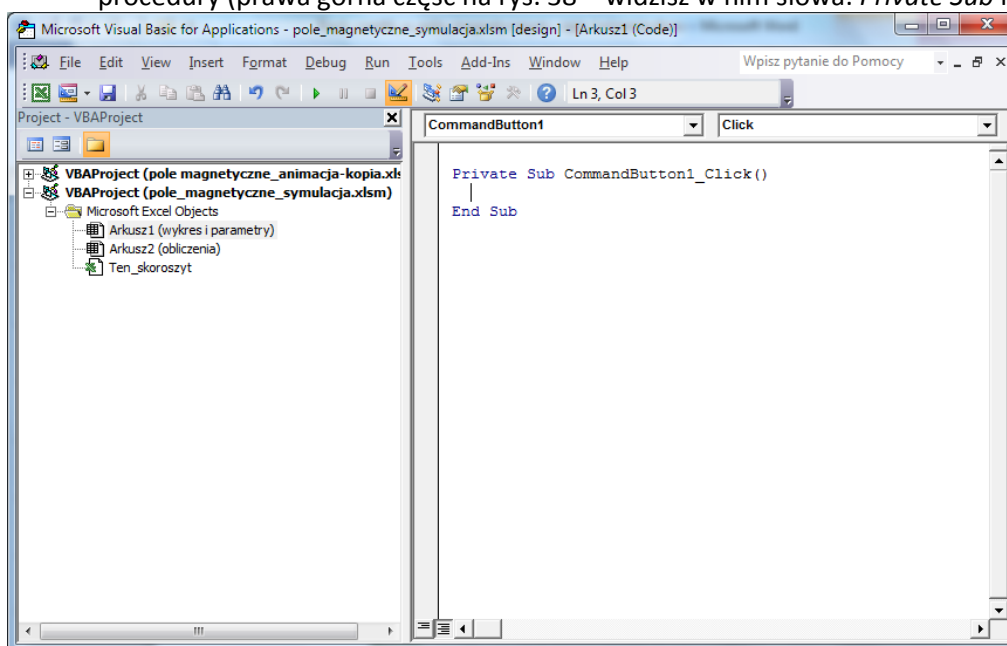
39. W oknie *Properties*, którego fragment widzisz na rysunku 37, w sekcji *Caption* wpisz słowo „start”. Zamknij okno właściwości (nie musisz zatwierdzać wprowadzonych zmian). Wykonanie tych czynności spowoduje, że na przycisku pojawi się słowo „start”.



Rysunek 37. Właściwości (*Properties*) Przycisku polecenia

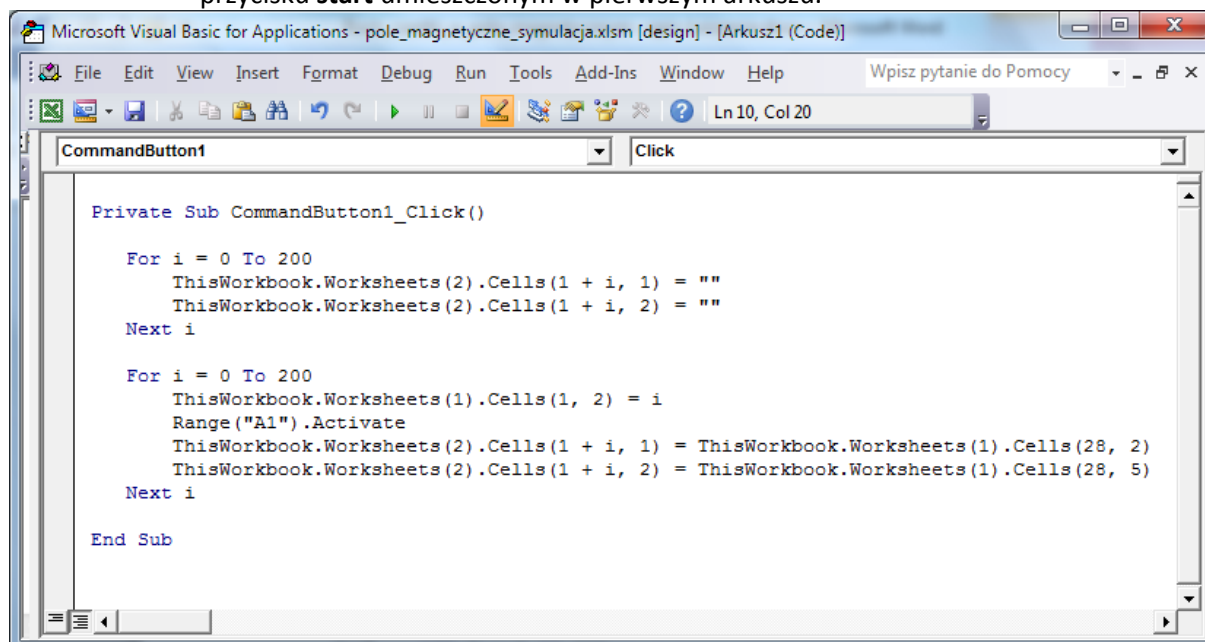
40. Upewnij się, czy przycisk *Tryb projektowania* na karcie *Deweloper* jest włączony. Jeśli nie jest – włącz go! Kliknij teraz szybko dwa razy na przycisku **start**. Powinien pojawić się edytor kodu VBA (rys. 38). Okno edytora domyślnie składa się z kilku części, z których – z naszego punktu widzenia – najistotniejsze są: okno projektu widoczne

po lewej stronie (*Project – VBA Project*) oraz okno służące do wpisywania kodu procedury (prawa górna część na rys. 38 – widzisz w nim słowa: *Private Sub* itd.).



Rysunek 38. Okno edytora VBA

41. Aby dokładnie zrozumieć kod makr, które zaprezentujemy, i dowiedzieć się „o co w ogóle chodzi z tym VBA” powinieneś teraz zapoznać się ze specjalnym *Dodatkiem* zamieszczonym na końcu niniejszej instrukcji.
42. W części służącej do programowania, w miejscu, gdzie widoczny jest kursor, wpisz kod przedstawiony na rys. 39. Jest to procedura, która uruchomi się po kliknięciu na przycisku **start** umieszczonym w pierwszym arkuszu.

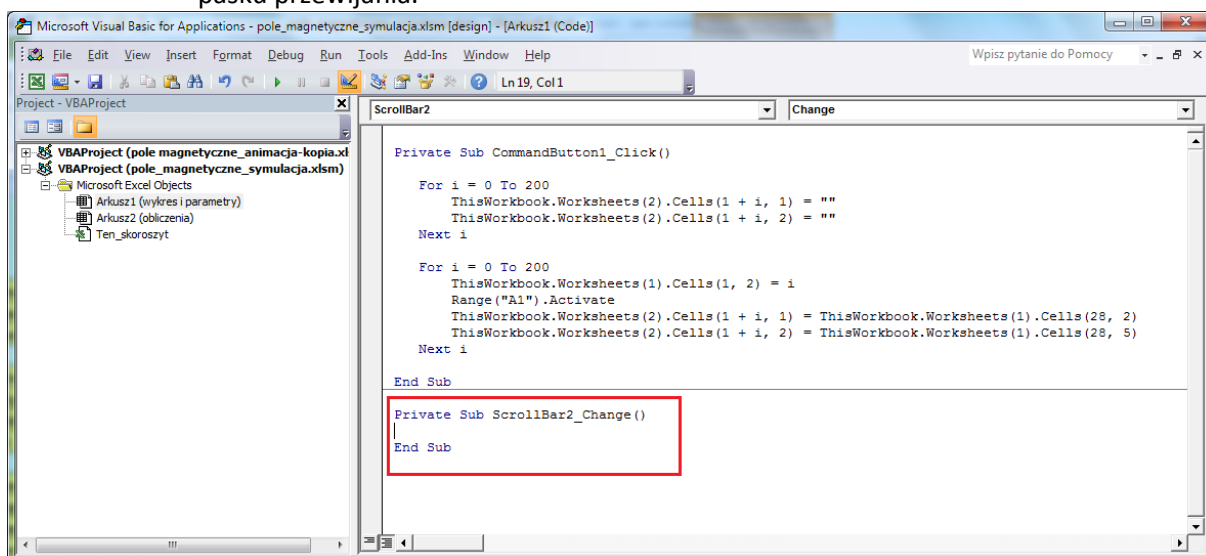


Rysunek 39. Kod procedury *CommandButton1_Click()*, który przygotowuje wartości kolejnych położenia cząstki naładowanej poruszającej się w polu magnetycznym w czasie symulacji trwającej 200 ns.

W procedurze z rys. 39 widzisz dwie pętle *For ... Next*. Pierwsza z nich wykona się 201 razy (zmienna *i* – licznik pętli – będzie przyjmowała wartości od zera do 200, w każdym kroku licznik będzie zwiększał się o 1). W pętli zostaną wykonane następujące czynności: zostaną wyzerowane komórki w kolumnach A i B, wierszach od 1 do 201, w drugim arkuszu bieżącego skoroszytu.

Druga pętla *For ... Next* wykona się taką samą ilość razy, jak pierwsza. Pierwsza instrukcja w tej pętli spowoduje, że do komórki o numerze wiersza 1 i numerze kolumny 2 (chodzi oczywiście o komórkę B1) pierwszego arkusza (czyli *wykres i parametry*) będzie podstawiana bieżąca wartość licznika. Zwróć uwagę, że w tej komórce znajduje się informacja o kolejnym numerze przedziału trwania symulacji (cała symulacja została podzielona na 200 przedziałów czasowych). Druga instrukcja oznacza po prostu: „zaznacz komórkę A1 w aktywnym arkuszu”. Instrukcja trzecia to podstawianie kolejno do komórek A1:B201 bieżących wartości współrzędnych *x* obliczanych w arkuszu *wykres i parametry* w komórce B28. Czwarta instrukcja podstawia bieżącą wartość współrzędnej *zp* położenia ciała, która jest wyliczana w komórce E28 w arkuszu pierwszym.

