



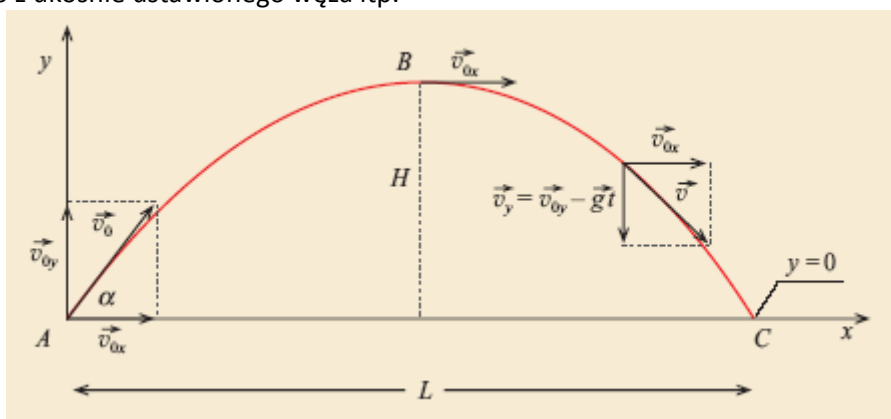
Rzut ukośny – ćwiczenie w Excelu

Wstęp

Wiemy, że każde ciało, które spada swobodnie z wysokości h , porusza się z przyspieszeniem równym przyspieszeniu ziemskiemu ($a = g$). Przyspieszenie ziemskie jest skierowane pionowo w dół, do środka Ziemi. Jego wartość jest różna w różnych miejscach naszej planety ze względu na niejednorodności w budowie skorupy ziemskiej. Do celów obliczeniowych przyjmuje się najczęściej wartość $g = 9,81\text{m/s}^2$.

Powiedzieliśmy sobie, że wszystkie ciała spadają z jednakowym **przyspieszeniem**, ale jednak z codziennych obserwacji wynika, że nie wszystkie spadają z jednakową **prędkością**. Na przykład, człowiek ze spadochronem spada wolniej niż bez spadochronu, piórko spada wolniej niż kamień (łatwo można to sprawdzić upuszczając oba przedmioty jednocześnie z tej samej wysokości). Dlaczego tak się dzieje, skoro Ziemia przyciąga wszystkie przedmioty jednakowo? Odpowiedzią jest **siła oporu powietrza**, która przeciwdziała sile grawitacji. My w naszych rozważaniach dotyczących ruchu ciał w powietrzu przyjmujemy obraz nieco uproszczony, gdyż **nie będziemy uwzględniać sił oporu powietrza**. W rzeczywistości odpowiadałoby to sytuacji, w której rozważalibyśmy ruch ciał znajdujących się w doskonałej próżni poruszających się w polu grawitacyjnym naszej planety.

Kiedy rzucimy jakieś ciało pod kątem α do poziomu z prędkością początkową \vec{v}_0 (spójrz na rys. 1) mówimy, że mamy do czynienia z **rzutem ukośnym**. Przykładem rzutu ukośnego może być balistyczny ruch pocisku wystrzelonego z wyrzutni, rzut piłką do kosza, ruch strumienia wody wylatującego z ukośnie ustawionego węża itp.



Rysunek 1. Rzut ukośny. Torem ciała jest parabola. Prędkość w kierunku poziomym jest stała przez cały czas. W kierunku pionowym prędkość zmienia się według wzoru $v_y = v_{0y} - gt$

Ruch ciała w rzucie ukośnym możemy w układzie współrzędnych kartezjańskich (x, y) rozłożyć na dwa niezależne, wzajemnie prostopadłe ruchy składowe: **poziomy** i **pionowy** (przypomnij sobie sumowanie wektorów).

Prędkość początkową, z jaką wyrzuciliśmy ciało, \vec{v}_0 , rozłożymy na dwie składowe: \vec{v}_{0x} i \vec{v}_{0y} , jak na rysunku 1. Składowe te są jednocześnie rzutami wektora prędkości \vec{v}_0 na osie układu współrzędnych. Znając wartość kąta α i korzystając ze znanych funkcji trygonometrycznych, wartości prędkości składowych wyrazimy za pomocą wzorów:

$$\left. \begin{aligned} v_{0x} &= v_0 \cos \alpha \\ v_{0y} &= v_0 \sin \alpha \end{aligned} \right\} \quad (1)$$



Ruch ciała w kierunku **poziomym** jest ruchem jednostajnym ze stałą prędkością \vec{v}_{0x} (w tym kierunku nie występuje żadne przyśpieszenie), więc x – położenie ciała w danej chwili t opiszemy wyrażeniem analogicznym do wzoru na drogę w ruchu jednostajnym ($s = vt$):

$$x(t) = v_{0x}t \quad (2)$$

W przypadku **ruchu** ciała w kierunku **pionowym** występuje stałe przyśpieszenie ziemskie \vec{g} zwrócone w dół w kierunku ujemnym osi y oraz prędkość początkowa \vec{v}_{0y} zwrócona w kierunku dodatnim osi y . Mamy tu do czynienia z ruchem takim, jakbyśmy rzucili ciało pionowo w górę. Takie ciało, wznosząc się, stale wytracałoby prędkość, aż do chwilowego zatrzymania się – wtedy prędkość chwilowa byłaby równa zeru, a ciało osiągnęłoby maksymalną wysokość – H . Mielibyśmy tu do czynienia z ruchem jednostajnie opóźnionym ze stałym opóźnieniem skierowanym przeciwnie do kierunku ruchu, równym co do wartości g . Po chwilowym zatrzymaniu ciała rozpoczęłoby się druga faza ruchu – zaczęłoby ono spadać swobodnie w dół (z przyspieszeniem g).

Zatem położenie ciała w kierunku pionowym (współrzędna y), czyli jego wysokość w dowolnej chwili t , wyrażamy za pomocą wzoru analogicznego do wyrażenia na drogę w ruchu jednostajnie opóźnionym ($s = s_0 + v_0t - \frac{at^2}{2}$):

$$y(t) = y_0 + v_{0y}t - \frac{gt^2}{2} \quad (3)$$

Najwyższą wysokość, na którą wzniesie się ciało, H_{max} , czyli **zasięg pionowy**, ciało uzyska po czasie wznoszenia t_w danym wzorem:

$$t_w = \frac{v_{0y}}{g} = \frac{v_0 \sin \alpha}{g} \quad (4)$$

H_{max} możemy otrzymać ze wzoru (3) wstawiając t_w w miejsce t . Wykonując przekształcenia otrzymujemy kolejno:

$$H_{max} = y_0 + v_{0y}t_w - \frac{gt_w^2}{2} = y_0 + v_{0y} \frac{v_{0y}}{g} - \frac{g \left(\frac{v_{0y}}{g}\right)^2}{2} = y_0 + \frac{v_{0y}^2}{g} - \frac{v_{0y}^2}{2g}$$

i ostatecznie

$$H_{max} = y_0 + \frac{v_{0y}^2}{2g} \quad (5)$$

Zależność (5) opisuje maksymalną wysokość ciała rzuconego w górę z prędkością v_{0y} – jest to wzór opisujący zasięg pionowy ciała w rzucie pionowym.

W przypadku rzutu ukośnego, w którym ciało rzucono z prędkością v_0 , a v_{0y} jest jej składową w kierunku pionowym, uwzględniając, że prędkość początkowa w kierunku pionowym wynosi (1) $v_{0y} = v_0 \sin \alpha$, wzór na maksymalną wysokość ciała przyjmie postać:

$$H_{max} = y_0 + \frac{v_0^2 \sin^2 \alpha}{2g} \quad (6)$$

Ze wzoru (6) wynika, że największą wysokość ciało osiągnie, gdy $\sin \alpha = 1$. Wtedy $\alpha = 90^\circ$ i rzut ukośny w naturalny sposób przechodzi w rzut pionowy w górę.

W przypadku, kiedy ciało rzucamy z poziomu ziemi, $t_w = t_s = \frac{1}{2}t_c$, gdzie t_s – czas spadania, t_c – czas całkowity ruchu. Kiedy wyrzut następuje z innej wysokości, niż ta, na której ciało spadnie, wówczas $t_c = t_w + t_s$ ale $t_w \neq t_s$.

W przypadku ogólnym, czas spadania t_s , wyznaczymy ze wzoru:

$$H_{max} = \frac{gt_s^2}{2}$$

stąd otrzymujemy

$$t_s = \sqrt{\frac{2H_{max}}{g}} \quad (7)$$

Torem ciała w rzucie ukośnym jest **parabola** (spójrz na rysunek 1) o ramionach skierowanych w dół, która ma ogólną postać: $y = -ax^2 + bx$. To znaczy, że ciało w każdej chwili swojego ruchu, t , znajduje się w jednym z punktów paraboli.

Wzory (2) i (3) wyrażają współrzędne dowolnego punktu na krzywej toru w dowolnym czasie t . Po uwzględnieniu zależności (1) przyjmą one postać

$$\left. \begin{aligned} x(t) &= v_0 t \cos \alpha \\ y(t) &= v_0 t \sin \alpha - \frac{gt^2}{2} \end{aligned} \right\} \quad (8)$$

Z pierwszego równania (wzór 8) mamy $t = \frac{x}{v_0 \cos \alpha}$. Podstawiając to do równania drugiego, otrzymamy ostatecznie równanie toru $y(x)$ w postaci paraboli

$$y(x) = x \operatorname{tg} \alpha - \frac{g}{2v_0^2 \cos^2 \alpha} x^2 \quad (9)$$

Równanie toru (9) pozwala wyznaczyć w bardzo prosty sposób **zasięg poziomy** L rzutu ukośnego, czyli odległość w poziomie od miejsca wyrzucenia, do miejsca, w którym ciało spadnie na ziemię. W tym celu wystarczy w równaniu (9) podstawić $y = 0$, czyli

$$0 = x \operatorname{tg} \alpha - \frac{g}{2v_0^2 \cos^2 \alpha} x^2 \quad (10)$$

Równanie kwadratowe (10) ma dwa pierwiastki określające wartości przecięcia punktów paraboli z osią x (punktów A i C na rysunku 1). Pierwszy pierwiastek $x = 0$ odpowiada punktowi A . Drugi pierwiastek jest równy właśnie zasięgowi poziomemu rzutu $x = L$. Zatem z (10) mamy

$$L = x = \frac{v_0^2 2 \cos \alpha \sin \alpha}{g}$$

Wykorzystamy tutaj zależność trygonometryczną: $\sin 2\alpha = 2 \cos \alpha \sin \alpha$ i ostatecznie zasięg poziomy w rzucie ukośnym wyrazimy za pomocą wzoru

$$L = \frac{v_0^2 \sin 2\alpha}{g} \quad (11)$$

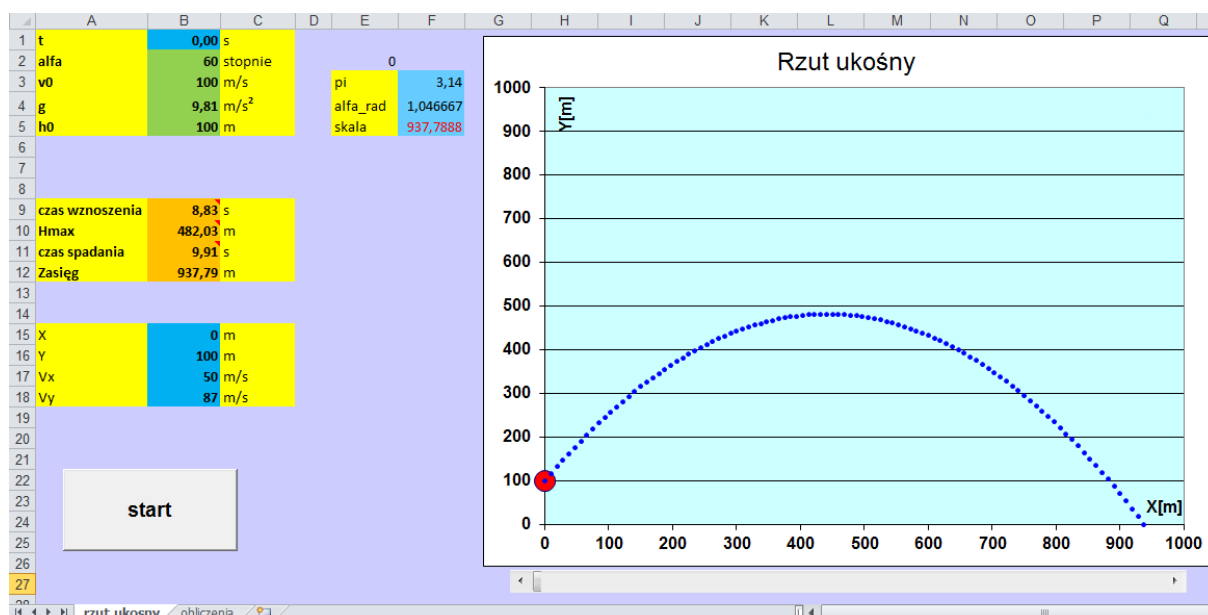
Widzimy, że zasięg rzutu L będzie największy wtedy, gdy dla określonej prędkości $\sin 2\alpha = 1$, czyli gdy $\alpha = 45^\circ$.

