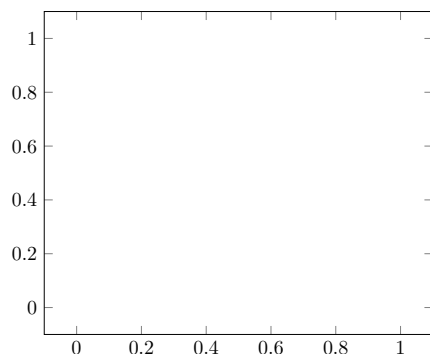


1 Podstawowe informacje

Celem laboratorium jest zapoznanie się z tworzeniem i stylizowaniem wykresów przy użyciu pakietów `tikz` oraz `pgfplots`¹. Pakiet `tikz` jest bardzo rozbudowanym narzędziem do tworzenia grafiki, rysunków i schematów. Jego uniwersalność niesie ze sobą duży poziom skomplikowania i obszerność dokumentacji². Niemniej, w trakcie dzisiejszych zajęć skupimy się wyłącznie na tworzeniu wykresów.

Rysunki tworzone z wykorzystaniem pakietu `tikz` trzeba umieszczać wewnątrz otoczenia `tikzpicture`, w innym wypadku nie będą działać. Przy użyciu opcji `scale` możemy odpowiednio przeskalować cały rysunek. Do rysowania wykresów będziemy używać otoczenia `axis`, które utworzy kartezjański układ współrzędnych.



```
1 \begin{tikzpicture}[scale=0.75]
2   \begin{axis}
3   \end{axis}
4 \end{tikzpicture}
```

Obszar wykresu możemy dostosować przy użyciu następujących opcji otoczenia `axis`:

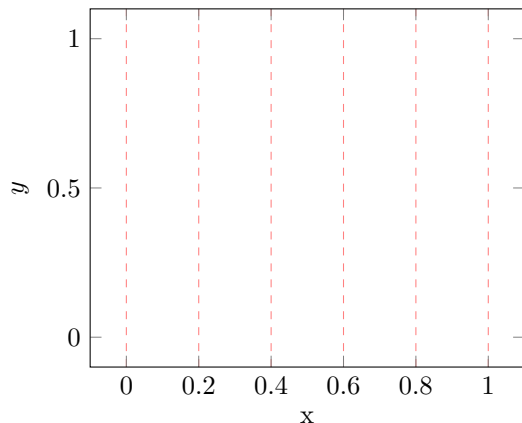
<code>xlabel</code>	podpis osi x
<code>ylabel</code>	podpis osi y
<code>xmin, xmax</code>	minimalna i maksymalna wartość na osi x
<code>ymin, ymax</code>	minimalna i maksymalna wartość na osi y
<code>xtick</code>	wartości na osi x
<code>ytick</code>	wartości na osi y
<code>xmajorgrids</code>	pokaż linie pionowe siatki
<code>ymajorgrids</code>	pokaż linie poziome siatki
<code>grid style</code>	typ siatki
<code>legend pos</code>	pozycja legendy
<code>mark sizes</code>	wielkość znaczników

W \LaTeX -ie do zamierzonego efektu można dojść na różne sposoby. Na przykład, jeśli chcemy aby linie siatki pionowe jak i poziome były widoczne, możemy osobno ustawić dla nich widoczność (`xmajorgrids=true` oraz `ymajorgrids=true`) lub zrobić to jednocześnie za pomocą `grid=both`. Domyślnie linie siatki są wyrysowane ciągłą, jasno-szarą linią, ale ich wygląd możemy zmienić według własnych preferencji przy ustawieniu opcji `grid style`. W wypadku więcej niż jednej wartości, należy je umieszczać w nawiasach klamrowych, np. gdy chcemy jednocześnie zmienić ciągłość linii oraz jej kolor `grid style = {dashed, red!50}`. Wartości na osiach są dobierane automatycznie, ale możemy ręcznie ustawić zarówno początek osi OX (`xmin = 0`) lub koniec

¹Zaraz po wczytaniu paczki `pgfplots`, należy wywołać makro `\pgfplotsset{compat=newest}`. Pozwala to uzyskać dostęp do jej najnowszej implementacji, co ma spore znaczenie przy rysowaniu wykresów.