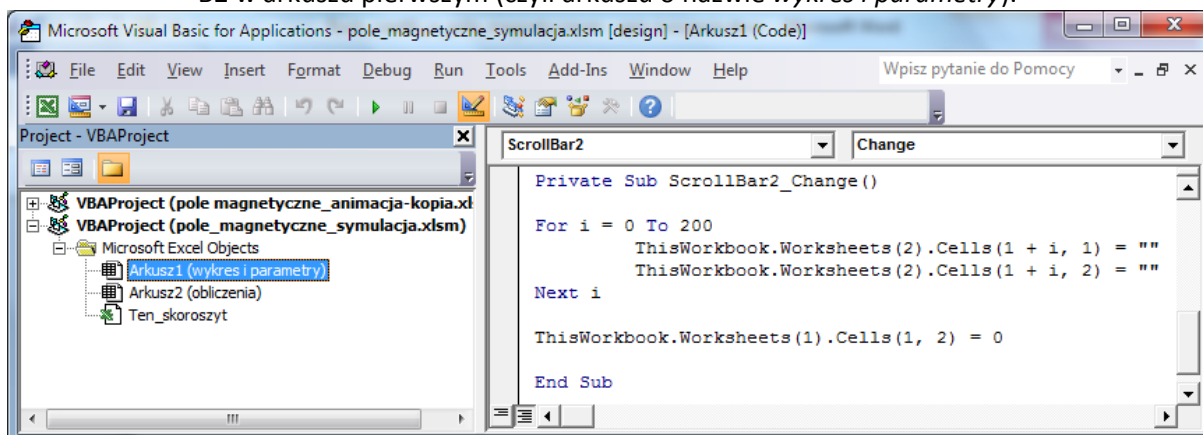
43. Po przepisaniu procedury z rys. 39, wróć do arkusza *wykres i parametry*.
Uwaga: W każdej chwili możesz z edytora VBA przejść do skoroszytu Excela (bez zamykania go) używając *Paska zadań* (na dole monitora).
44. Wyłącz przycisk *Tryb projektowania* na karcie *Deweloper*, a następnie kliknij na wstawionym przez siebie przycisku „start”. Zaobserwuj działanie makra (w komórce B1 pojawi się wartość 200, w arkuszu *obliczenia* w kolumnach A i B pojawią się wartości – położenia cząstki). W dalszej części ćwiczenia wartości te przedstawimy na wykresie.
45. Będziemy chcieli teraz oprogramować paski przewijania, aby po kliknięciu na nich wyzerowały się („zresetowały”) wartości w arkuszu *obliczenia*. Włącz ponownie tryb projektowania za pomocą przycisku *Tryb projektowania* dostępnego na karcie *Deweloper*.
46. Kliknij dwa razy na pasku przewijania znajdującym się w komórce A7. Ta czynność powinna przenieść Cię do edytora VBA i spowodować umieszczenie w nim procedury *ScrollBar2_Change()*, w której wpiszesz kod, jaki wykona się po zmianie wartości na pasku przewijania.



Rysunek 40. Procedura zdarzeniowa *ScrollBar2_Change*, która pojawia się po dwukrotnym kliknięciu na pasku przewijania z komórki A7

47. Pomiędzy słowami kluczowymi (*Sub* i *End Sub*) procedury *ScrollBar2_Change()* wprowadź kod zamieszczony na rysunku 41. Pętla *For ... Next* jest identyczna, jak w

przypadku procedury *CommandButton1_Click()*. Jej zadaniem jest wyzerowanie tabelki współrzędnych położenia cząstki w arkuszu *obliczenia*. Instrukcja *ThisWorkbook.Worksheets(1).Cells(1,2) = 0* wstawia dodatkowo wartość 0 do komórki B2 w arkuszu pierwszym (czyli arkuszu o nazwie *wykres i parametry*).



Rysunek 41. Kod procedury *ScrollBar2_Change*

48. Podobnie powinny „reagować” pozostałe paski przewijania. Przejdź do arkusza *wykres i parametry* i kliknij dwa razy na pasku przewijania znajdującym się w komórce A9 (zmiana współrzędnej *x*). Cursor zostanie przeniesiony do edytora VBA i znajdzie się w procedurze zdarzeniowej *ScrollBar3_Change()*. Wpisz do niej kod taki sam, jak w przypadku procedury *ScrollBar2_Change()* (możesz go skopiować i wkleić). Podobnie oprogramuj pasek przewijania z komórki A11 (zmiana współrzędnej *z*) – powinna pojawić się procedura *ScrollBar4_Change()* – wklej/wpisz do niej taki sam kod, jak w przypadku dwóch poprzednich.
49. Po wykonaniu opisanych wyżej czynności okno edytora VBA powinno zawierać cztery procedury i wyglądać tak, jak na rys. 42.

```
Ln 38, Col 47
ScrollBar4 Change
Private Sub CommandButton1_Click()
    For i = 0 To 200
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ""
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ""
    Next i

    For i = 0 To 200
        ThisWorkbook.Worksheets(1).Cells(1, 2) = i
        Range("A1").Activate
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ThisWorkbook.Worksheets(1).Cells(28, 2)
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ThisWorkbook.Worksheets(1).Cells(28, 5)
    Next i
End Sub

Private Sub ScrollBar2_Change()
    For i = 0 To 200
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ""
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ""
    Next i
    ThisWorkbook.Worksheets(1).Cells(1, 2) = 0
End Sub

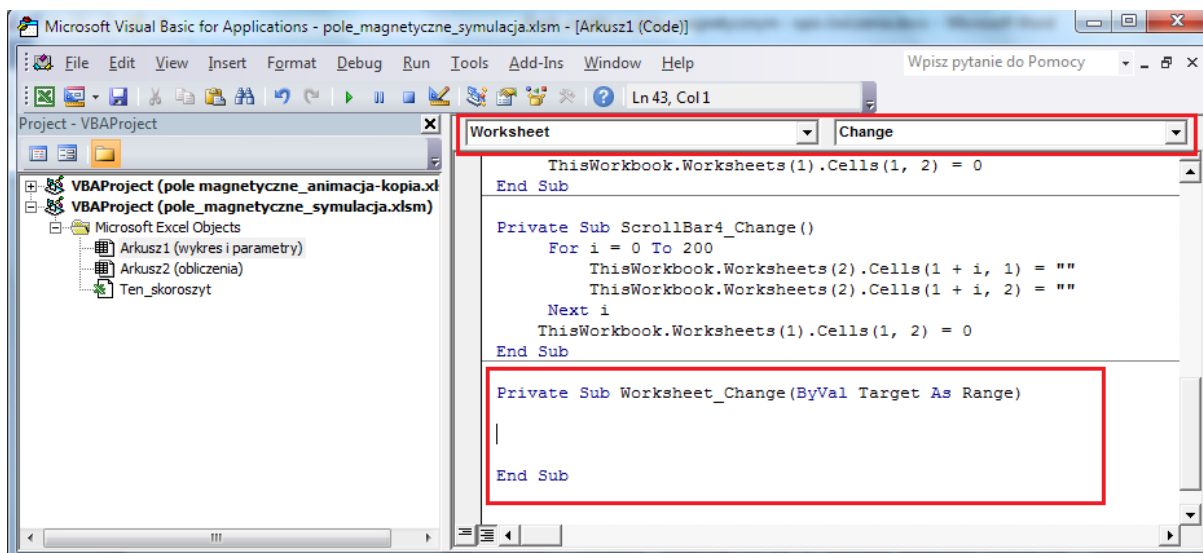
Private Sub ScrollBar3_Change()
    For i = 0 To 200
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ""
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ""
    Next i
    ThisWorkbook.Worksheets(1).Cells(1, 2) = 0
End Sub

Private Sub ScrollBar4_Change()
    For i = 0 To 200
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ""
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ""
    Next i
    ThisWorkbook.Worksheets(1).Cells(1, 2) = 0
End Sub
```

Rysunek 42. Widok edytora VBA po oprogramowaniu pasków przewijania

50. Ustaw kursor tekstowy na końcu, poniżej słów kluczowych *End Sub* ostatniej procedury i za pomocą list wyboru znajdujących się ponad edytorem tekstu wybierz pozycję *Worksheet* z listy po lewej stronie i *Change* – po prawej stronie (rys. 43). W wyniku wykonania tych czynności na dole okna powinna pojawić się procedura *Worksheet_Change()*. Zadaniem tej procedury, która uruchomi się automatycznie po zmianie wartości w którejkolwiek z komórek sformatowanych na zielono w pierwszym arkuszu, będzie wyzerowanie wartości komórek w zakresie A1:B201 w arkuszu drugim (*obliczenia*).

Uwaga: jeżeli przy okazji wstawiania procedury *Worksheet_Change()* w edytorze pojawi się jakaś inna procedura, niż ta, którą miałeś zamiar użyć, zaznacz ją (tak, jak w dowolnym edytorze tekstu) i skasuj klawiszem **Delete**.



Rysunek 43. Widok edytora VBA po wstawieniu procedury *Worksheet_Change*

51. Poprzednie procedury nie zawierały żadnych argumentów (spojrz na nawiasy po nazwie procedury), tutaj występuje zmienna *Target*, która jest typu zakres (*Range*). Zaprogramujemy procedurę tak, aby „reagowała” wyłącznie na zmianę wartości w trzech komórkach z niebieskim tłem.

```
Private Sub Worksheet_Change(ByVal Target As Range)

Dim t1 As String
t1 = Target.Address

If (t1 = "$B$7" Or t1 = "$B$9" Or t1 = "$B$11") Then


For i = 0 To 200
ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = ""
ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ""
Next i

ThisWorkbook.Worksheets(1).Cells(1, 2) = 0

End If

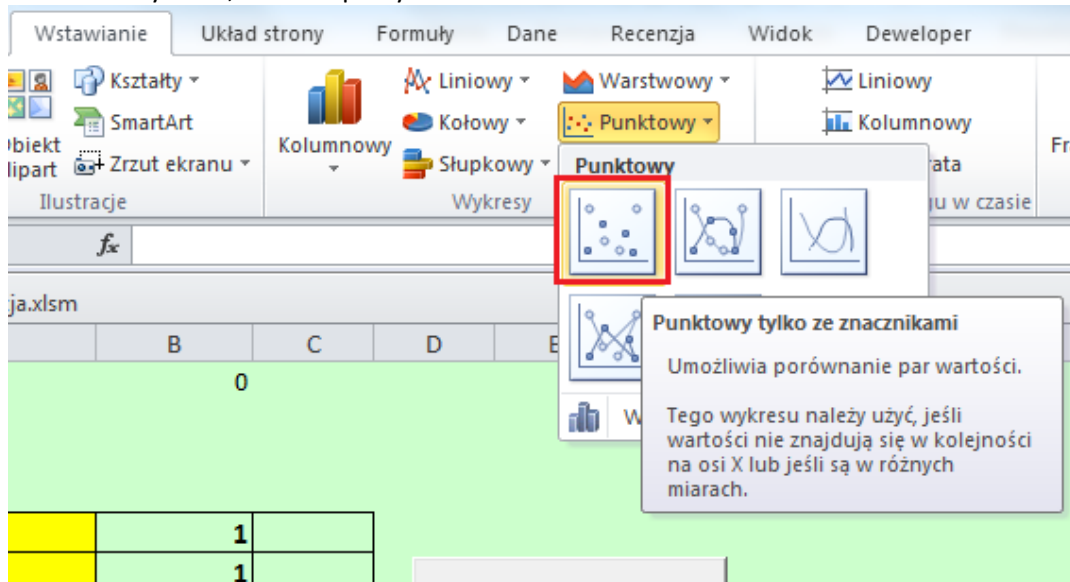
End Sub
```