Jak działa model rzutu ukośnego w Excelu

W arkuszu *rzut ukosny* umieszczony został wykres przedstawiający tor rzutu ukośnego w układzie współrzędnych (x, y) (rys. 2). Czas lotu ciała został podzielony na 100 równych przedziałów. Wykres w postaci małych granatowych kropek przedstawia kolejne położenia ciała od momentu wyrzucenia do upadku (wszystkie potencjalne położenia), natomiast duża czerwona kropka pokazuje bieżące położenie ciała (dla czasu t widocznego w komórce B1).

W komórkach kolumny B sformatowanych na zielono znajdziesz wejściowe parametry rzutu do sterowania wykresem, i tak: w komórce B2 możesz wprowadzić wartość kąta (w stopniach), pod jakim wyrzucamy ciało, w komórce B3 – prędkość początkową v_0 wyrażoną w m/s, B5 – wysokość początkową h_0 (wartość równa zero odpowiada poziomowi ziemi). W komórce B4 umieściliśmy wartość przyspieszenia ziemskiego.

Dla wprowadzonych przez użytkownika parametrów do komórek B2:B5, w komórkach B9:B12 (za pomocą formuł, które opisujemy poniżej) obliczane są stałe wartości charakteryzujące ruch, takie jak czas wznoszenia i spadania, maksymalna wysokość oraz zasięg poziomy. W komórkach B15:B18 znajdują się wartości chwilowe, takie jak: bieżące położenie (współrzędne x i y) oraz składowe prędkości v_x i v_y .



Rysunek 2. Rzut ukośny – model w Excelu

Wprowadź przykładowe wartości do komórek B2:B5 (mogą być takie, jak na rysunku 2), a następnie naciśnij przycisk **start**. Przyciskiem tym uruchomisz *makro*, które przygotuje i przeprowadzi symulację rzutu ukośnego, wyznaczając bieżące wartości położenia x i y – tabela wartości znajduje się w arkuszu *obliczenia*. Wartości z arkusza *obliczenia* zostaną wyświetlone na wykresie ukazując przebieg ruchu, czyli tor poruszającego się ciała.

Makro – jest to program napisany w języku VBA (Visual Basic for Application), który wykonuje zapisane w nim instrukcje automatyzując nam 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.

Całkowity czas ruchu (czyli suma czasu wznoszenia i czasu spadania: $t_c = t_w + t_s$) zostanie w tej symulacji podzielony na 100 równych przedziałów – w komórce E2 możesz obserwować aktualną wartość kroku, czyli bieżący przedział trwania symulacji. Bieżąca wartość czasu symulacji, wyrażona w sekundach, wyświetlana jest w komórce B1.

Poniżej wykresu umieściliśmy pasek przewijania, który jest powiązany z komórką E2 i pozwala na zmianę wartości z zakresu od 0 do 100 – odpowiadają one poszczególnym przedziałom trwania symulacji. Zmieniając wartość przedziału za pomocą paska zaobserwuj jak zmienia się czas lotu ciała w komórce B1.

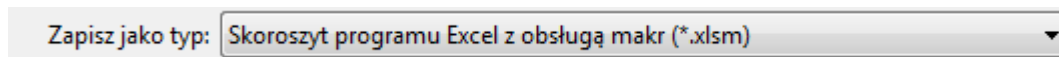
Zwróć uwagę, że jeśli będziesz klikać na strzałkach paska przewijania, na wykresie będzie się przemieszczała duża czerwona kropka ukazując bieżące położenie ciała wyrzuconego z parametrami podanymi w komórkach B2:B5.

Sprawdź, że zmiana jakiegokolwiek wartości w komórkach B2:B5 (sformatowanych na zielono) spowoduje wyzerowanie wartości w komórce E2 oraz tabeli z wartościami x i y w arkuszu *obliczenia*, co będzie skutkowało również „wyzerowaniem” wykresu – nie będzie widoczny wykres składający się z małych granatowych kropek.

Jeśli chcesz z powrotem zobaczyć na wykresie pełny tor, czyli przeprowadzić pełną symulację rzutu ukośnego z zadanymi parametrami, kliknij ponownie na przycisku **start**.

Wykonanie

Do wykonania symulacji rzutu ukośnego potrzebny będzie pusty plik Excela. Zapisz go na dysku swojego komputera pod nazwą *rzut_ukośny_symulacja*. Poniżej pola z nazwą pliku znajdziesz pole *Zapisz jako typ* (Rysunek 3). Wybierz w nim *Skoroszyt programu Excel z obsługą makr*. Wykonanie tej akcji spowoduje, że plik otrzyma rozszerzenie *.xslm* zamiast typowego rozszerzenia *.xlsx*.



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

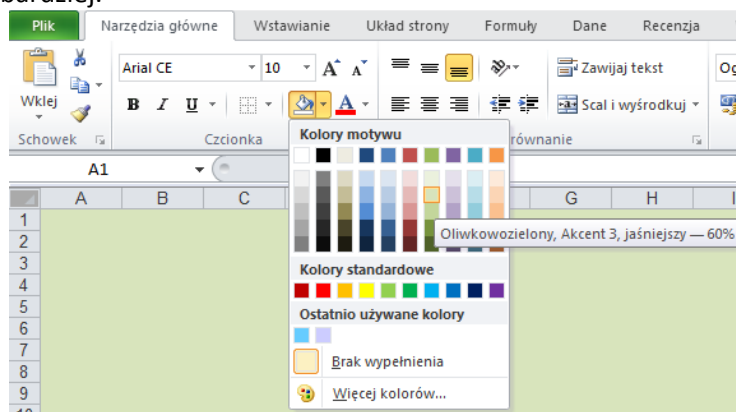
W pliku *rzut_ukośny_symulacja.xslm* zmień nazwę pierwszego arkusza na *rzut ukośny*, drugiemu nadaj nazwę *obliczenia*. Pozostałe arkusze możesz usunąć – nie będą nam potrzebne.



Rysunek 4. Nazwy arkuszy w pliku *rzut_ukośny_symulacja*

Tworzenie symulacji rzutu ukośnego rozpoczniemy od przygotowania danych w arkuszu *rzut_ukośny*. 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 (Rysunek 5) – wybierz ten, który lubisz najbardziej.



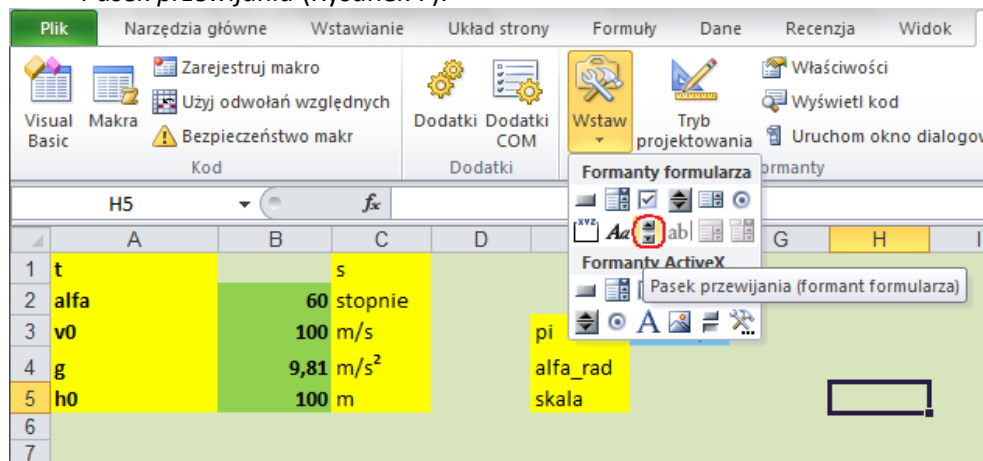
Rysunek 5. Zmiana tła komórek w arkuszu *rzut_ukośny*

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. Zajmiemy się teraz ich wprowadzeniem. W tym celu postępuj następująco: do komórki A1 wpisz tekst „t”, do komórki A2 – „alfa” itd. W ten sposób wpisz do arkusza wszystkie etykiety. Tło komórek zawierających etykiety sformatuj na żółto (rys. 6).

	A	B	C	D	E	F
1	t		s			
2	alfa		stopnie			
3	v0		m/s		pi	
4	g		m/s ²		alfa_rad	
5	h0		m		skala	
6						
7						
8						
9	czas wznoszenia		s			
10	Hmax		m			
11	czas spadania		s			
12	Zasięg		m			
13						
14						
15	x		m			
16	y		m			
17	Vx		m/s			
18	Vy		m/s			
19						

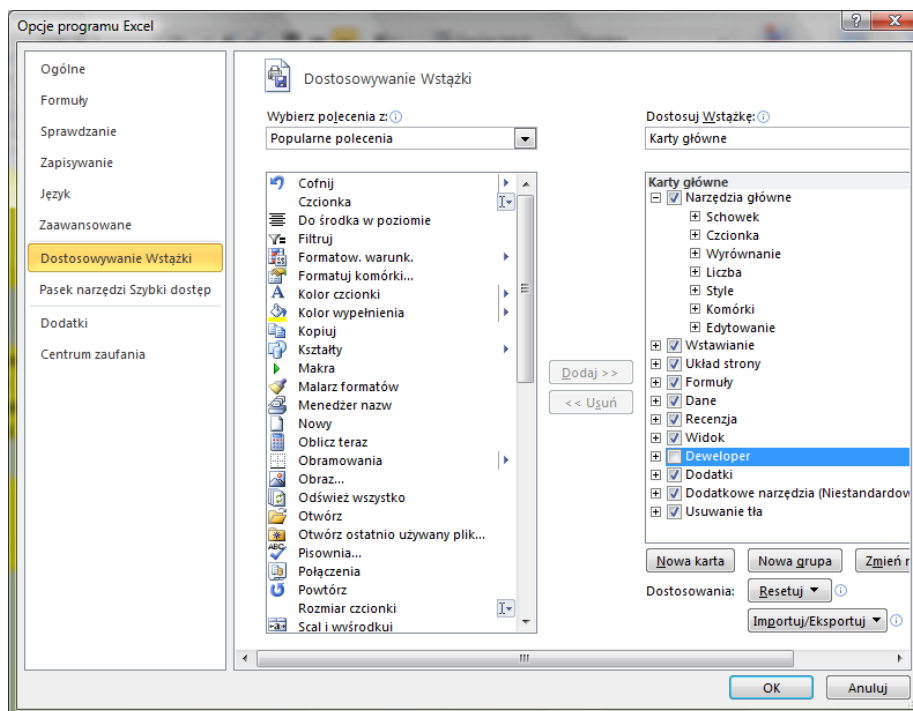
Rysunek 6. Wygląd arkusza *rzut_ukośny* po wykonaniu pkt. 1 i 2 instrukcji

- Teraz wprowadzimy parametry, które, jako użytkownik, będziesz mógł modyfikować: do komórki B2 wpisz wartość kąta rzutu wyrażoną w stopniach, na przykład 60. Do komórki B3 wprowadź wartość 100, B4: 9,81 i do B5 wartość 100. Tło komórek sformatuj na zielono.
- Do komórki F3 wprowadź wartość liczby π : 3,14. Tło komórki sformatuj na niebiesko.
- Spójrz jeszcze raz na rysunek 2. Poniżej wykresu znajduje się pasek przewijania, postaraj się wstawić go mniej-więcej tym samym miejscu, co w pliku wzorcowym (np. w wierszu 27, w zakresie kolumn od G do Q). W tym celu na karcie *Developer* (jeśli nie widzisz jej na swojej wstążce, w następnym punkcie opisaliśmy jak ją wyświetlić), w grupie *Formanty* znajdź polecenie *Wstaw* i z wewnętrznej listy wybierz formant *Pasek przewijania* (Rysunek 7).