²<https://sunsite.icm.edu.pl/pub/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>

(`xmax=10`) oraz analogicznie dla osi OY . Możemy również wskazać dokładnie, które wartości mają być uwzględnione na konkretnej osi, np.: (`xtick={1, 2, 3, 4, 5}`). Oś układu podpisujemy za pomocą opcji `xlabel` oraz `ylabel`. Domyślnie podpisujemy w trybie tekstowym, ale przy użyciu `$$` możemy opisywać osie w trybie matematycznym. Dla wygody warto ustawić oczekiwaną szerokość wykresu, w naszym wypadku posłużymy się szerokością tekstu `width=\textwidth`.



```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     xlabel=x,
5     ylabel=$y$,
6     xmajorgrids=true,
7     grid style={dashed, red!50},
8   ]
9   \end{axis}
10 \end{tikzpicture}

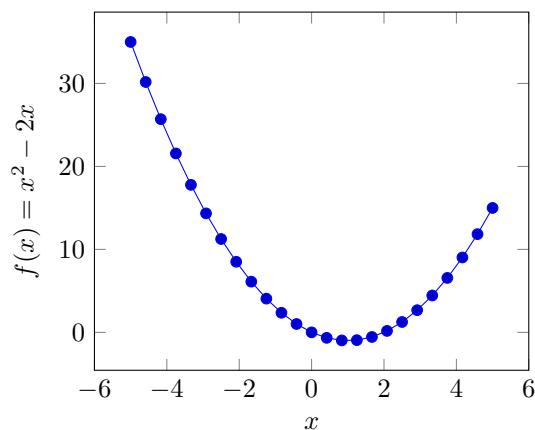
```

1.1 Wykres na podstawie wzoru

Podstawowym poleceniem z paczki `pgfplots` służącym do tworzenia wykresu jest makro:

$$\backslash\text{addplot}[\text{opcje}]\{\text{wzór_funkcji}\};$$

Makro to jako argument przyjmuje wzór funkcji. Przy mnożeniu konieczne jest używanie symbolu „`*`”. Jeśli chcemy narysować wykres funkcji sinus, w zależności od stopni trzeba użyć konstrukcji `sin(deg(x))`. Istotną zmianą w otoczeniu `tikzpicture` jest to, że po każdym rozkazie należy umieścić średnik „`;`” na końcu.

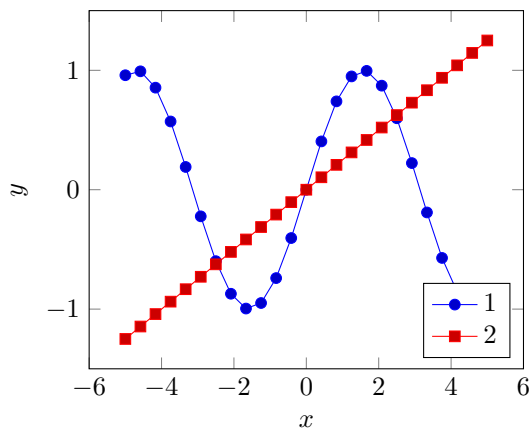


```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     xlabel=$x$,
5     ylabel={f(x) = x^2 - 2x}
6   ]
7   \addplot{x^2 - 2*x};
8   \end{axis}
9 \end{tikzpicture}

```

Jeśli chcemy narysować więcej funkcji na jednym wykresie, należy dodać kolejny `addplot` wewnątrz tego samego otoczenia `axis`. Domyślnie każda dodawana funkcja będzie miała inny kolor oraz kształt znaczników (`marks`). W celu rozróżnienia wykresów funkcji można stosować legendę, dostępną za pomocą polecenia `\legend{lista_etykiet}`, gdzie po przecinku „`,`” umieszczamy kolejno opisy dla konkretnych wykresów w kolejności ich wywołania. Domyślnie legenda pojawia się w prawym górnym rogu (`north east`) obszaru wykresu, ale możemy zmienić jej położenie z pomocą opcji `legend pos = south east` dla otoczenia `axis`.

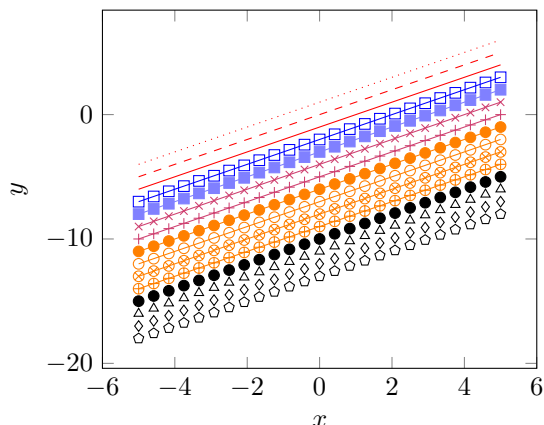


```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     xlabel=$x$,
5     ylabel=$y$,
6     legend pos=south east
7   ]
8   \addplot{sin(deg(x))};
9   \addplot{x/4};
10  \legend{1, 2}
11  \end{axis}
12 \end{tikzpicture}