Rysunek 44. Kod procedury *Worksheet_Change*

52. W pierwszej linii kodu została zadeklarowana zmienna *t1* jako zmienna typu tekstowego (*String*). Instrukcja *t1 = Target.Address* oznacza: „zmiennej *t1* przypisz adres komórki (w domyśle, w której nastąpiła zmiana wartości)”. Instrukcja warunkowa *If ... End If* sprawdza czy zmiana wartości nastąpiła w jednej z komórek z zakresu z niebieskim tłem (chodzi o komórki B7 lub B9 lub B11). Jeśli warunek jest spełniony, pętla *For ... Next* (znana nam doskonale z poprzednich procedur) zeruje wartości w tabelce w arkuszu drugim.
53. Po wpisaniu kodu wszystkich procedur, zapisz wszystko – możesz w tym celu użyć standardowej ikony  dostępnej na pasku narzędziowym lub kombinacji klawiszy **Ctrl+S**.
54. **Uwaga:** Zwróć uwagę, że edytor VBA jest odrębnym oknem. Możesz je zamknąć w standardowy sposób. Aby ponownie wywołać okno edytora VBA naciśnij kombinację klawiszy **Alt+F11** (**F11** to klawisz funkcyjny, znajdziesz go w górnej części klawiatury).

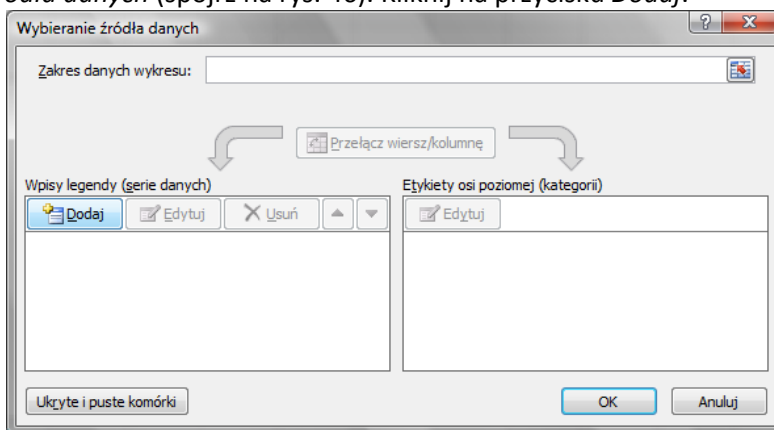
55. Mając wykonane wszystkie obliczenia zajmiemy się przygotowaniem wykresu. Zaznacz dowolną pustą komórkę arkusza *wykras i parametry* znajdująca się poza obszarem danych (na przykład H2).

56. Przejdź na kartę *Wstawianie*. W grupie *Wykresy* wybierz typ *Punktowy*, a następnie podtyp *Tylko ze znacznikami* (rys. 45). W arkuszu powinien pojawić się obszar wykresu, na razie pusty.



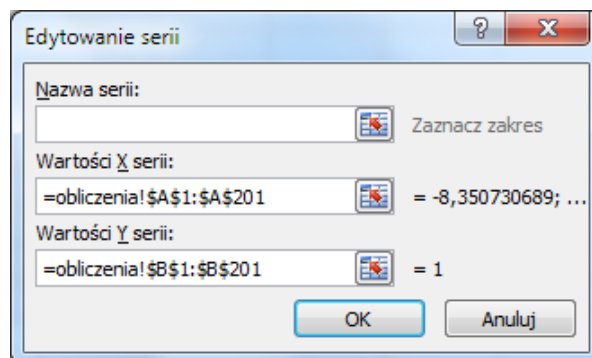
Rysunek 45. Wybór wykresu, typ *Punktowy*

57. Zwróć uwagę czy obszar wykresu jest aktywny. Jeśli tak, przejdź na kartę *Projektowanie* (znajdącą się na końcu wstążki, w grupie *Narzędzia wykresów*) i z grupy poleceń *Dane* wybierz *Zaznacz dane*. Na ekranie pojawi się okno *Wybieranie źródła danych* (spójrz na rys. 46). Kliknij na przycisku *Dodaj*.



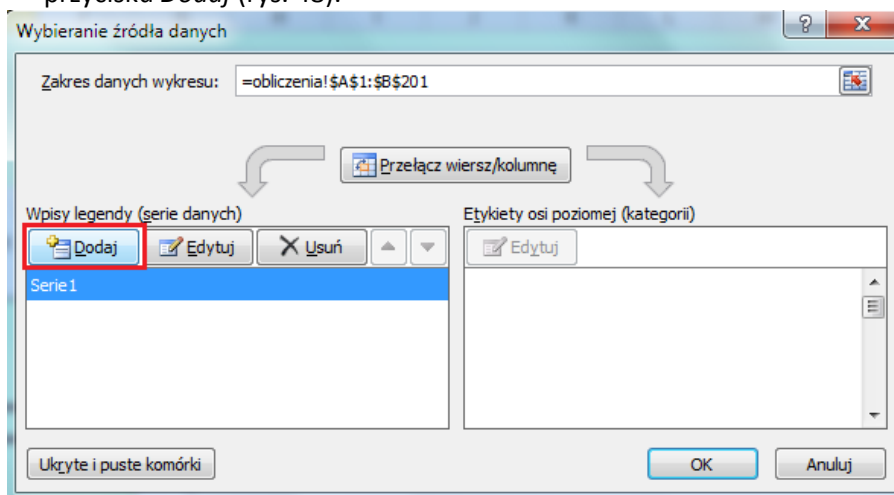
Rysunek 46. Dodawanie serii danych do wykresu

58. W oknie *Edytowanie serii* (rys. 47) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek od A1 do A201. Następnie przestaw kursor do pola *Wartości Y serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek od B1 do B201. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



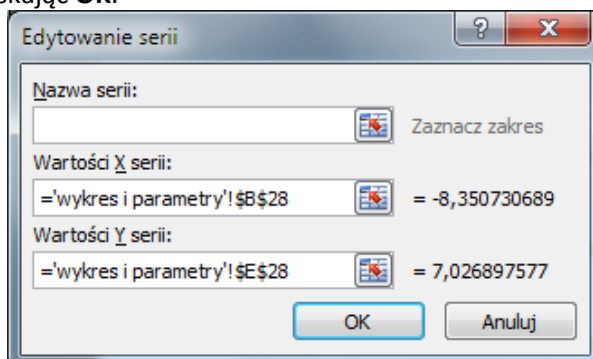
Rysunek 47. Definiowanie pierwszej serii danych

59. W oknie *Wybieranie źródła danych* jest zdefiniowana pierwsza seria, czyli współrzędne położenia cząstki przy zadanych parametrach. Dodamy jeszcze drugą serię danych, która będzie się składała z zaledwie jednego punktu. Będzie ona pokazywała bieżące położenie cząstki. W oknie *Wybieranie źródła danych* kliknij na przycisku *Dodaj* (rys. 48).



Rysunek 48. Okno *Wybieranie źródła danych* ze zdefiniowaną pierwszą serią

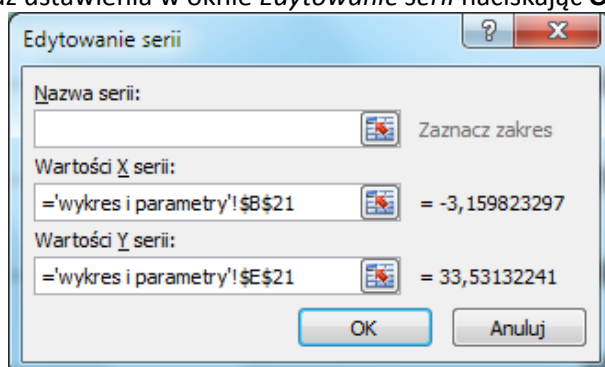
60. W oknie *Edytowanie serii* (rys. 49) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii* i zaznacz komórkę B28. Następnie przestaw kursor do pola *Wartości Y serii* i zaznacz komórkę E28. Punkt ten będzie „cząstką” poruszającą się po torze spiralnym w polu magnetycznym Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



Rysunek 49. Definiowanie drugiej serii danych

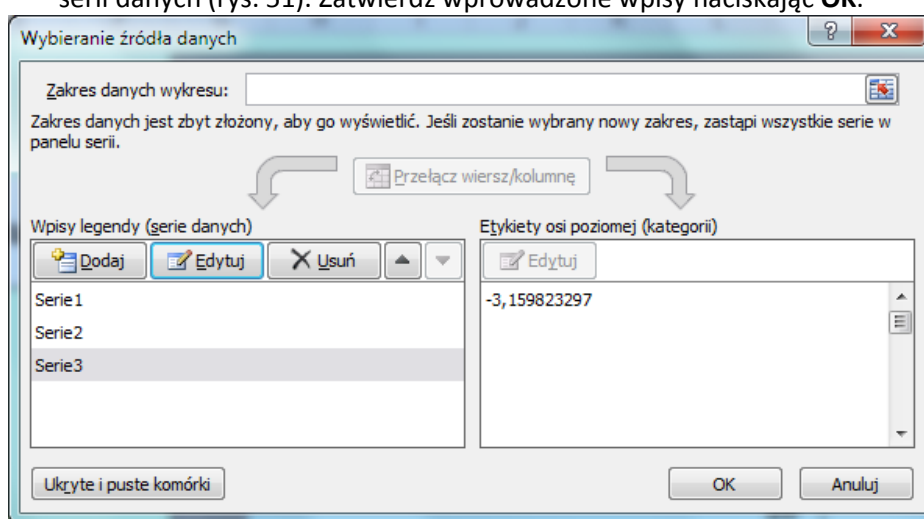
61. W oknie *Wybieranie źródła danych* kliknij ponownie na przycisku *Dodaj*. W oknie *Edytowanie serii* (rys. 50) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii* i zaznacz komórkę B21. Następnie przestaw kursor