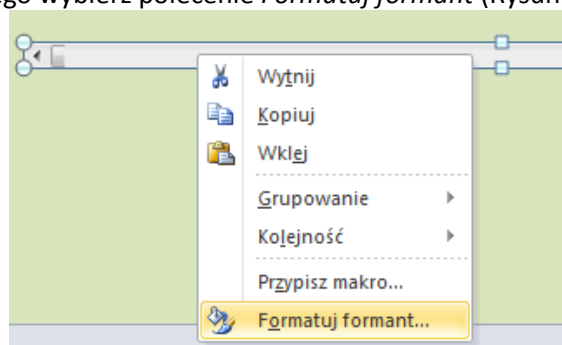
Rysunek 7. Wstawianie paska przewijania

- Uwaga:** jeżeli na wstążce nie ma karty *Developer*, 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 *Developer* (Rysunek 8). Od tej pory będzie ona na stałe widoczna na Twojej wstążce.



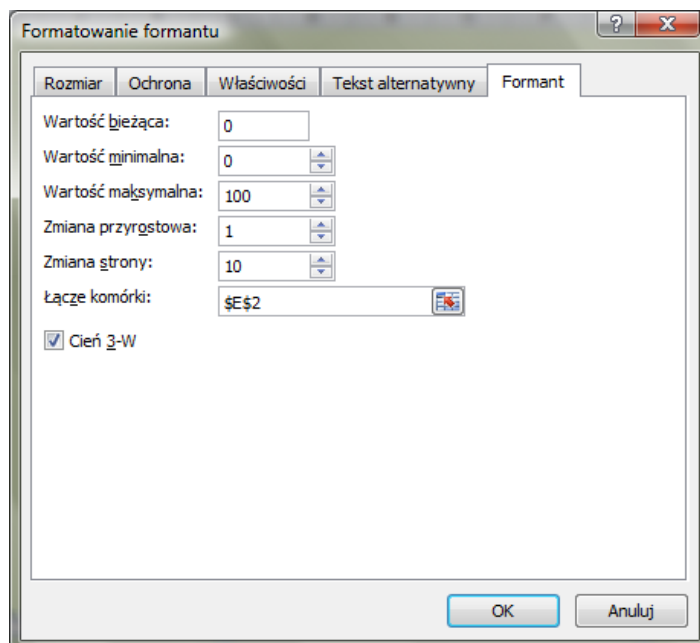
Rysunek 8. Włączanie karty Deweloper

7. „Narysuj” pasek przewijania w odpowiednim miejscu arkusza (spójrz na rysunek 2). Następnie kliknij prawym przyciskiem myszki na pasku przewijania i z menu podręcznego wybierz polecenie *Formatuj formant* (Rysunek 9).



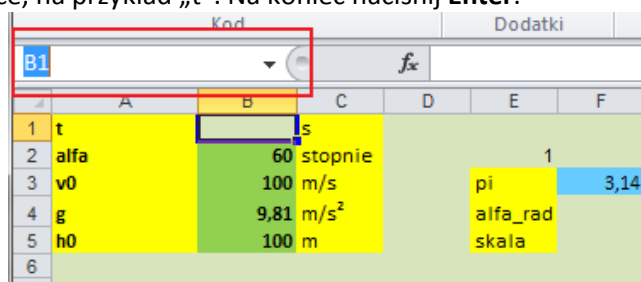
Rysunek 9. Z menu podręcznego wybierz polecenie *Formatuj formant*

8. W oknie *Formatowanie formantu* wprowadź parametry jak na rysunku 10. Pasek będzie obsługiwał wartości od zera (*Wartość minimalna*) do 100 (*Wartość maksymalna*). W komórce E2 (jej adres wprowadź w polu *Łącze komórki*) będzie się wyświetlała bieżąca wartość paska przewijania, czyli jedno ze stu położań ciała w naszej symulacji rzutu ukośnego. Podczas klikania na strzałki paska przewijania, wartości w komórce E2 będą się zmieniały o 1 (świadczy o tym ustawienie w polu *Zmiana przyrostowa*). Po kliknięciu „w pole” paska (czyli gdziekolwiek pomiędzy strzałkami), wartość w komórce E2 zmieni się o 10 (pole *Zmiana strony*). Zatwierdź wprowadzone ustawienia przyciskiem **OK**.



Rysunek 10. Parametry paska przewijania

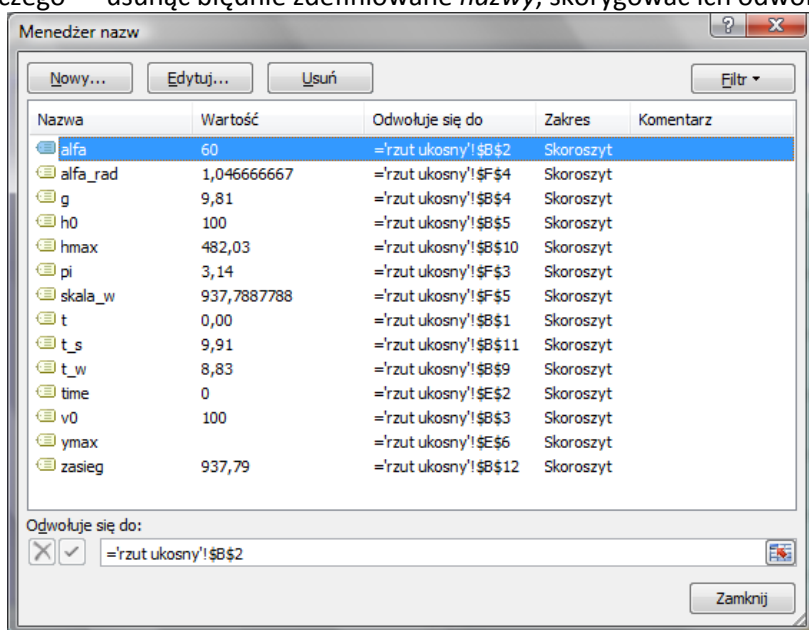
9. Odkliknij zaznaczenie paska (kliknij w dowolnym miejscu arkusza, aby pozbyć się zaznaczenia z paska przewijania). Wypróbuj jego działanie. Zaobserwuj, jak zmieniają się wartości w komórce E2.
10. Kolejnym krokiem tworzenia symulacji będzie wprowadzenie formuł do arkusza. Najpierw jednak, dla wygody, 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*.
11. Komórce B1 nadamy nazwę „t”. Uaktywnij komórkę B1 (kliknij na niej). Zwróć uwagę, że adres aktywnej komórki pojawił się w tzw. *Polu nazwy* (pole otoczone czerwoną obwódką na rysunku 11). Kliknij w *Polu nazwy* i wpisz nazwę, którą chcesz nadać komórce, na przykład „t”. Na koniec naciśnij **Enter**.



Rysunek 11. Lokalizacja *Pola nazwy*, za pomocą którego nadajemy *nazwy* komórkom

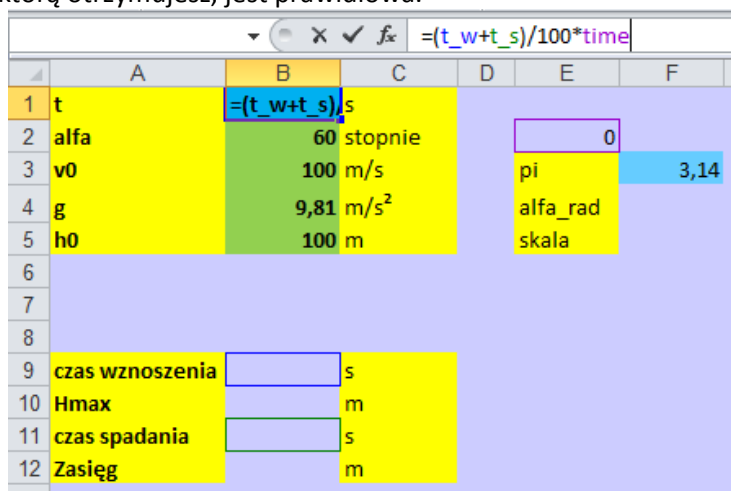
12. Zaznacz komórkę B2 i postępując w sposób opisany w poprzednim punkcie nadaj jej nazwę *alfa*.
Uwaga: jeśli zechcesz nadać komórce nazwę wielocłonową, nie używaj spacji! Zamiast niej możesz użyć znaku podkreślenia, np. *kąt_alfa*, lub wpisać nazwę w taki sposób: *KątAlfa* – bez odstępów, ale zachowując czytelność nazwy.
13. W podobny sposób nadaj nazwy pozostałym komórkom, i tak, komórkę B3 nazwij „v0”, B4 – „g”, B5 – „h0”, B9 – „t_w”, B10 – „hmax”, B11 – „t_s”, B12 – „zasieg”, E2 – „time”, F3 – „pi”, F4 – „alfa_rad”, F5 – „skala_w”.
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 skoroszytcie. Możesz w nim – „w razie czego” – usunąć błędnie zdefiniowane *nazwy*, skorygować ich odwołanie itp.



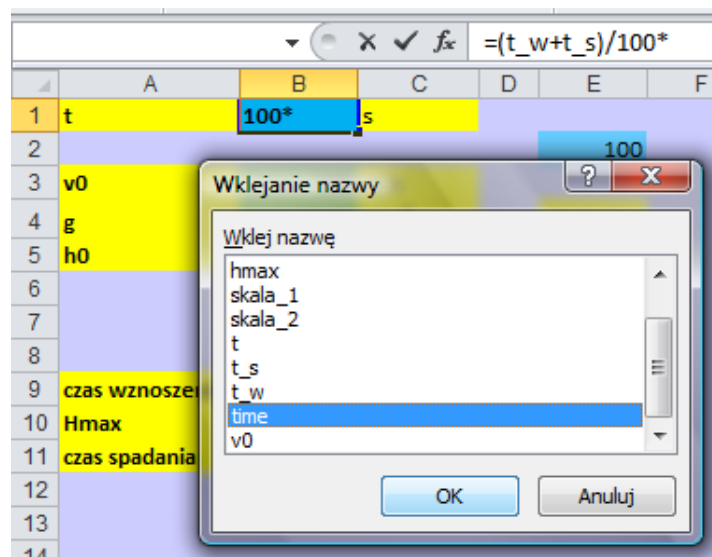
Rysunek 12. Okno *Menedżera nazw* z listą zdefiniowanych *nazw*, ich bieżące wartości (u Ciebie na tym etapie wykonywania ćwiczenia większość wartości będzie równa zero), odwołania do komórek itp.

14. Najwyższa pora zająć się obliczeniami! Na początek w komórce B1 wyznaczmy czas (wyrażony w sekundach), jaki upłynął od momentu wyrzucenia ciała. Wprowadź do komórki B1 formułę, jak na rysunku 14. Spójrz na swój *Pasek formuły* i sprawdź czy formuła, którą otrzymujesz, jest prawidłowa.



Rysunek 13. Formuła w komórce B1

15. **Uwaga:** 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 *Wstawianie nazwy*, które zawiera spis wszystkich *nazw* występujących w skoroszytcie. Okno *Wstawianie nazwy* wywołasz naciskając klawisz funkcyjny **F3**.