```

Możemy dowolnie zmieniać ustawienia rysowanych funkcji. Zmiana koloru następuje poprzez wprost podanie koloru `\addplot[kolor]{wzór_funkcji}`; lub `color=kolor` jako opcji dla konkretnej funkcji. Kolory mamy zdefiniowane w pakiecie `xcolor`. Standardowo możemy sterować nasyceniem kolorów przy użyciu „!””. Poprzez opcje `mark=znacznik` możemy wybrać z puli znaczników jeden konkretny (na rysunku poniżej pokazano wiele przykładów). Dodanie „*” przy znaczniku spowoduje jego wypełnienie. Dzięki opcji `no marks` możemy całkowicie zrezygnować ze znaczników lub przy pomocy `only marks` zostawić tylko znaczniki. Użycie opcji `dashed` spowoduje narysowanie przerywanej linii wykresu funkcji a `dotted` wykropkowaną. Istnieje możliwość tworzenia cyklicznych list z ustawieniami dla kolejno wywoływanych funkcji, co jeszcze wygodniejszą opcją niż ręczne modyfikowanie wywołania każdej funkcji.



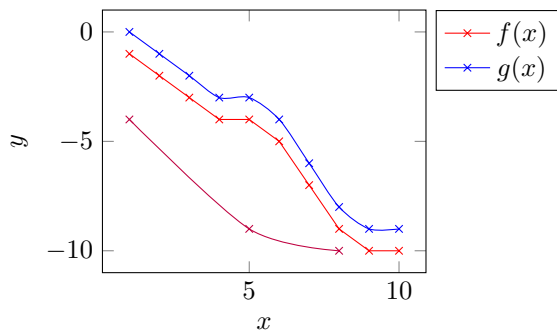
```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     xlabel=$x$,
5     ylabel={ $y$ }
6   ]
7   \addplot[dotted, red, no marks]{x+1};
8   \addplot[dashed, red, no marks]{x};
9   \addplot[red, no marks]{x-1};
10  \addplot[blue, mark=square]{x-2};
11  \addplot[blue!50, mark=square*]{x-3};
12  \addplot[purple!75, mark=x]{x-4};
13  \addplot[purple!75, mark=+]{x-5};
14  \addplot[orange, mark=*]{x-6};
15  \addplot[orange, mark=ol]{x-7};
16  \addplot[orange, mark=otimes]{x-8};
17  \addplot[orange, mark=oplus]{x-9};
18  \addplot[black, mark=*, only marks]{x-10};
19  \addplot[mark=triangle, only marks]{x-11};
20  \addplot[mark=diamond, only marks]{x-12};
21  \addplot[mark=pentagon, only marks]{x-13};
22  \end{axis}
23 \end{tikzpicture}

```

1.2 Wykres z punktów

W przypadku potrzeby ręcznego wyrysowania wykresu funkcji na podstawie punktów należy argument rozkazu `\addplot{}` poprzedzić słowem kluczowym `coordinates`. Należy jeden umieścić je po opcjach. Punkty podajemy kolejno wewnątrz nawiasów okrągłych w postaci pary (x_A, y_A) . Wartości oddzielamy przecinkiem „,”. **Separatorem dziesiętnym** w systemie \LaTeX jest **kropka** „.”. Wartości można podać obok w jednej lub w wielu liniach, nie należy jednak zostawiać pustych linii. Opcje rysowanych przebiegów funkcji zmienia się tak samo jak przy wcześniejszym wywołaniu funkcji `addplot`. Jeśli chcemy wygładzić przebieg funkcji należy użyć opcji `smooth`. Poprzez zastosowanie opcji `legend pos = outer north east` możemy przesunąć legendę poza obszar wykresu. Jednak działa to jedynie dla pozycji koło górnego prawego rogu. oczywiście przez ręczne dobranie pozycji moglibyśmy przesunąć legendę w dowolne miejsce na czy poza obszarem wykresu. Jeśli lista podanych etykiet w rozkazie `legend` nie będzie kompletna to brakujące elementy zostaną pominięte.