Y serii i zaznacz komórkę E21. Punkt ten pokaże „maksymalne” położenie cząstki. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



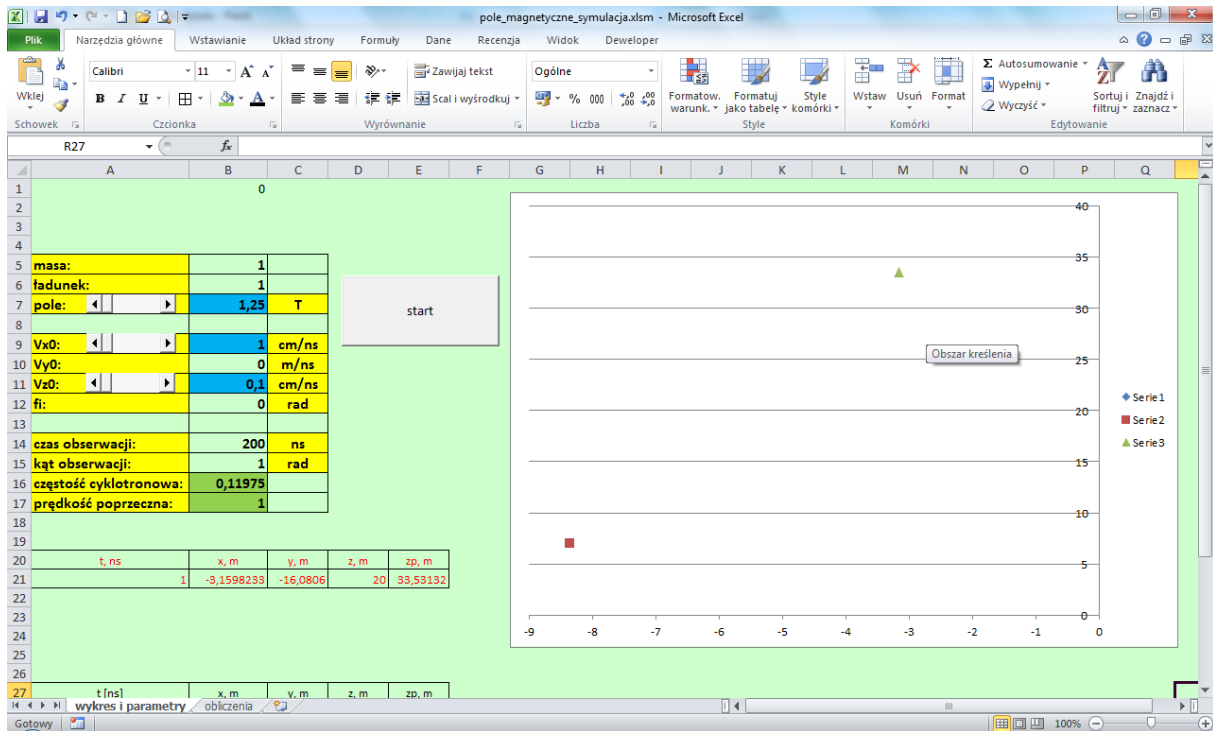
Rysunek 50. Definiowanie trzeciej serii danych

62. Spójrz na okno *Wybieranie źródła danych*. Pojawiły się w nim wpisy wszystkich trzech serii danych (rys. 51). Zatwierdź wprowadzone wpisy naciskając **OK**.



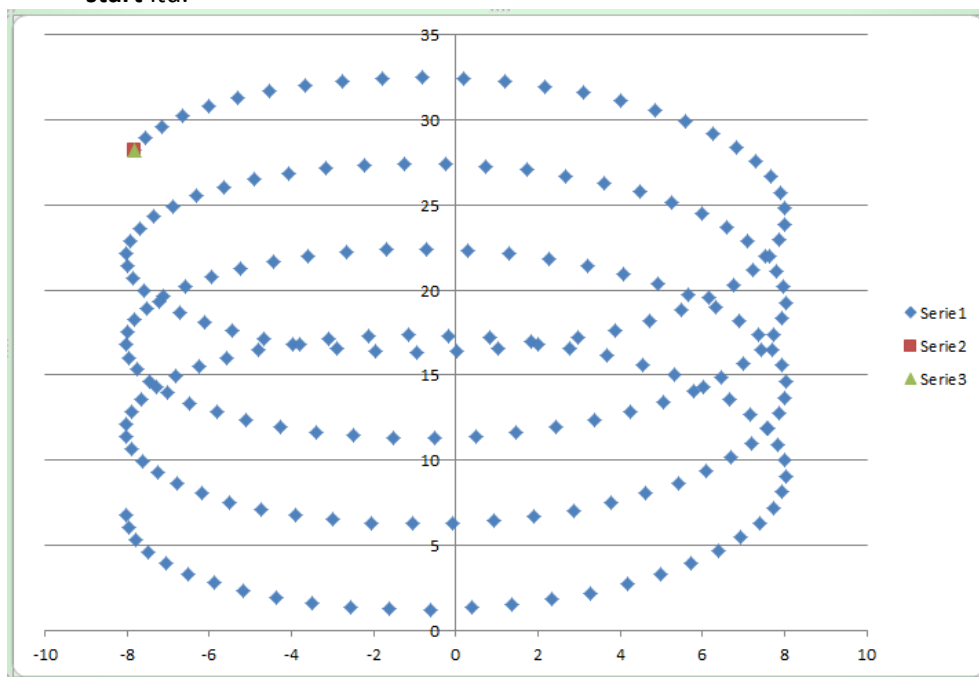
Rysunek 51. Wpisy dwóch serii danych w oknie *Wybieranie serii danych*

63. Po zatwierdzeniu wpisów w oknie *Wybieranie serii danych* w arkuszu powinien pojawić się wykres składający się z dwóch serii danych, jak na rys. 52.



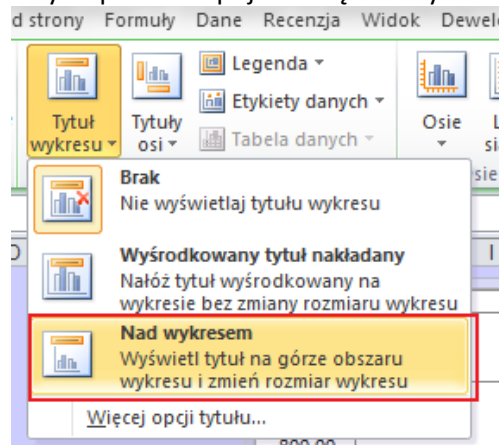
Rysunek 52. Wygląd arkusza wykres i parametry z wykresem

64. Wygląd wykresu w dużym stopniu będzie zależał od tego, co znajduje się w arkuszu *obliczenia* w komórkach A1:B101. Jeśli są w nich zera, na wykresie będą zaledwie 2 punkty. Jeśli uruchomisz teraz przyciskiem **start** symulację ruchu cząstki, do komórek A1:B101 w arkuszu *obliczenia* zostaną podstawione niezerowe wartości i wykres będzie wyglądał jak na rysunku 53. Jeśli wprowadzisz zmianę którejkolwiek wartości w komórkach z niebieskim tłem, komórki w arkuszu *obliczenia* zostaną wyzerowane (za sprawą makra *Worksheet_change*) i wykres ponownie będzie wyglądał jak na rysunku 52, dopóki ponownie nie uruchomisz symulacji przyciskiem **start** itd.



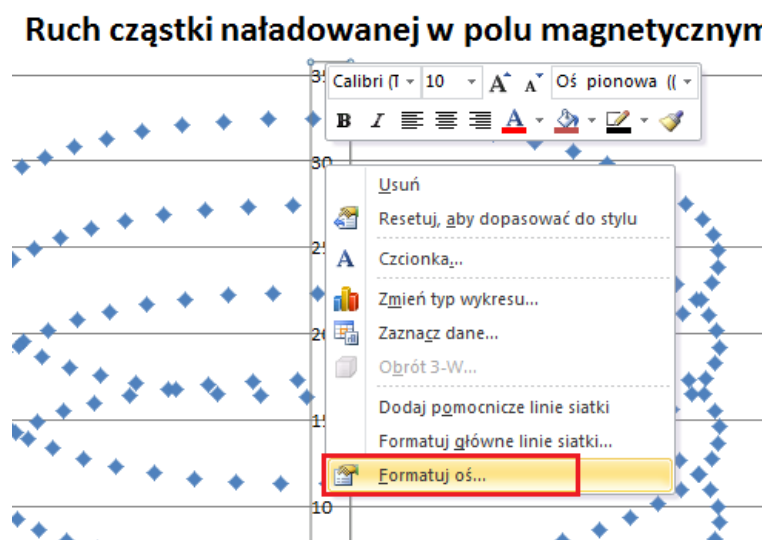
Rysunek 53. Wykres przedstawiający symulację ruchu cząstki w polu magnetycznym

65. Konieczne będzie sformatowanie wykresu, bowiem jego obecny wygląd znacznie odbiega od wzorca, do którego dążymy. Pierwsze, co możemy zrobić, to powiększenie obszaru wykresu tak, aby znalazł się w zakresie wierszy 1:25 i kolumn G:Q. Operując kursorem myszki w rogach wykresu uzyskaj znacznik do zmiany rozmiaru i powiększ wykres, jeśli trzeba przesun go w odpowiednie miejsce.
66. Przejdź na kartę *Układ* (znajdującą się w grupie *Narzędzia wykresów* na końcu wstążki; grupa narzędzi wykresów będzie widoczna pod warunkiem, że będzie aktywny wykres) i w sekcji *Etykiety* wybierz ikonę *Legenda*, a następnie, z wewnętrznej listy, *Brak*. W ten sposób wyłączymy wyświetlanie legendy na wykresie. Nie jest nam ona potrzebna, bowiem nie niesie żadnej istotnej informacji.
67. Na karcie *Układ* w sekcji *Etykiety* wybierz *Tytuł* wykresu, a następnie *Nad wykresem*, jak na rysunku 54. Wpisz tekst „Ruch cząstki naładowanej w polu magnetycznym” i naciśnij **Enter**. Tytuł powinien pojawić się nad wykresem.