Rysunek 14. Tworzenie formuły z użyciem okna *Wstawianie nazwy*

16. W komórce F4 przeliczymy na radiany wartość kąta w stopniach z komórki B2, według następującej formuły:

$$f_x = \text{pi} * \text{alfa} / 180$$

Rysunek 15. Formuła w komórce F4

17. W komórce B9 obliczymy czas wznoszenia ciała rzuconego pod kątem α . We wzorze do obliczenia t_w (4) występuje funkcja trygonometryczna sinus. Sinus kąta α policzymy w Excelu za pomocą funkcji arkuszowej SIN(). Jest ona łatwa w użyciu. Najwygodniej będzie wpisać ją „z ręki”.

$$f_x = v0 * \text{SIN}(\text{alfa_rad}) / g$$

Rysunek 16. Formuła w komórce B9

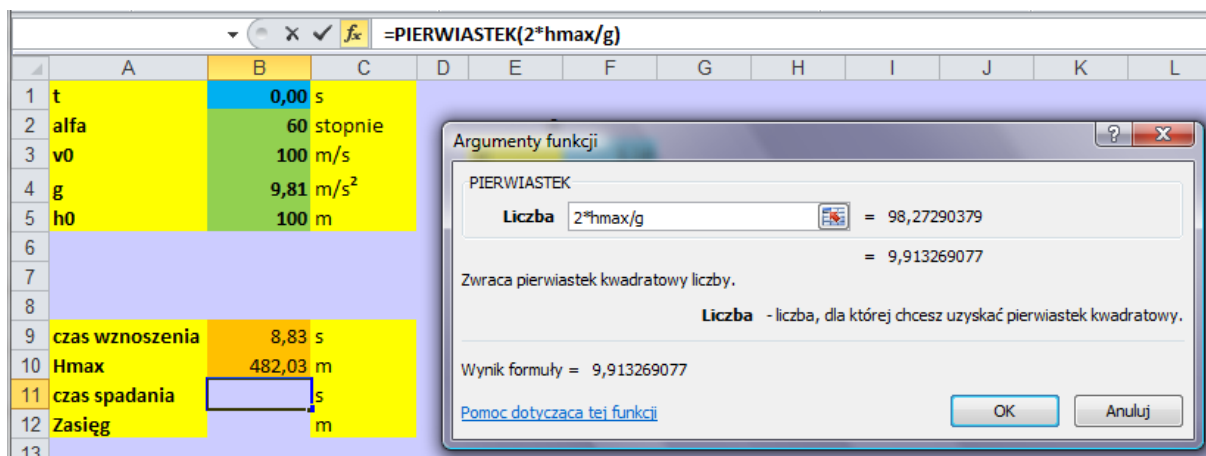
18. Do komórki B10 wprowadź formułę do obliczenia maksymalnej wysokości (6), jaką osiągnie ciało w rzucie ukośnym:

$$f_x = h0 + (v0^2 * (\text{SIN}(\text{alfa_rad}))^2) / (2 * g)$$

Rysunek 17. Formuła w komórce B10

19. W komórce B11 obliczymy czas spadania t_s z wysokości H_{max} według wzoru (7). We wzorze tym występuje pierwiastek. Aby obliczyć pierwiastek w Excelu musimy zastosować funkcję PIERWIASTEK (z kategorii *Matematyczne i trygonometryczne*). Możemy, podobnie jak to robiliśmy z funkcją SIN(), wpisać ją ręcznie, albo wstawić ją korzystając z narzędzi do wstawiania funkcji. Przybliżymy tutaj ten drugi sposób. Aby przygotować formułę z rysunku 18, zaznacz komórkę B11, 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**.

Uwaga: zwróć uwagę, że kiedy wysokość początkowa h_0 w komórce B5 jest równa zero (nie ma przesunięcia wysokości, ciało spada na takiej samej wysokości z jakiej jest wyrzucane), wówczas $t_w = t_s$ – czas wznoszenie jest równy czasowi spadania.



Rysunek 18. Formuła w komórce B11

20. W komórce B12 wyliczymy zasięg poziomy rzutu. COS() w formule oznacza funkcję cosinus.

$$f_x = =v0 * \text{COS}(alfa_rad) * (t_w + t_s)$$

Rysunek 19. Formuła w komórce B12

21. Współrzędna x opisująca chwilowe położenie ciała w komórce B15:

$$f_x = =v0 * \text{COS}(alfa_rad) * t$$

Rysunek 20. Formuła w komórce B15

22. Współrzędna y (komórka B16):

$$f_x = =h0 + v0 * \text{SIN}(alfa_rad) * t - g * (t^2) / 2$$

Rysunek 21. Formuła w komórce B16

23. Bieżąca wartość poziomej składowej prędkości v_x (komórka B17):

$$f_x = =v0 * \text{COS}(alfa_rad)$$

Rysunek 22. Formuła w komórce B17

24. Bieżąca wartość pionowej składowej prędkości v_y (komórka B18):

$$f_x = =v0 * \text{SIN}(alfa_rad) - g * t$$

Rysunek 23. Formuła w komórce B18

25. W komórce F5 obliczymy wartość, która pomoże nam później utworzyć wykres wyglądający proporcjonalnie. Docelowo będziemy chcieli aby skala na obu osiach wykresu obejmowała taki sam zakres wartości.

W formule korzystamy z funkcji MAX() (kategoria *Statystyczne*), która w wyniku swojego działania zwraca wartość największą ze zbioru wartości podanych jako argumenty. W naszym przypadku funkcja MAX() zwraca większą z dwóch wartości: zasięgu pionowego i poziomego. Musimy wybrać większą z wartości, aby wykres nie został „ucięty”.

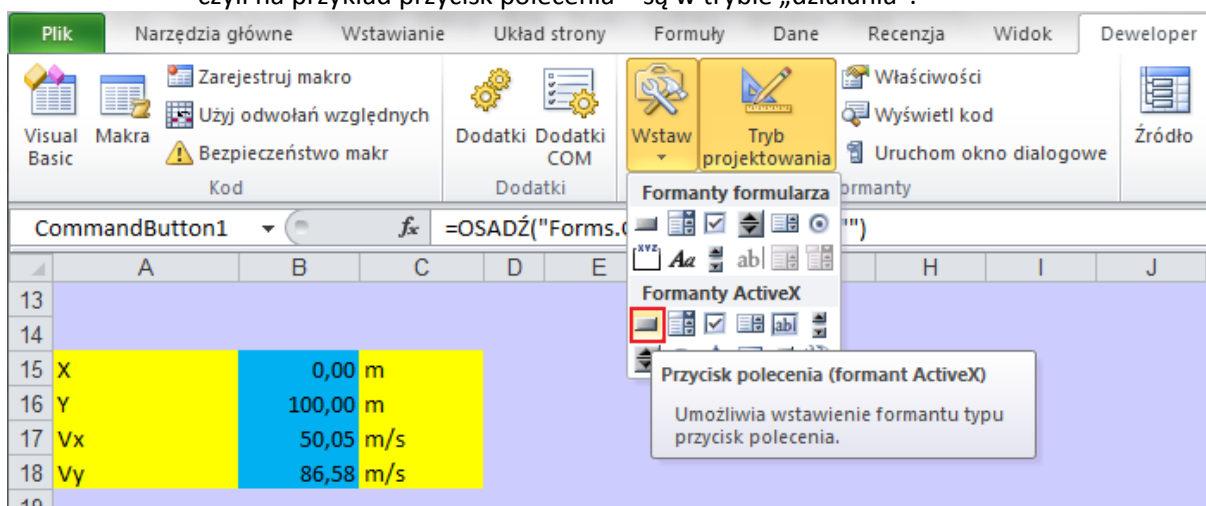
	A	B	C	D	E	F
1	t	0,00	s			
2	alfa	60	stopnie		0	
3	v0	100	m/s		pi	3,14
4	g	9,81	m/s ²		alfa_rad	1,046667
5	h0	100	m		skala	
6						
7						
8						
9	czas wznoszenia	8,83	s			
10	Hmax	482,03	m			
11	czas spadania	9,91	s			
12	Zasięg	937,79	m			
13						

Rysunek 24. Formuła w komórce F5

26. Do tej pory zajmowaliśmy się wyłącznie arkuszem *rzut ukosny*. Teraz przygotujemy dane w arkuszu *obliczenia*, w zakresie komórek A1:B101, które będą bazą do wykonania wykresu obrazującego ruch ciała w rzucie ukośnym. Dane w arkuszu *obliczenia* będą wyliczane za pomocą makra (dla parametrów wprowadzonych przez użytkownika w arkuszu *rzut ukosny*).

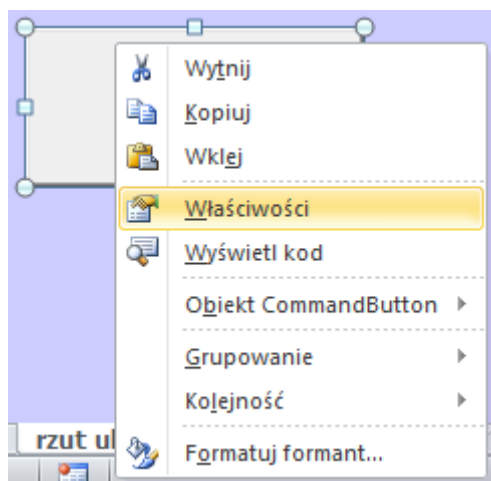
Przygotowania rozpoczniemy od wstawienia przycisku w arkuszu *rzut ukosny*, który będzie uruchamiał makro. Aby wstawić przycisk przejdź na kartę *Deweloper*, w sekcji *Formanty* kliknij na ikoncie *Wstaw* i z wewnętrznej listy, z grupy *Formanty ActiveX*, wybierz *Przycisk polecenia* (Rysunek 25). „Narysuj” przycisk poniżej komórek z danymi, gdzieś w zakresie wierszy 22-25 i kolumn A-C. 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”.



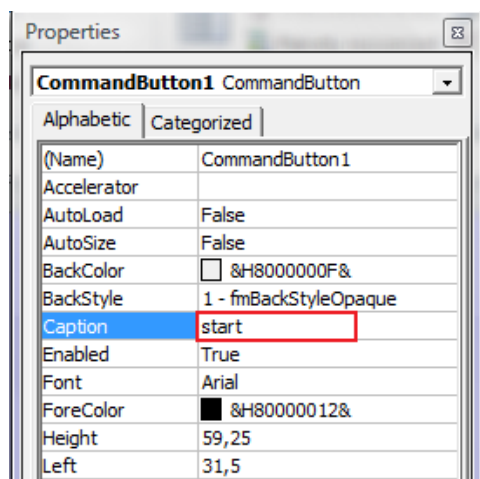
Rysunek 25. Wstawianie Przycisku polecenia, który będzie uruchamiał makro

27. Kliknij prawym klawiszem myszki na wstawionym przycisku i z menu podręcznego wybierz *Właściwości* (Rysunek 26).



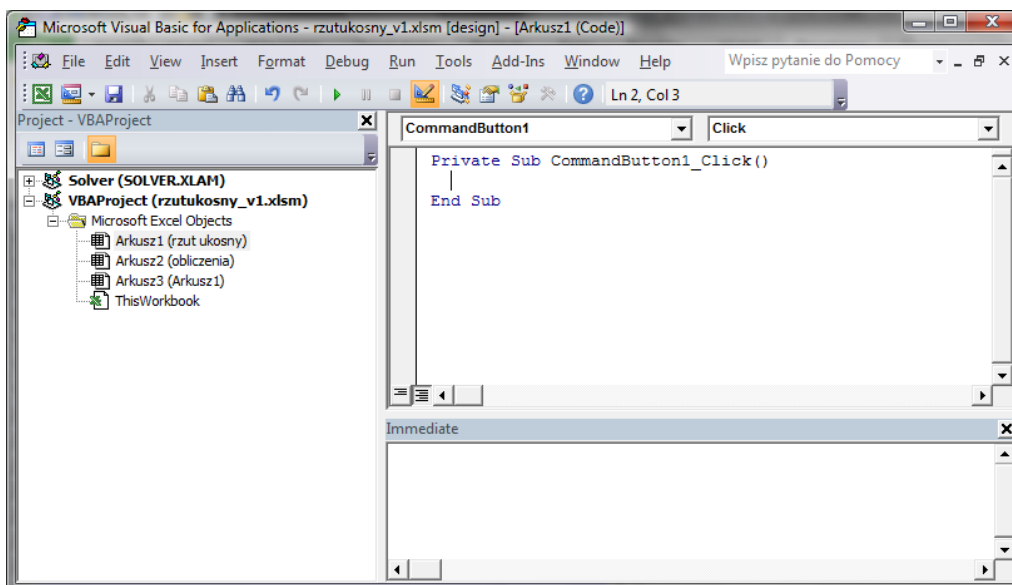
Rysunek 26. Menu podręczne Przycisku polecenia

28. W oknie *Properties*, którego fragment widzisz na rysunku 27, 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 27. Właściwości Przycisku polecenia

29. 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 (Rysunek 28). 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 rysunku 28 – widzisz w nim słowa: *Private Sub* itd.).