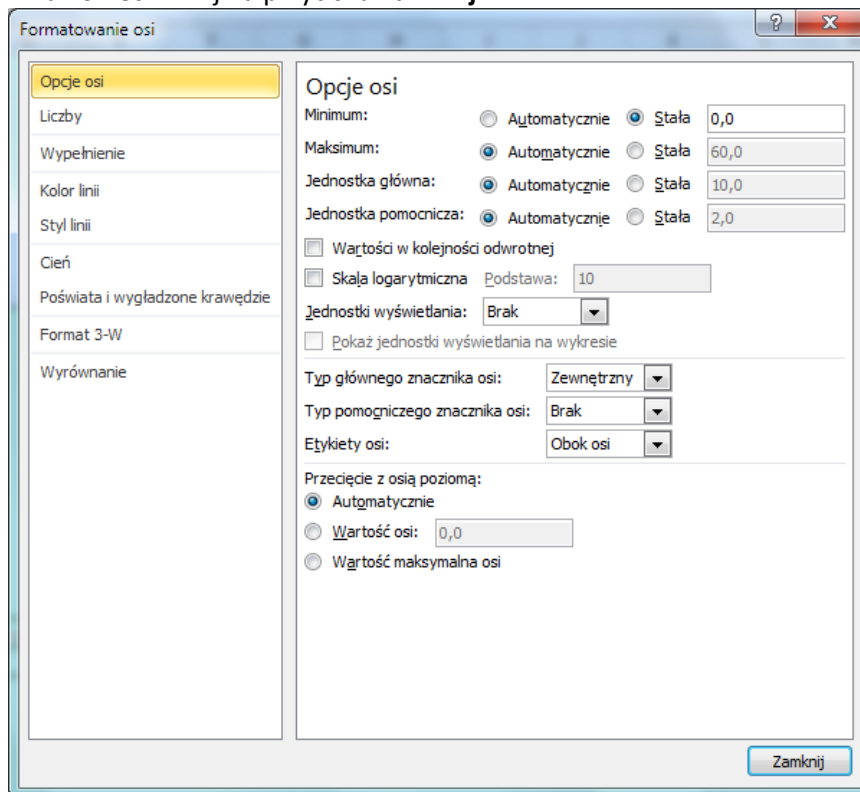
Rysunek 54. Dodawanie tytułu wykresu

68. Kliknij na jednej z wartości osi pionowej (powinna zaznaczyć się cała oś). Przejdź na kartę *Narzędzia główne* i w grupie *Czcionka* włącz pogrubienie (ikona z literą **B**). Możesz również zmienić rozmiar czcionki na większy (np. 12), żeby opisy osi były bardziej czytelne.
69. Kliknij jeszcze raz na jednej z wartości, ale tym razem prawym klawiszem myszki, i z menu podręcznego wybierz *Formatuj oś* (rys. 55).



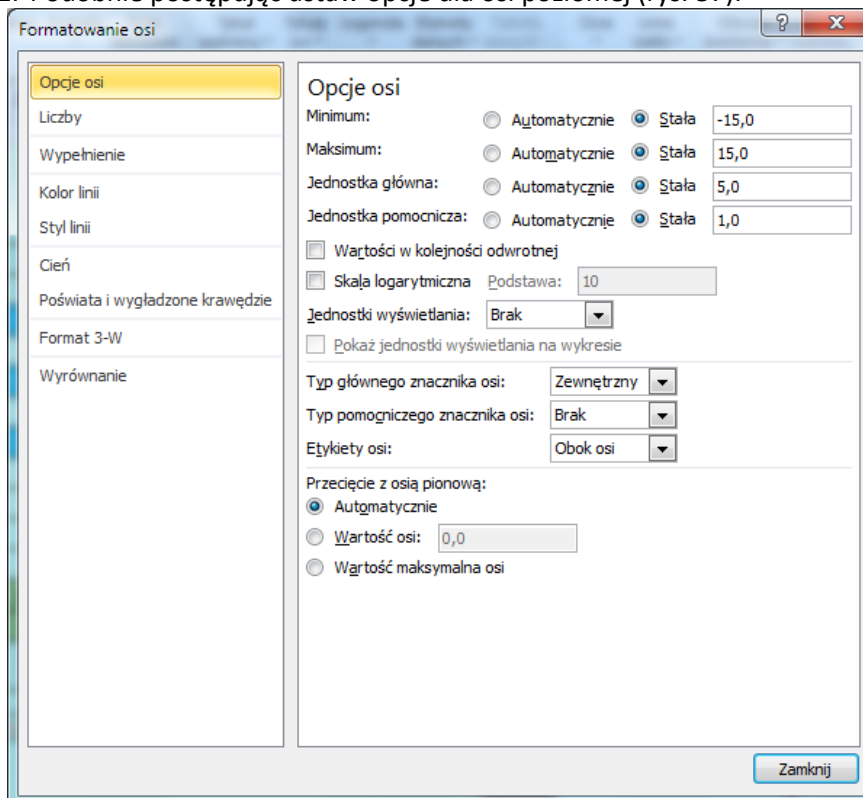
Rysunek 55. Menu podręczne osi pionowej

70. Po lewej stronie okna *Formatowanie osi* wybierz kartę *Opcje osi*. Na karcie *Opcje osi*, w prawej części okna, ustaw kategorię *Minimum*, nadając jej stałą wartość 0 (rys. 56). Na koniec kliknij na przycisku **Zamknij**.



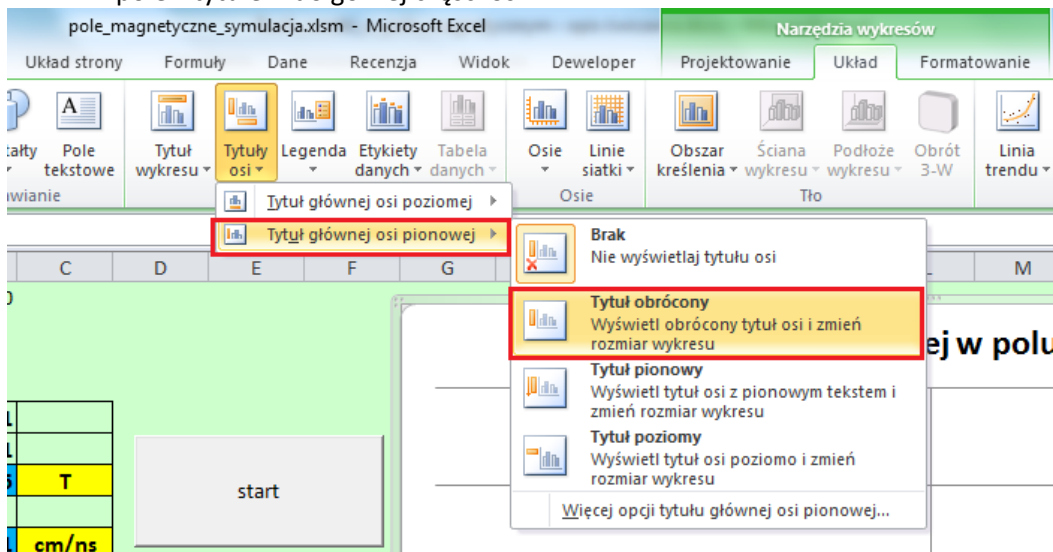
Rysunek 56. Opcje wyświetlania liczb na osi pionowej

71. Podobnie postępując ustaw opcje dla osi poziomej (rys. 57).



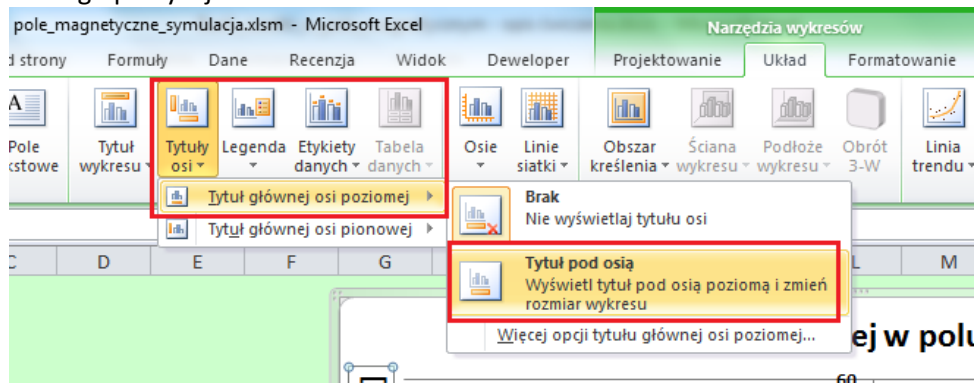
Rysunek 57. Opcje wyświetlania liczb na osi poziomej

72. Przejdź na kartę *Układ* i w sekcji *Etykiety* wybierz *Tytuły osi*, a następnie *Tytuł głównej osi pionowej* i z wewnętrznej listy *Tytuł obrócony* (rys. 58). Wpisz tekst „Z[m]” i naciśnij **Enter**. Powiększ czcionkę do rozmiaru 16 pt. Włącz pogrubienie. Przeciągnij pole z tytułem do górnej części osi.



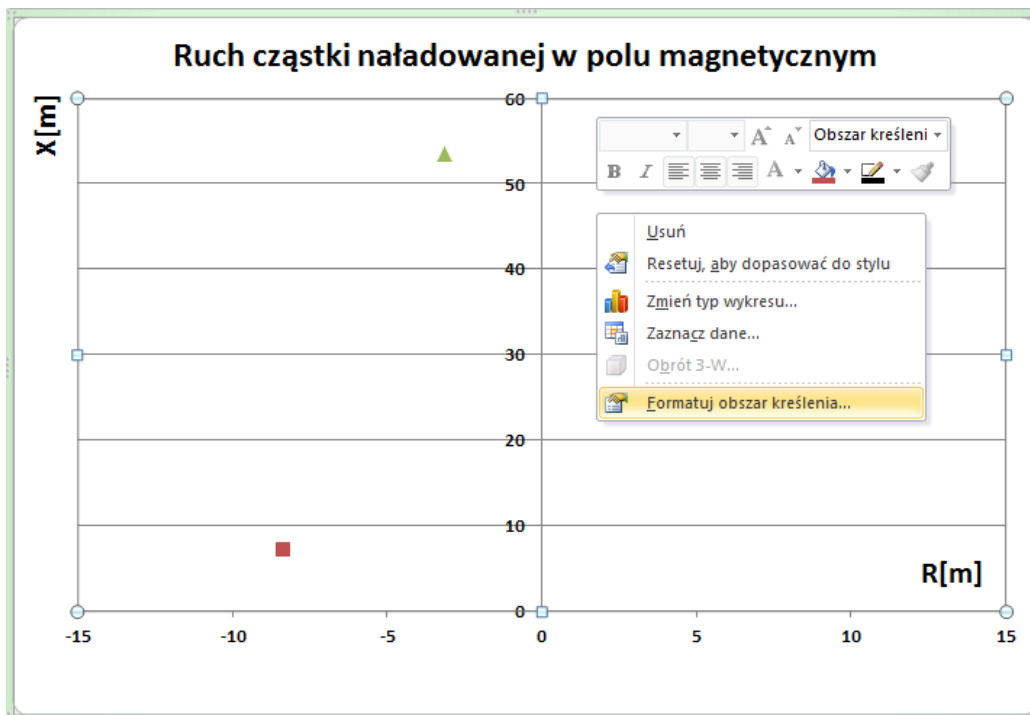
Rysunek 58. Dodanie tytułu osi pionowej

73. Ponownie wybierz polecenie *Tytuły osi*, a następnie *Tytuł głównej osi poziomej* i – z wewnętrznej listy – *Tytuł pod osią* (rys. 59). Wpisz tekst „R[m]” i naciśnij **Enter**. Sformatuj tak samo, jak tytuł osi pionowej. Przeciągnij tytuł w prawą stronę. Umieść go powyżej osi.



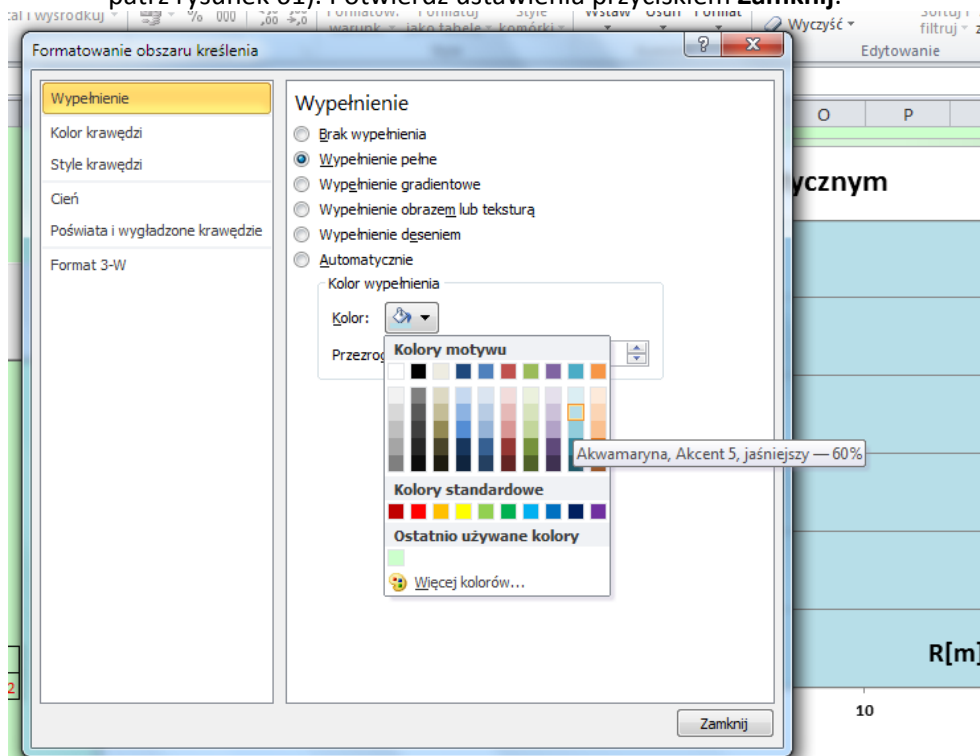
Rysunek 59. Dodanie tytułu osi poziomej

74. Kliknij prawym klawiszem myszki gdzieś w tle wykresu i z menu podręcznego wybierz *Formatuj obszar kreślenia* (rys. 60).



Rysunek 60. Wybór polecenia do sformatowania tła wykresu

75. W oknie *Formatowanie obszaru kreślenia*, na karcie *Wypełnienie* wybierz opcję *Wypełnienie pełne*, a następnie jakiś jasnoniebieski kolor (np. Akwamaryna, Akcent 5 – patrz rysunek 61). Potwierdź ustawienia przyciskiem **Zamknij**.

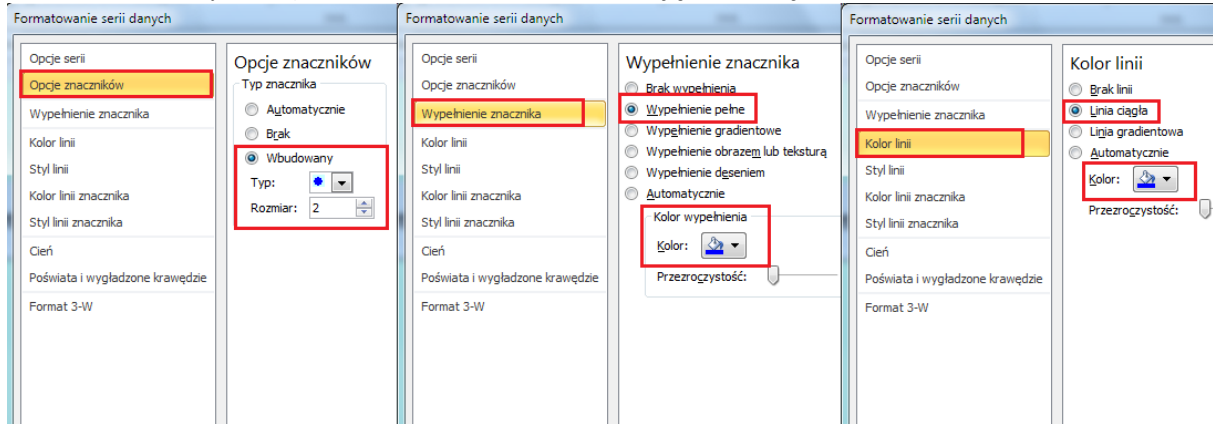


Rysunek 61. Wybór koloru do wypełnienia *Obszaru kreślenia*

76. Kliknij prawym klawiszem myszki na jednym ze znaczników pierwszej serii danych (punkty z arkusza *obliczenia*) i z menu podręcznego wybierz polecenie *Formatuj serię danych*.

Uwaga: Jeśli seria jest niewidoczna, uruchom symulację!

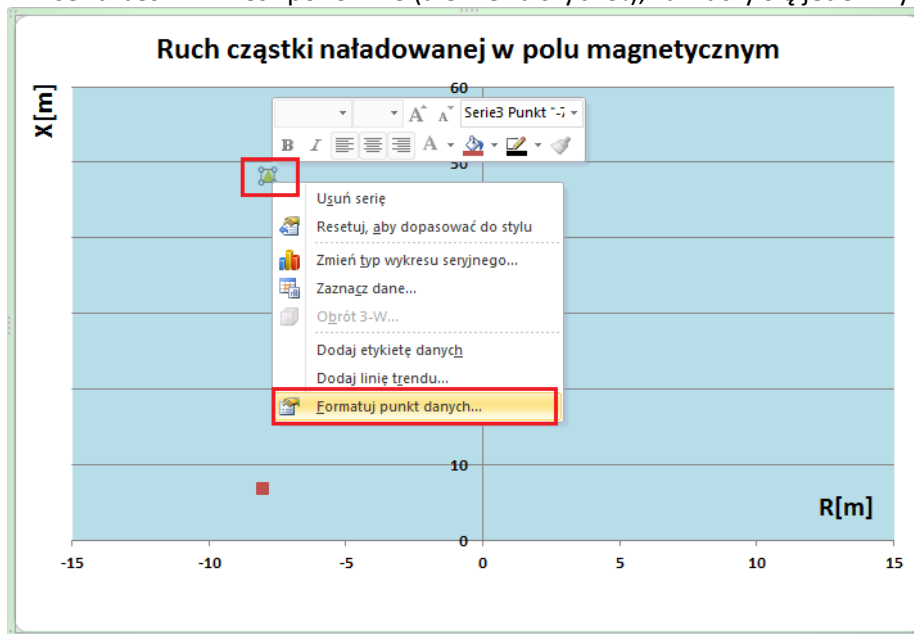
77. W oknie *Formatuj serię danych* wprowadź ustawienia jak na rysunku 62: karta *Kolor linii* wybierz: Linia ciągła, kolor – ciemny granat, karta *Opcje znaczników*: typ wbudowany, okrągły, o rozmiarze 2. Na karcie *Wypełnienie znacznika* wybierz odpowiedni kolor – ciemny niebieski, taki jak kolor linii. Przejdź na kartę *Kolor linii znacznika* i wybierz taki sam kolor, jak kolor linii i wypełnienia (nie pokazane na rysunku). Zatwierdź ustawienia naciskając **Zamknij**.