Rysunek 28. Okno edytora VBA

30. 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.

31. W części służącej do programowania, w miejscu, gdzie widoczny jest kursor, wpisz kod przedstawiony na rysunku 29. Jest to procedura, która uruchomi się po kliknięciu na przycisku **start** umieszczonym w pierwszym arkuszu.

```

CommandButton1 Click
Private Sub CommandButton1_Click()

    For i = 0 To 100
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = 0
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = 0
    Next i

    For i = 0 To 100
        ThisWorkbook.Worksheets(1).cells(2, 5) = i
        Range("A1").Activate
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = ThisWorkbook.Worksheets(1).cells(15, 2)
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = ThisWorkbook.Worksheets(1).cells(16, 2)
    Next i

End Sub

```

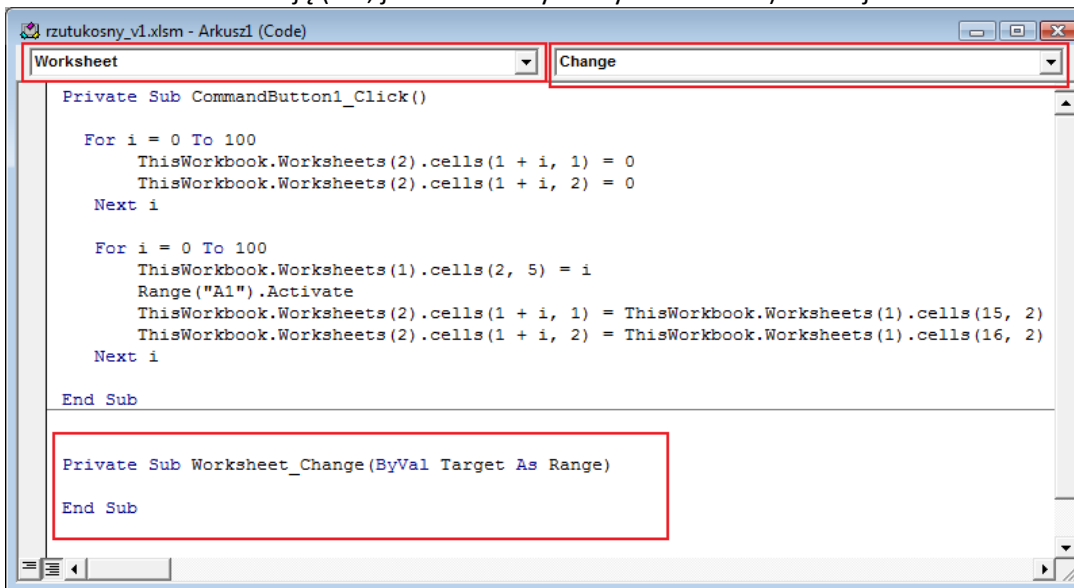
Rysunek 29. Kod procedury, która wyliczy współrzędne x,y położenia ciała w rzucie ukośnym

W procedurze z rysunku 29 widzisz dwie pętle *For ... Next*. Pierwsza z nich wykona się 101 razy (zmienna *i* – licznik pętli – będzie przyjmowała wartości od zera do 100, 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 101, 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 2 i numerze kolumny 5 (chodzi oczywiście o komórkę E2) pierwszego arkusza (czyli *rzut ukosny*) 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 100 przedziałów czasowych). Druga instrukcja oznacza po prostu: „zaznacz komórkę A1 w aktywnym arkuszu”. Instrukcja trzecia to podstawianie kolejno do komórek A1:B101 wartości współrzędnych *x,y* położenia ciała w rzucie ukośnym wyliczanych w komórkach B15 i B16 w arkuszu pierwszym.

32. Po przepisaniu procedury z rysunku 29 ustaw kursor tekstowy poza procedurą (poniżej słów kluczowych *End Sub*) i za pomocą list wyboru znajdujących się ponad edytorem tekstu wybierz pozycję *Worksheet* z listy po lewej stronie i *Change* – po prawej stronie. 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:B101 w arkuszu drugim.

Uwaga: jeżeli przy okazji, niechcący, w edytorze pojawi się jakaś inna procedura, zaznacz ją (tak, jak w dowolnym edytorze tekstu) i skasuj klawiszem **Delete**.

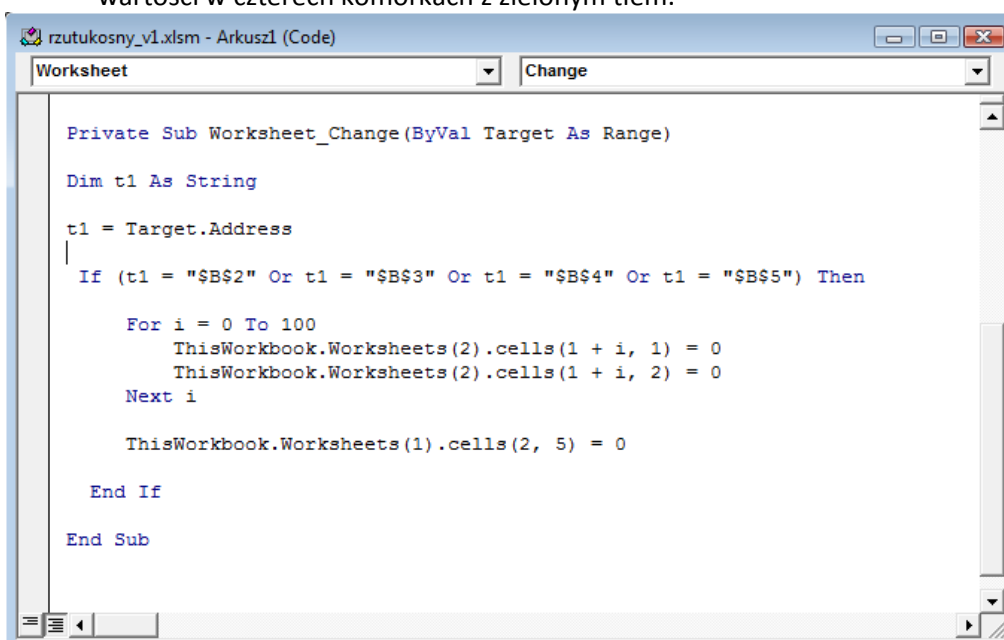


The screenshot shows the VBA Code Editor window titled "rzutukosny_v1.xlsm - Arkusz1 (Code)". The "Worksheet" dropdown menu is selected on the left, and the "Change" dropdown menu is selected on the right. The code in the editor is as follows:

```
Private Sub CommandButton1_Click()  
  
    For i = 0 To 100  
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = 0  
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = 0  
    Next i  
  
    For i = 0 To 100  
        ThisWorkbook.Worksheets(1).cells(2, 5) = i  
        Range("A1").Activate  
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = ThisWorkbook.Worksheets(1).cells(15, 2)  
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = ThisWorkbook.Worksheets(1).cells(16, 2)  
    Next i  
  
End Sub  
  
Private Sub Worksheet_Change(ByVal Target As Range)  
  
End Sub
```

Rysunek 30. Wybór procedury zdarzeniowej *Worksheet_Change*


33. Pomędzy słowami kluczowymi (*Sub* i *End Sub*) procedury *Worksheet_Change* wprowadź kod zamieszczony na rysunku 31. Poprzednia procedura nie zawierała żadnych argumentów, tutaj występuje zmienna *Target*, która jest typu zakres (*Range*). Zaprogramujemy procedurę tak, aby „reagowała” wyłącznie na zmianę wartości w czterech komórkach z zielonym tłem.

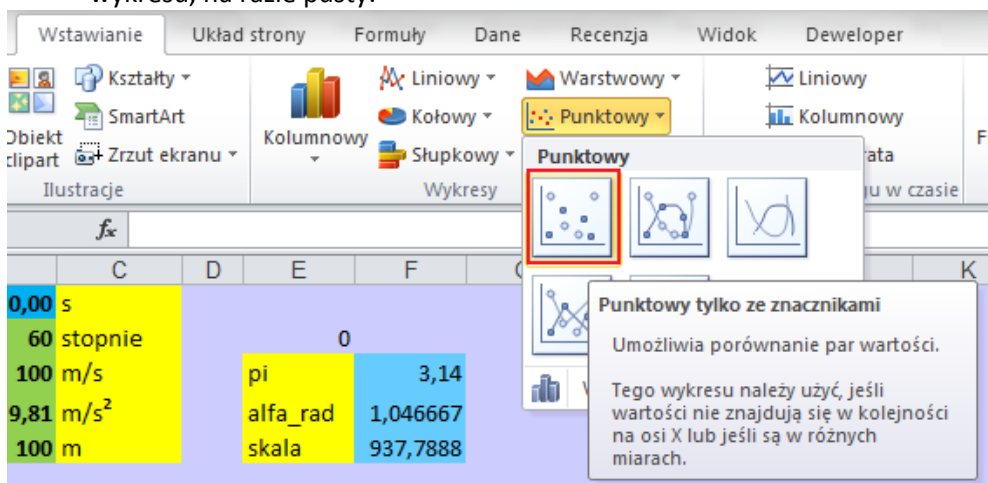


The screenshot shows the VBA Code Editor window titled "rzutukosny_v1.xlsm - Arkusz1 (Code)". The "Worksheet" dropdown menu is selected on the left, and the "Change" dropdown menu is selected on the right. The code in the editor is as follows:

```
Private Sub Worksheet_Change(ByVal Target As Range)  
  
    Dim t1 As String  
  
    t1 = Target.Address  
    If (t1 = "$B$2" Or t1 = "$B$3" Or t1 = "$B$4" Or t1 = "$B$5") Then  
  
        For i = 0 To 100  
            ThisWorkbook.Worksheets(2).cells(1 + i, 1) = 0  
            ThisWorkbook.Worksheets(2).cells(1 + i, 2) = 0  
        Next i  
  
        ThisWorkbook.Worksheets(1).cells(2, 5) = 0  
  
    End If  
  
End Sub
```

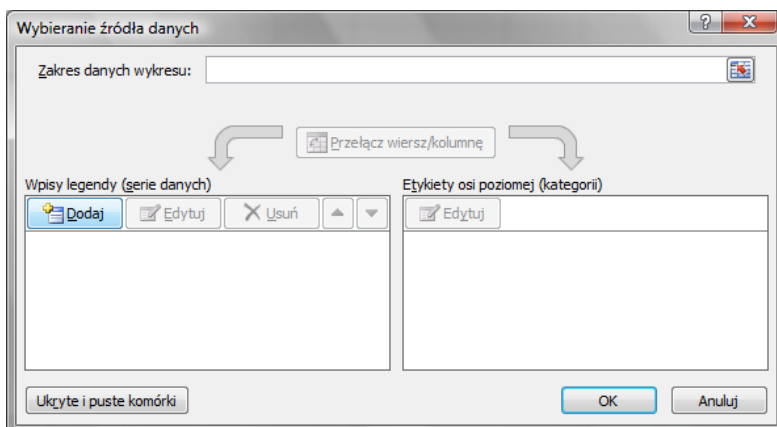
Rysunek 31. Kod procedury *Worksheet_Change*

34. W pierwszej linii kodu została zadeklarowana zmienna `t1` jako zmienna typu tekstowego (*String*). Instrukcja `t1 = Target.Address` oznacza: „zmiennej `t1` przypisz adres komórki (zakresu komórek), w której nastąpiła zmiana wartości”. Instrukcja warunkowa `If ... End If` sprawdza warunek, czyli, czy zmiana nastąpiła w jednej z komórek z zakresu z zielonym tłem (chodzi o komórki B2 lub B3 lub B4 lub B5). Jeśli warunek jest spełniony, pętla `For ... Next` (znanej nam doskonale z poprzedniej procedury) zeruje wartości w tabelce w arkuszu drugim. Instrukcja `ThisWorkbook.Worksheets(1).cells(2, 5) = 0` wstawia dodatkowo wartość 0 do komórki E2.
35. Po wpisaniu kodu obu procedur, zapisz wszystko – możesz w tym celu użyć standardowej ikony  dostępnej na pasku narzędziowym lub kombinacji klawiszy **Ctrl+S**.
36. **Uwaga:** Zwróć uwagę, że edytor VBA jest oddzielnym 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). W każdej chwili możesz z edytora VBA przejść do skoroszytu Excela (bez zamykania go) używając *Paska zadań* (na dole monitora).
37. Mając wykonane wszystkie obliczenia zajmiemy się przygotowaniem wykresu. Zaznacz dowolną pustą komórkę arkusza *rzut ukosny* znajdująca się poza obszarem danych (na przykład H2).
38. Przejdź na kartę *Wstawianie*. W grupie *Wykresy* wybierz typ *Punktowy*, a następnie podtyp *Tylko ze znacznikami* (Rysunek 32). W arkuszu powinien pojawić się obszar wykresu, na razie pusty.