Rysunek 62. Formatowanie znaczników pierwszej serii danych

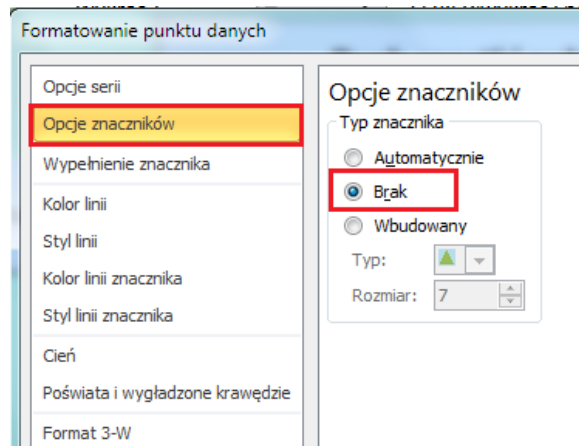
78. „Ukryjemy” teraz punkt trzeciej serii danych, który nie przedstawia żadnych istotnych treści fizycznych, a służy nam jedynie do skalowania wykresu (chodzi tutaj o punkt pokazujący maksymalne położenie cząstki). W tym celu kliknij prawym klawiszem na punkcie i z menu podręcznego wybierz polecenie *Formatuj punkt danych* (rys. 63).

Uwaga: jeśli klikniesz raz na jednym ze znaczników serii danych – zaznaczy się cała seria. Jeśli klikniesz ponownie (ale nie za szybko!), zaznaczy się jeden wybrany punkt.



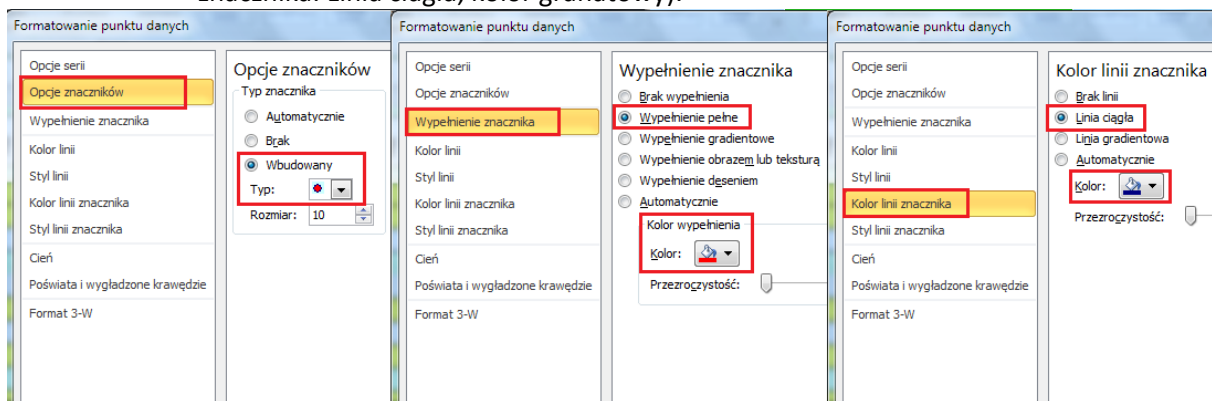
Rysunek 63. Menu podręczne punktu trzeciej serii danych

79. W oknie *Formatuj punkt danych* na karcie *Opcje znaczników* wybierz: *Brak* (rys. 64).



Rysunek 64. Formatowanie punktu drugiej serii danych

80. Został nam jeszcze jeden punkt do sformatowania – ten, który symuluje ruch cząstki w polu magnetycznym. Zaznacz ten punkt. Kliknij na nim prawym przyciskiem myszki i z menu podręcznego wybierz *Formatuj punkt danych*. W oknie *Formatuj punkt danych* wprowadź ustawienia jak na rysunku 65 (opcje znaczników: wbudowany, okrągły, rozmiar 10; wypełnienie znacznika: pełne, kolor czerwony; kolor linii znacznika: Linia ciągła, kolor granatowy).



Rysunek 65. Opcje punktu drugiej serii symulującego ruch ciała

----- KONIEC -----

Dodatek – podstawowe informacje o VBA

Makropolecenia (makra) składają się z serii poleceń języka programowania zintegrowanego z Excelem. Językiem tym jest **Visual Basic for Application**, skrótowo nazywany **VBA**. Jest to obiektowy język programowania umożliwiający zarówno korzystanie z właściwości arkusza, jak również wywoływanie funkcji systemowych Windows.

Makro jest programem, który wykonuje określone zadania. W Excelu spotykamy się z dwoma rodzajami aplikacji użytkownika:

- ❑ procedury, których użycie pozwala na zautomatyzowanie jednej lub wielu następujących po sobie czynności. Rozpoczynają się od słowa kluczowego *Sub*, po którym następuje nazwa procedury nadana przez użytkownika, a kończą słowami kluczowymi *End Sub*.
- ❑ funkcje, czyli makra wykonujące obliczenia w komórce arkusza. Wynik działania funkcji użytkownika – podobnie jak w przypadku korzystania z funkcji wbudowanych – jest umieszczany w komórce arkusza, w której funkcja została wywołana. Łatwo je odróżnić od procedur typu *Sub*, gdyż rozpoczynają się od słowa kluczowego *Function*, po którym następuje nazwa funkcji nadana przez użytkownika, a kończą słowami kluczowymi *End Function* – nie będziemy z nich korzystać w ćwiczeniu – symulacji rzutu pionowego.

Edytor VBA wywołujemy naciskając kombinację klawiszy **Alt+F11**. Jest on wyposażony w polecenia i paski narzędzi ułatwiające tworzenie, analizę i uruchamianie makr. Gotowe makra można również uruchamiać na przykład: poleceniem ze wstążki **WIDOK→MAKRA→WYŚWIETL MAKRA**, klikając myszą na specjalnie utworzonym przycisku lub naciskając ustaloną kombinację klawiszy (tzw. skrót klawiaturowy).

Kod źródłowy makra ma postać tekstu, który można również edytować zwyczajnym edytorem. Aby podejrzeć tekst źródłowy makropoleceń zapisanych w arkuszu należy otworzyć to makro w edytorze VBA. W tym celu można kliknąć polecenie **WIDOK→MAKRA→WYŚWIETL MAKRA**, wybrać z listy odpowiednie makropolecenie i nacisnąć przycisk **Edycja**. Można również otworzyć edytor VBA kombinacją klawiszy **Alt+F11**, odnaleźć odpowiednie makro (może one być przypisane do konkretnego arkusza, całego skoroszytu, lub znajdować się w specjalnym „kontenerze” zwanym *Module*, który jest przypisany do całego skoroszytu. Lokalizacja makra w dużym stopniu zależy od tego, co ma ono za zadanie wykonać. Do konkretnych arkuszy są często przypisywane tzw. procedury zdarzeniowe. Bardziej ogólnie działające procedury, często operujące na danych z wielu arkuszy, wstawiane są właśnie w dodatkowych „modułach”. Podobnie jest z funkcjami tworzonymi przez użytkownika. Informacja w edytorze VBA, w przeciwieństwie do normalnego arkusza Excela, nie jest podzielona na komórki, lecz przedstawiona w formie zwartej ciągu tekstu.

Typy danych

Zadaniem procedur VBA jest manipulowanie różnego rodzaju danymi. Typ danych określa, w jaki sposób są one przechowywane w pamięci komputera. Informuje też użytkownika, z jakiego typu danymi ma do czynienia. Dla przykładu, dane mogą być przechowywane jako liczby (całkowite lub rzeczywiste), łańcuchy znaków (teksty), wartości logiczne (*True* (Prawda) lub *False* (Fałsz)), daty itd.

Jeśli, pisząc procedury, nie określimy typu danych, to Visual Basic automatycznie zastosuje typ *Variant*. *Variant* sam sprawdza, z jaką wartością ma do czynienia i „dopasowuje się” do niej. Może on zastępować dowolny typ danych, ale używanie go powoduje marnowanie czasu i pamięci komputera, szczególnie w przypadku tworzenia długich i skomplikowanych procedur. Przypisanie określonych typów danych do zmiennych używanych przez nas w kodzie przyspiesza wykonywanie programu i oszczędza pamięć, ponieważ Excel rezerwuje tylko tyle pamięci, ile jest potrzebne. Poniżej przedstawiamy zestawienie przykładowych typów danych (podamy je informacyjnie):

- ❑ *String* łańcuch tekstowy o długości od 0 do 65 535 znaków;

- ❑ *Byte* liczba całkowita z przedziału od 0 do 255
- ❑ *Integer* liczba całkowita z przedziału od -32 768 do 32 767;
- ❑ *Long* liczba całkowita długa z zakresu od -2 147 483 648 do 2 147 483 648;
- ❑ *Currency* liczba całkowita z czterema miejscami po przecinku od -922.337.203.685.477,5808 do 922.337.203.685.477,5807;
- ❑ *Date* data w przedziale od 1/1/100 do 31/12/9999;
- ❑ *Single* liczba rzeczywista (może mieć część ułamkową) od -3,402823E38 do 1,401298E45;
- ❑ *Double* liczba rzeczywista dwukrotnie większa od liczby *Single* z zakresu -1,79769313486232E308 do -4,94065645841247E-324 dla liczb ujemnych i od 4,94065645841247E-324 do 1,79769313486232E308 dla liczb dodatnich;
- ❑ *Variant* dane dowolnego typu;
- ❑ i wiele innych typów, również definiowanych przez użytkownika.

Zmienne i ich deklaracje

Deklarowanie zmiennych jest dobrym zwyczajem, ponieważ ułatwia analizę kodu programu, pomaga też łatwiej wychwycić niektóre błędy. Zmienna jest wyrazem, skrótem, który reprezentuje wynik działania jakiejś instrukcji. Jak sama nazwa wskazuje, wartość zmiennej może ulegać zmianie w czasie wykonywania programu.

Nazwy nadawane zmiennym nie mogą być zupełnie dowolne. Podlegają one pewnym ścisłym regułom. Mogą składać się z liter, cyfr i niektórych znaków przestankowych z wyjątkiem znaków: . # \$ % & @ ! Nazwa zmiennej nie może się zaczynać od cyfry i nie może zawierać spacji! Visual Basic zawiera też listę słów w języku angielskim, które nie mogą być używane jako nazwa zmiennej. Są to tzw. słowa kluczowe języka, takie jak np. *Abs*, *Date*, *False*, *Select*, *Static*, i wiele innych. Visual Basic nie rozróżnia małych i dużych liter w nazwach zmiennych.