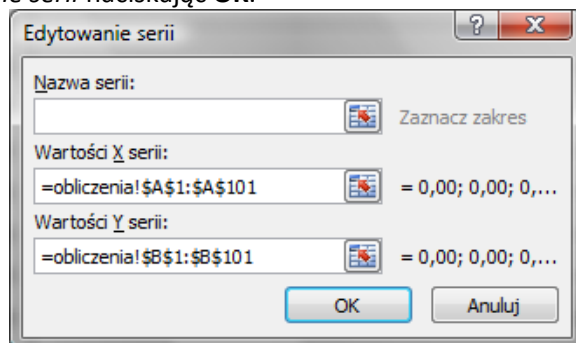
Rysunek 32. Wybór wykresu, typ Punktowy

39. 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 Rysunek 33). Kliknij na przycisku *Dodaj*.



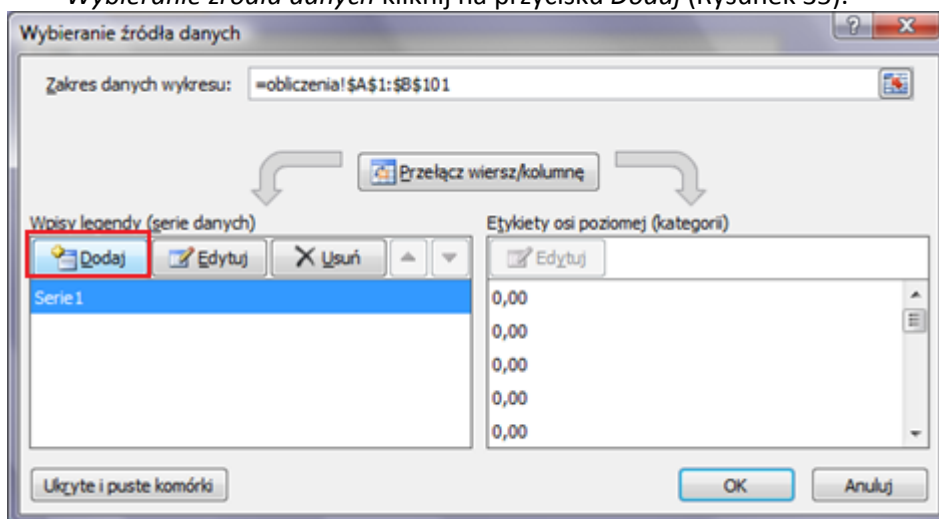
Rysunek 33. Dodawanie serii danych do wykresu

40. W oknie *Edytowanie serii* (Rysunek 34) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek od A1 do A101. Następnie przestaw kursor do pola *Wartości Y serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek od B1 do B101. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



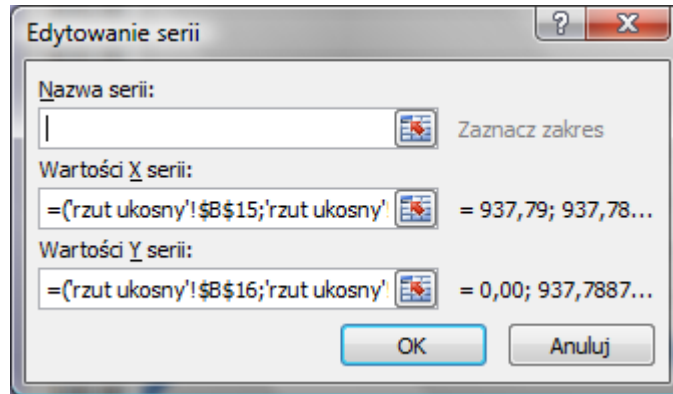
Rysunek 34. Definiowanie pierwszej serii danych

41. W oknie *Wybieranie źródła danych* jest zdefiniowana pierwsza seria, czyli współrzędne wszystkich położenia x, y (czyli tor) ciała w rzucie ukośnym przy zadanych parametrach. Dodamy jeszcze drugą serię danych, która będzie się składała z zaledwie dwóch punktów. Będzie ona – przede wszystkim – pokazywała bieżące położenie ciała. Drugim jej zadaniem będzie wyskalowanie wykresu w taki sposób, by na obu osiach układu współrzędnych był dokładnie taki sam zakres wartości. W oknie *Wybieranie źródła danych* kliknij na przycisku *Dodaj* (Rysunek 35).



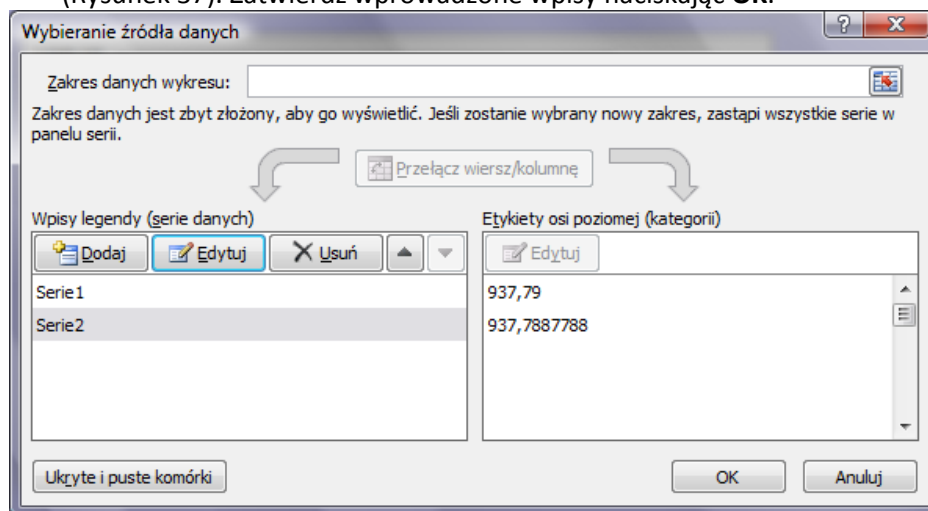
Rysunek 35. Okno Wybieranie źródła danych ze zdefiniowaną pierwszą serią

42. W oknie *Edytowanie serii* (Rysunek 36) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii* i zaznacz komórkę B15 oraz dodatkowo – przytrzymując wciśnięty klawisz **Ctrl** – komórkę F5. Następnie przestaw kursor do pola *Wartości Y serii* i zaznacz komórkę B16 oraz – dodatkowo, z wciśniętym klawiszem **Ctrl** – komórkę F5. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



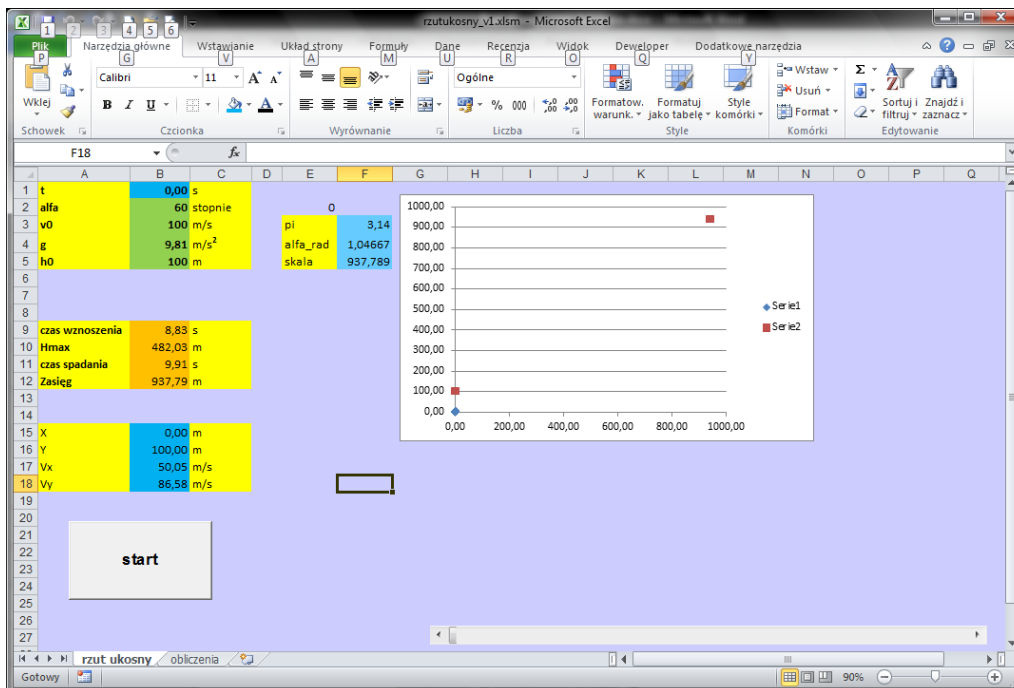
Rysunek 36. Definiowanie drugiej serii danych

43. Spójrz na okno *Wybieranie źródła danych*. Pojawiły się w nim wpisy obu serii danych (Rysunek 37). Zatwierdź wprowadzone wpisy naciskając **OK**.



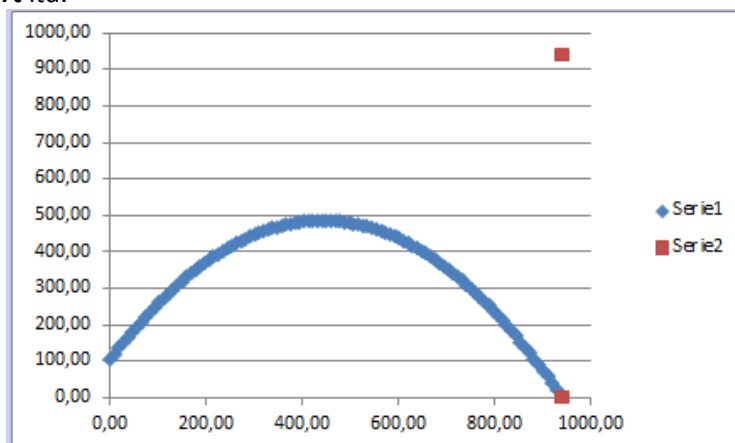
Rysunek 37. Wpisy dwóch serii danych w oknie *Wybieranie serii danych*

44. Po zatwierdzeniu wpisów w oknie *Wybieranie serii danych* w arkuszu powinien pojawić się wykres składający się z dwóch serii danych, jak na rysunku 38.