Jak już wcześniej wspomnieliśmy, każda zmienna posiada określony typ. Deklarowanie nazw zmiennych i ich typów przed pierwszym użyciem w procedurze daje dwie korzyści:

- ❑ przyspiesza wykonanie procedury i oszczędza pamięć,
- ❑ zapobiega błędom wynikającym z nieprawidłowego wpisania nazwy zmiennej.

Deklaracja zmiennej polega na wpisaniu instrukcji *Dim* (skrót od Dimension), po której następuje nazwa zmiennej oraz jej typ. Przyjęta składnia deklaracji zmiennych jest następująca:

Dim zmienna *As* typ

gdzie **zmienna** stanowi nazwę zmiennej, a **typ** określa typ danych, które będą przechowywane w zmiennej. Na przykład polecenie:

Dim Nazwisko *As* String

deklaruje zmienną *Nazwisko* jako zmienną typu *String* (tekst, łańcuch znaków). Natomiast deklaracja:

Dim Nazwisko

deklaruje zmienną *Nazwisko* jako zmienną typu *Variant*.

Instrukcja warunkowa *If ... Then*

If ... Then jest najprostszą strukturą do podejmowania decyzji w Visual Basicu. Umożliwia ona wykonanie jednej lub więcej instrukcji w zależności od podanego warunku. **Warunek** może być liczbą lub tekstem, który przekazuje wartość logiczną *True* lub *False*. **Instrukcje** – jedna lub więcej dowolnych instrukcji Visual Basicu, które procedura ma wykonać kiedy **warunek** okaże się prawdą. Struktura *If Then* może być zapisana następujący sposób:

```
If warunek Then
    [instrukcje]
End If
```

Słowa kluczowe *If* i *End if* są tu jakby klamrą spinającą cały blok instrukcji. Dokładnie wiadomo kiedy blok się zaczyna i gdzie się kończy.

W ogólności zdarza się, że potrzebne są jedne **instrukcje**, jeśli **warunek** jest prawdziwy oraz inne, jeżeli **warunek** jest fałszywy. Wówczas pomocna będzie struktura określana jako *If ... Then ... Else* (podajemy ją tylko informacyjnie, nie stosujemy w niniejszym opracowaniu):

```
If warunek Then
    [instrukcje, gdy warunek jest prawdą]
Else
    [instrukcje, gdy warunek jest fałszywy]
End If
```

Powyższa struktura blokowa wykonuje **instrukcję** lub blok instrukcji, gdy **warunek** jest prawdziwy i jedną **instrukcję** lub blok instrukcji, gdy **warunek** jest fałszywy. Jeżeli warunek jest prawdziwy, to Visual Basic wykona instrukcje zawarte pomiędzy *Then* i *Else*, natomiast opuści instrukcje znajdujące się pomiędzy *Else* i *End If*, a potem wykona dalsze instrukcje procedury. Jeżeli warunek jest fałszywy, to Visual Basic opuści instrukcje znajdujące się pomiędzy *Then* i *Else*, wykona instrukcje zawarte między *Else* i *End If*, a następnie wykona dalsze instrukcje procedury.

Przykład zastosowania znajduje się na końcu.

Pętla liczona *For ... Next*

Pętla *For ... Next* wykona dany blok **instrukcji** z góry określoną ilość razy. Oto jak wygląda składnia pętli:

```
For licznik = start To stop (Step krok)
    [instrukcje]
Next licznik
```

Licznik jest dowolną nazwą zmiennej używanej do przechowywania liczby powtórzeń. **Start** określa początkową wartość licznika, **stop** – maksymalną liczbę powtórzeń. Słowo kluczowe *Next* odsyła ponownie do pierwszej linijki pętli i sprawdzenia wartości licznika. Jeżeli liczba w liczniku okaże się większa niż wartość maksymalna (*stop*), to nastąpi wyjście z pętli. **Krok** określa o ile ma się zmienić wartość licznika przy przejściu do następnego powtórzenia pętli. Użycie wyrażenia podanego w nawiasie, tzn. *Step krok*, jest opcjonalne. Jeśli go pominiemy, za każdym razem, gdy Visual Basic wykona instrukcje w pętli, wartość licznika będzie zwiększana o jeden.

Przykład zastosowania pętli znajduje się na końcu.

Model obiektowy. Odwołania do obiektów

Excel posiada ponad 100 **obiektów**. Przykładowe obiekty to:

- ❑ skoroszyt (*Workbook*),
- ❑ arkusz (*Worksheet*),

- ❑ zakres komórek w arkuszu (*Range*),
- ❑ komórka (*Cells*), alternatywa dla obiektu *Range*, w niektórych przypadkach łatwiejsza w użyciu,

i wiele, wiele innych.

Obiekty są zorganizowane w pewną hierarchię. Niektóre obiekty zawierają w sobie inne obiekty. Excel jest obiektem zwanym aplikacją (*Application*) i jest on obiektem nadrzędnym w stosunku do innych. Obiekt *Application* zawiera w sobie obiekty, takie jak skoroszyty (*Workbooks*). Skoroszyt zaś zawiera inne obiekty, takie jak arkusze (*Worksheets*) czy wykresy (*Charts*).

Obiekt *Range* może oznaczać pojedynczą komórkę lub zakres komórek arkusza. Jak wyżej wspomnieliśmy, znajduje się on wewnątrz większego obiektu, jakim jest *Worksheet*.

Range pozwala na odwołanie do komórki arkusza lub większego zakresu komórek za pomocą adresu (np. A1, B5, A2:A10, itd.). Jeżeli chcemy odwołać się do komórki B5 pierwszego arkusza skoroszytu o nazwie *ćwiczenia* (za pomocą obiektu *Range*), żeby, na przykład, ją zaznaczyć, musimy w procedurze VBA wpisać następującą instrukcję:

```
Workbooks("ćwiczenia").Worksheets(1).Range("B5").Select
```

Jak widzisz – poszczególne elementy w hierarchii są rozdzielone kropką. Aby wybrać zakres komórek (np. od C6 do C10) wpisz:

```
Range("C6:C10").Select
```

Właściwość *Cells* wymaga zazwyczaj podania dwóch argumentów. Argument pierwszy wskazuje numer wiersza, natomiast drugi, to numer kolumny. W tej konwencji odwołanie do komórki B5, która znajduje się na przecięciu drugiej kolumny i piątego wiersza, będzie wyglądało następująco:

```
ThisWorkbook.Worksheets(1).Cells(5,2)
```

W powyższej instrukcji obiektem nadrzędnym jest *ThisWorkbook*. Oznacza dosłownie „ten skoroszyt”, ten – w którym piszemy makro. Taki sposób odwołania do skoroszytu jest alternatywą dla zapisu *Workbooks("nazwa")*. W przypadku, kiedy makro operuje na wielu skoroszytach, łatwiej jest odróżnić w kodzie programu te instrukcje, które mają się odwoływać do skoroszytu, w którym jest makro.

PRZYKŁADY:

Przykład zastosowania instrukcji warunkowej *If*: założmy, że w komórce A1 jest jakaś liczba, którą chcemy podzielić przez jakąś inną liczbę znajdującą się w komórce A2. Wynik dzielenia chcemy uzyskać w komórce B1.

Zwróć uwagę, że gdyby w komórce A2 użytkownik wpisał zero, wówczas w wyniku wykonania dzielenia w komórce wynikowej – B1 – pojawi się błąd!

Chcemy aby nasza procedura działała następująco: jeżeli liczba w komórce A2 jest różna od zera, wówczas wykonaj dzielenie i wynik wpisz do B1. W przeciwnym wypadku (oznacza to, że w komórce A2 jest wpisane zero), w komórce B1 wyświetli się komunikat w postaci tekstu „Błąd! Nie dziel przez zero!”.

```
If Range(„A2”) <> 0 Then  
    Range(„B1”) = Range(„A1”) / Range(„A2”)  
Else  
    Range(„B1”) = „Błąd! Nie dziel przez zero!”  
End If
```

Brak w powyższym przykładzie odwołania do nadrzędnych obiektów w stosunku do *Range* w postaci arkusza czy skoroszytu (na przykład: *ThisWorkbook.Worksheets(1).Range(„B1”)*) oznacza, że procedura ma się wykonać w aktywnym skoroszytcie, w aktywnym arkuszu.

Podamy jeszcze, jak należałoby zapisać tę samą instrukcję używając obiektu *Cells*:

```

If Cells(2,1) <> 0 Then
    Cells(1,2) = Cells(1,1) / Cells(2,1)
Else
    Cells(1,2) = "Błąd! Nie dziel przez zero!"
End If

```

Przykład zastosowania pętli For Next: chcemy wypełnić co drugą komórkę z zakresu od A2 do A10 wartościami z których każda jest 100 razy większa niż numer wiersza komórki, do której trafia. Wartość **stop** będzie równa 10 – jest to maksymalna wartość **licznika**. **Licznik** to zmienna reprezentująca obroty pętli – najczęściej nazywamy ją „i”. **Step**, czyli krok, informuje nas o ile różni się każda następną wartość licznika od poprzedniej. **Start** to wartość początkowa **licznika**, u nas wynosi 2.

```

For i = 2 To 10 Step 2
    Cells(i, 1) = i * 100
Next i

```

Po wejściu do pętli zmienna *i* przyjmuje wartość 2. Jeżeli jest ona mniejsza (lub równa) 10 – więc pętla się wykonuje: do komórki o numerze kolumny „1” (czyli A) i numerze wiersza równym bieżącej wartości *i*, czyli 2, podstawia się wynik działania $i*100$, czyli przy pierwszym obrocie naszej pętli $2*100=200$.

Licznik zwiększa się – *i* staje się większe o 2.

W drugim obrocie pętli *i* równa się więc 4. Następuje sprawdzenie czy 4 jest mniejsze lub równe 10, jeśli tak, do kolejnej komórki (o numerze kolumny 1 i numerze wiersza 4 – takim jak *i*) wstawia wartość $i*100=4*100=400$.

Itd. aż bieżąca wartość **licznika** przekroczy wartość **stop**. Wtedy pętla kończy działanie.

Wynik ćwiczenia:

	A	B	C	D
1				
2	200			
3				
4	400			
5				
6	600			
7				
8	800			
9				
10	1000			
11				