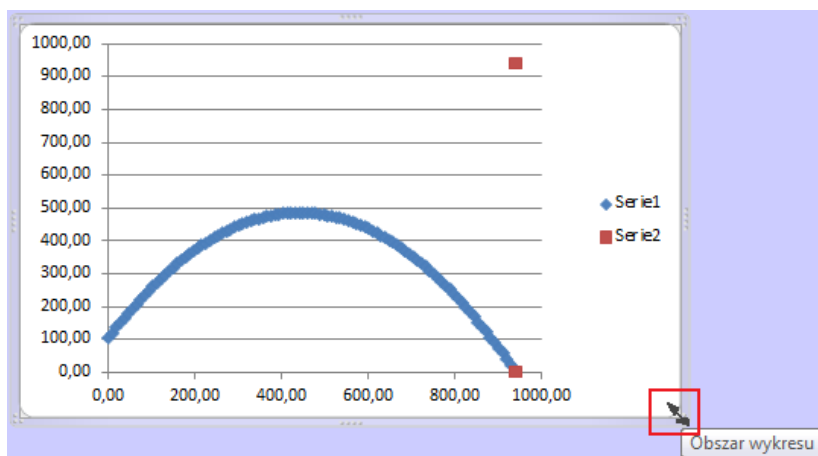
Rysunek 38. Wygląd arkusza rzut ukosny z wykresem

45. 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 3 punkty. Jeśli uruchomisz teraz przyciskiem **start** symulację rzutu ukośnego, do komórek A1:B101 w arkuszu *obliczenia* zostaną podstawione niezerowe wartości i wykres będzie wyglądał jak na rysunku 39. Jeśli wprowadzisz zmianę którejkolwiek wartości w komórkach z zielonym tłem, komórki w arkuszu *obliczenia* zostaną wyzerowane (za sprawą makra *Worksheet_change*) i wykres ponownie będzie wyglądał jak na rysunku 38, dopóki ponownie nie uruchomisz symulacji przyciskiem **start** itd.



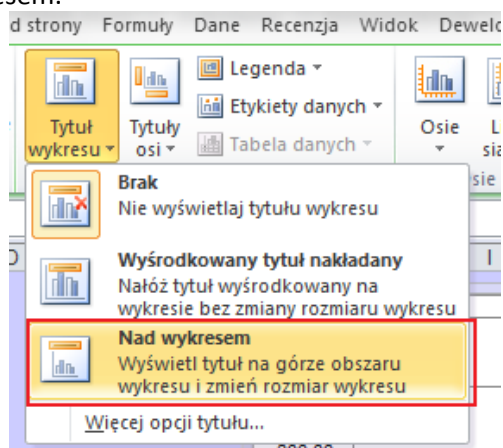
Rysunek 39. Wykres przedstawiający symulację rzutu ukośnego

46. 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:R. Operując kursorem myszki w rogach wykresu uzyskaj znacznik do zmiany rozmiaru (Rysunek 40) i powiększ wykres.



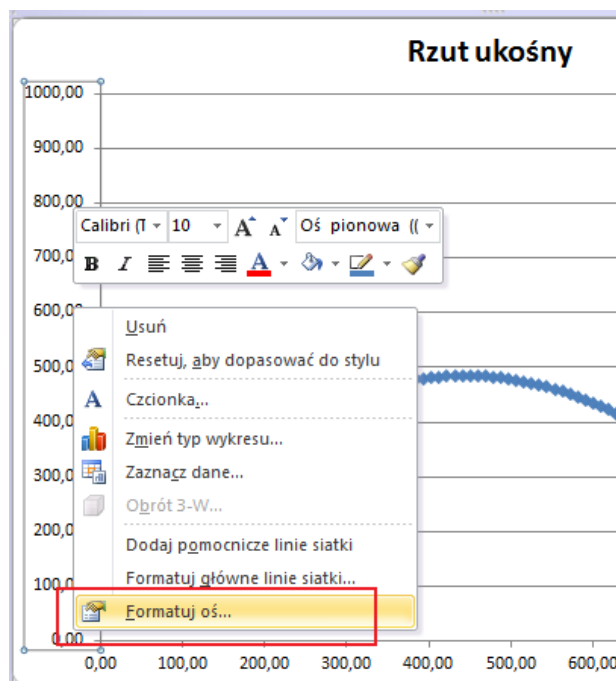
Rysunek 40. Najedź na róg wykresu – pojawi się znacznik służący do zmiany jego rozmiaru

47. Przejdź na kartę *Układ* (znajdącą 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.
48. Na karcie *Układ* w sekcji *Etykiety* wybierz *Tytuł* wykresu, a następnie *Nad wykresem*, jak na rysunku 41. Wpisz tekst „Rzut ukośny” i naciśnij **Enter**. Tytuł powinien pojawić się nad wykresem.



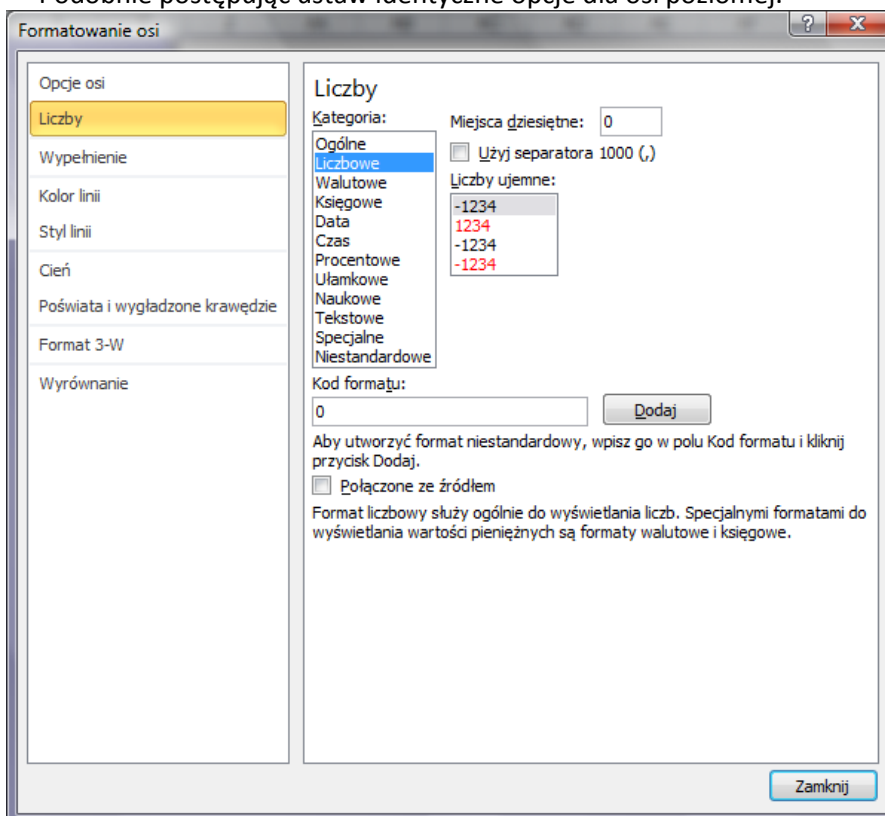
Rysunek 41. Dodawanie tytułu wykresu

49. 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.
50. Kliknij jeszcze raz na jednej z wartości, ale tym razem prawym klawiszem myszki, i z menu podręcznego wybierz *Formatuj oś* (Rysunek 42).



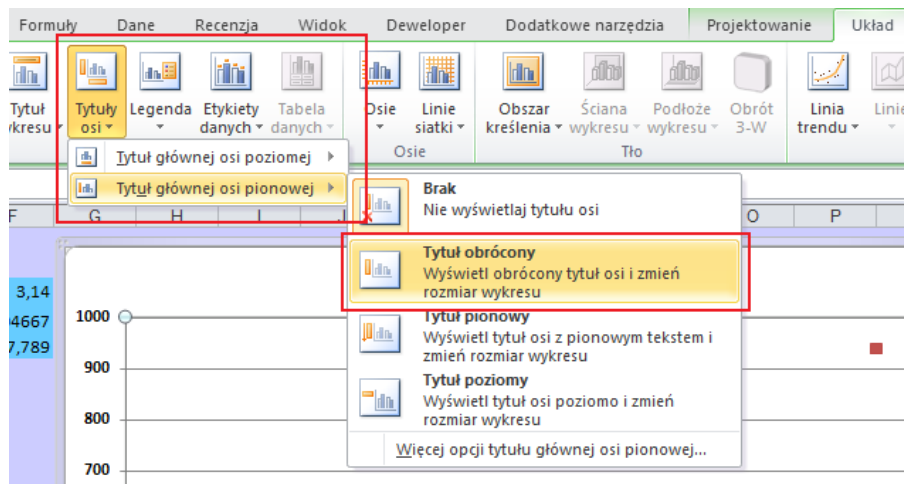
Rysunek 42. Menu podręczne osi pionowej

51. Po lewej stronie okna *Formatowanie* osi wybierz kartę *Liczby*. Na karcie *Liczby*, w prawej części okna, wybierz kategorię *Liczbowe*, w polu *Miejsca dziesiętne* wpisz 0, wyłącz separator tysięcy (Rysunek 43). Na koniec kliknij na przycisku **Zamknij**. Podobnie postępując ustaw identyczne opcje dla osi poziomej.



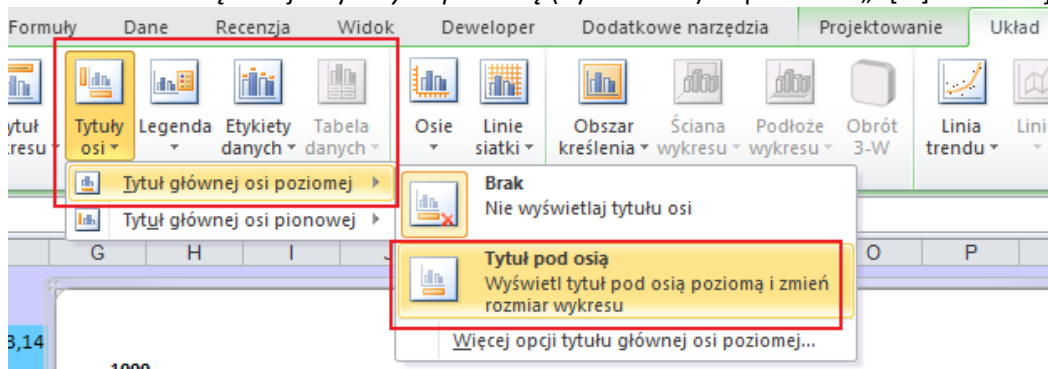
Rysunek 43. Opcje wyświetlania liczb na osi pionowej

52. Przejdź na kartę *Układ* i w sekcji *Etykiety* wybierz *Tytuły osi*, a następnie *Tytuł głównej osi pionowej* i z wewnętrznej listy *Tytuł obrócony* (Rysunek 44). Wpisz tekst „Y[m]” i naciśnij **Enter**.



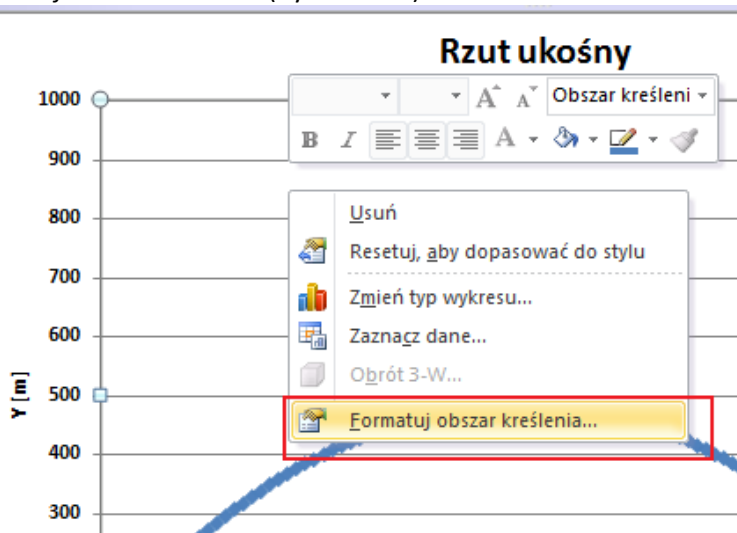
Rysunek 44. Dodanie tytułu osi pionowej

53. Ponownie wybierz polecenie *Tytuły osi*, a następnie *Tytuł głównej osi poziomej* i – z wewnętrznej listy – *Tytuł pod osią* (Rysunek 45). Wpisz tekst „X[m]” i naciśnij **Enter**.



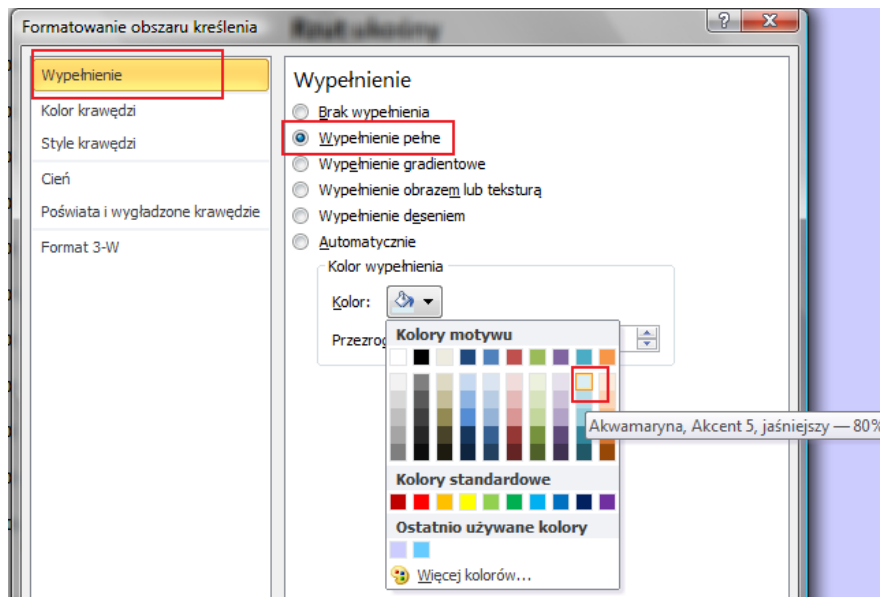
Rysunek 45. Dodanie tytułu osi poziomej

54. Kliknij prawym klawiszem myszki gdzieś w tle wykresu i z menu podręcznego wybierz *Formatuj obszar kreślenia* (Rysunek 46).



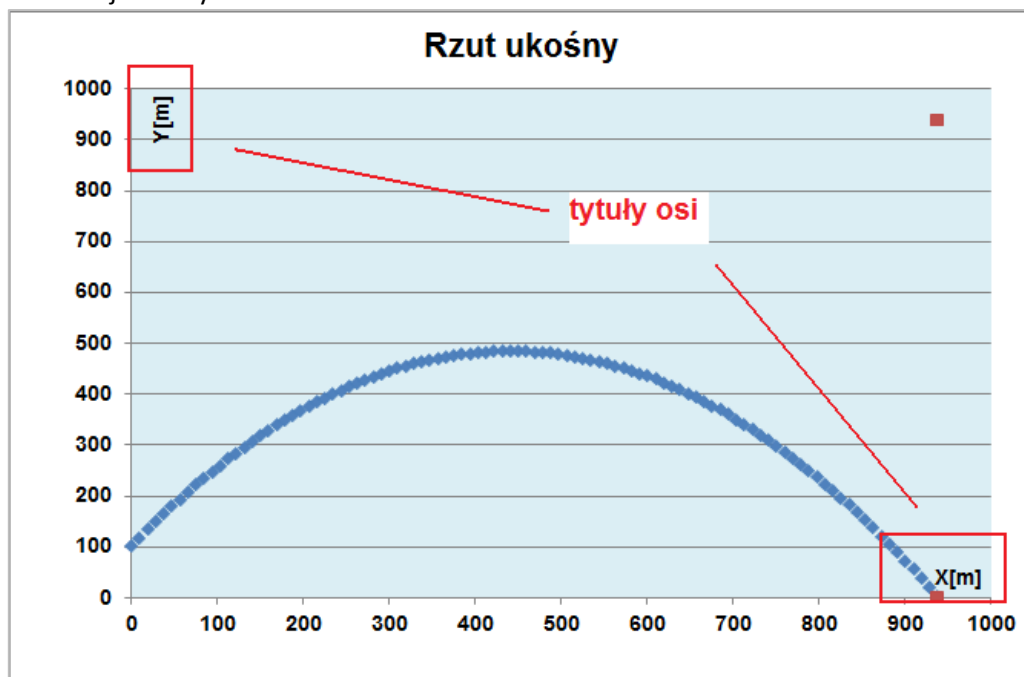
Rysunek 46. Wybór polecenia do sformatowania tła wykresu

55. W oknie *Formatowanie obszaru kreślenia*, na karcie *Wypełnienie* wybierz opcję *Wypełnienie pełne*, a następnie jakiś jasnoniebieski kolor (np. Akwamaryna, Akcent 5 – patrz rysunek 47). Potwierdź ustawienia przyciskiem **Zamknij**.



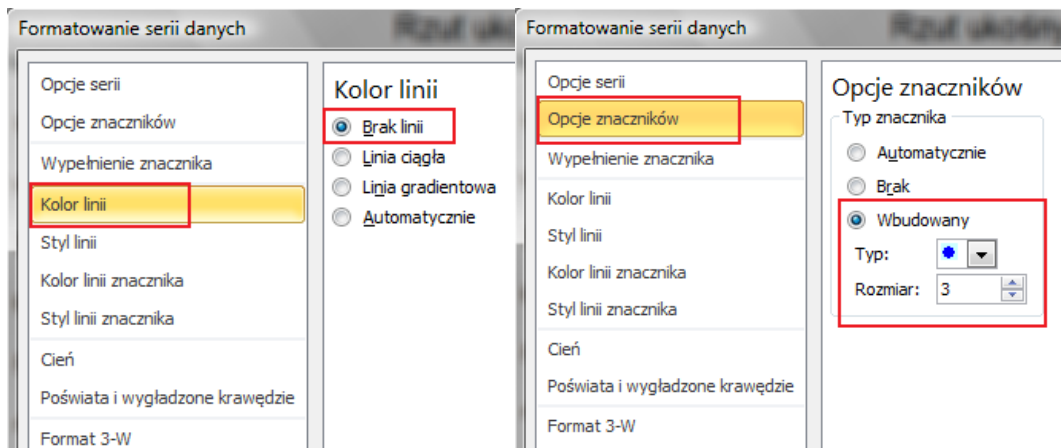
Rysunek 47. Wybór koloru

56. Metodą „przecignij i puść” (chodzi o „złapanie” kursorem myszki jakiegoś obiektu i przeciągnięcie go w odpowiednie miejsce) ustaw podpisy osi, aby całość wyglądała jak na rysunku 48.



Rysunek 48. Wykres po sformatowaniu osi oraz zmianie koloru tła

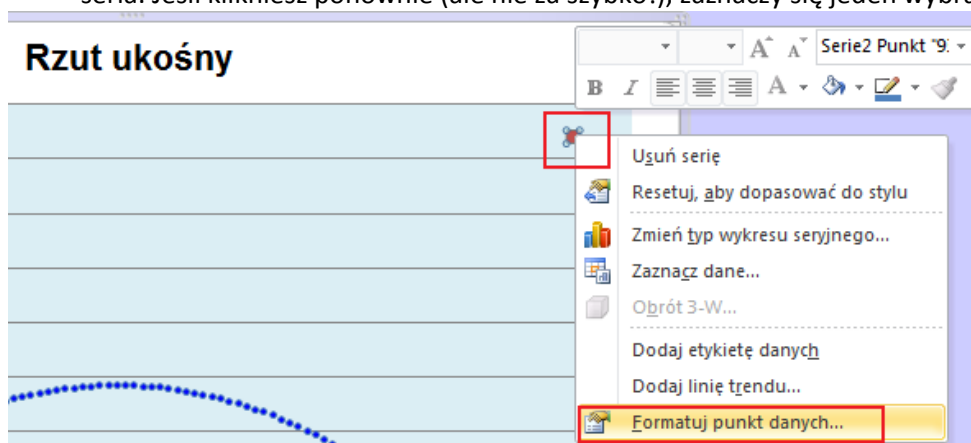
57. Kliknij prawym klawiszem myszki na jednym ze znaczników pierwszej serii danych (punkty z arkusza *obliczenia*) i z menu podręcznego wybierz polecenie *Formatuj serię danych*.
58. W oknie *Formatuj serię danych* wprowadź ustawienia jak na rysunku 49: karta *Kolor linii* wybierz: brak linii, karta *Opcje znaczników*: typ wbudowany, okrągły, o rozmiarze 3. Na karcie *Wypełnienie znacznika* wybierz odpowiedni kolor – na przykład intensywnie niebieski, jak na wykresie wzorcowym.



Rysunek 49. Formatowanie znaczników pierwszej serii danych

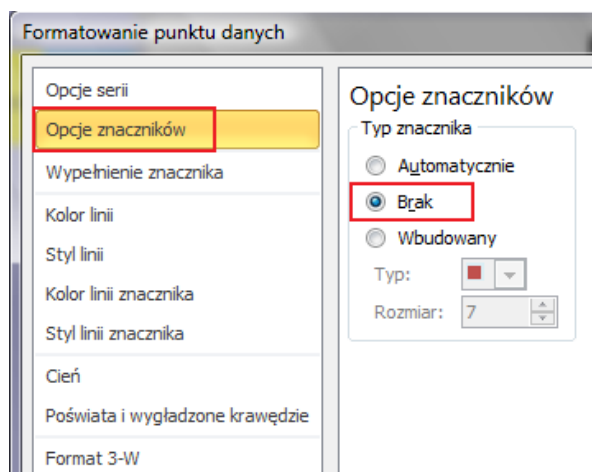
59. „Ukryjemy” teraz punkt drugiej serii danych, który nie przedstawia żadnych istotnych treści fizycznych, a służy nam jedynie do skalowania wykresu. W tym celu kliknij prawym klawiszem na punkcie w prawym górnym rogu drugiej serii danych, po chwili kliknij ponownie i z menu podręcznego wybierz polecenie *Formatuj punkt danych* (Rysunek 50).

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.



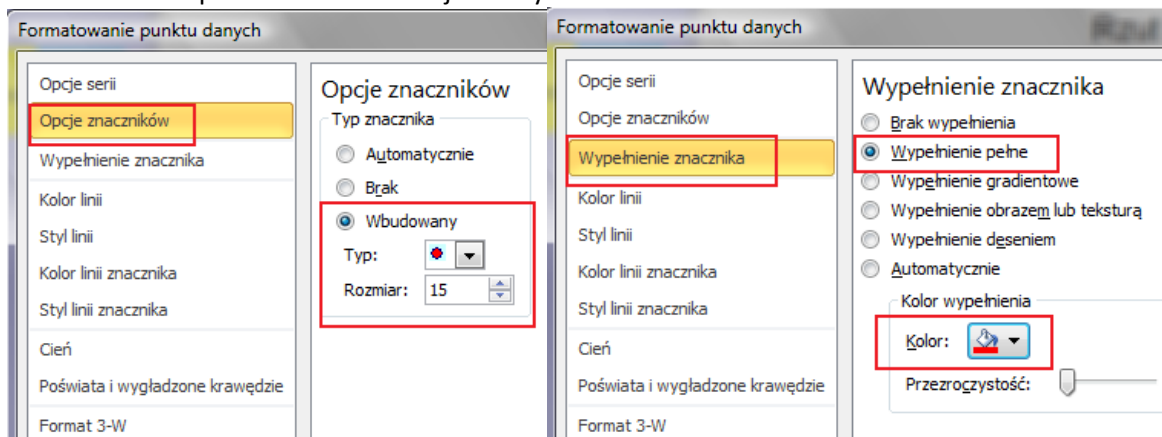
Rysunek 50. Menu podręczne punktu drugiej serii danych

60. W oknie *Formatuj punkt danych* na karcie *Opcje znaczników* wybierz: *Brak* (Rysunek 51).



Rysunek 51. Formatowanie punktu drugiej serii danych

61. Został nam jeszcze jeden punkt do sformatowania – ten, który symuluje ruch ciała po parabolicznym torze w rzucie ukośnym. Zaznacz ten punkt (przypominamy, że musisz kliknąć na nim wolno dwa razy, jeśli klikniesz raz – zaznaczą się wszystkie punkty należące do tej serii). Kliknij na zaznaczonym punkcie prawym przyciskiem myszki i z menu podręcznego wybierz *Formatuj punkt danych*. W oknie *Formatuj punkt danych* wprowadź ustawienia jak na rysunku .



Rysunek 52. 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 ukośnego.

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** lub klikając myszą na specjalnie utworzonym przycisku umieszczonym w arkuszu.

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 zwartego 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 takich, 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 Basic, 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 Dodatku.

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 Dodatku.

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 inne 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*: załóżmy, 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				