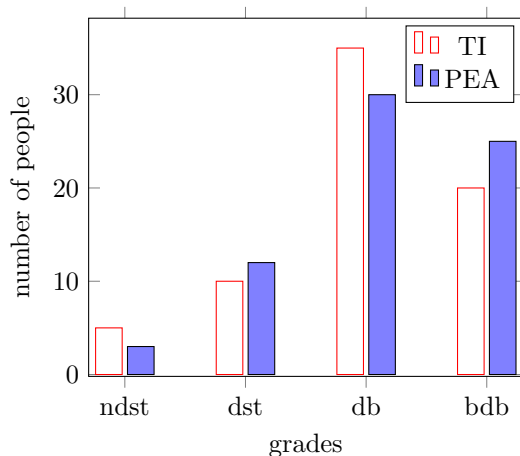
```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=0.8\textwidth,
4     xlabel=$x$,
5     ylabel=$y$,
6     legend pos = outer north east]
7   \addplot[color=red,mark=x]
8     coordinates {
9       (1,-1) (2,-2) (3,-3) (4,-4) (5,-4)
10      (6,-5) (7,-7) (8,-9) (9,-10) (10,-10)
11    };
12   \addplot[color=blue,mark=x, smooth]
13     coordinates {
14       (1,0) (2,-1) (3,-2) (4,-3) (5,-3)
15       (6,-4) (7,-6) (8,-8) (9,-9) (10,-9)
16    };
17   \addplot[color=purple,mark=x, smooth]
18     coordinates {
19       (1,-4) (5,-9) (8,-10)
20    };
21   \legend{f(x),g(x)};
22 \end{axis}
23 \end{tikzpicture}

```

1.3 Wykres słupkowy

Wykres słupkowy budujemy bardzo podobnie jak zwykły wykres, ponieważ "czubki" słupków to tak naprawdę współrzędnymi punktów do narysowania funkcji. Dzięki użyciu opcji `ybar` lub `xbar` system \LaTeX wie, że współrzędne ma traktować jako słupki pionowe lub poziome. W przypadku wykresy słupkowego zmiana opcji `color` zmieni nam kolor jedynie obramowania słupka. Jeśli zmienić kolor samego słupka należy użyć opcji do wypełnienia go kolorem `fill`. Opcja `fill` działa również dla zwykłych wykresów funkcji i wypełni pole wewnątrz funkcji. Jeśli dziedzina składa się z wartości tekstowych należy wskazać elementy i kolejność wchodzące w jej skład symbolic `x coords={ndst, dst, db, bdb}`.



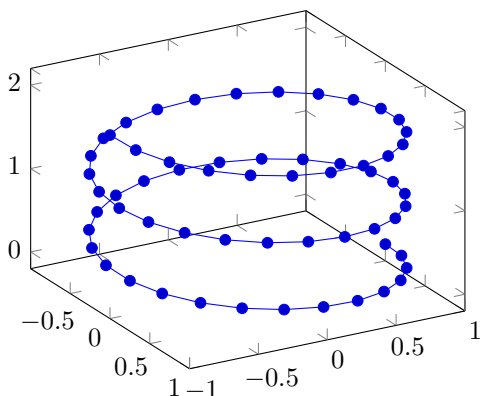
```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     ybar,
5     symbolic x coords={ndst, dst, db, bdb},
6     ylabel={number of people},
7     xlabel={grades}
8   ]
9   \addplot[color = red] coordinates {
10     (ndst,5) (dst,10) (db,35) (bdb,20)
11   };
12   \addplot[fill=blue!50] coordinates {
13     (ndst,3) (dst,12) (db,30) (bdb,25)
14   };
15   \legend{TI, PEA};
16 \end{axis}
17 \end{tikzpicture}

```

1.4 Podpisywanie wykresów

Wykresy podpisujemy tak identycznie jak inne obrazki przy użyciu otoczenia `figure`. Wykres tak samo jak inne rysunki należy wyśrodkować `\centering`, następnie podpisać pod rysunkiem `\caption{podpis}` oraz dołożyć etykietę `\label{fig:etykieta}`.



Rysunek 1: Wykres 3D

```

1 \begin{figure}
2   \centering
3   \begin{tikzpicture}
4     \begin{axis}[
5       width=\textwidth,
6       view={60}{30}
7     ]
8       \addplot3+[
9         domain=0:5*pi,
10        samples=60,
11        samples y=0
12      ](
13        {sin(deg(x))},
14        {cos(deg(x))},
15        {2*x/(5*pi)}
16      );
17     \end{axis}
18   \end{tikzpicture}
19   \caption{Wykres 3D}%
20   \label{fig:3dchart}
21 \end{figure}

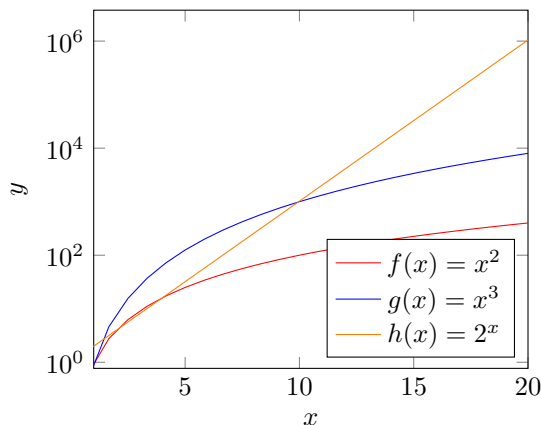
```

1.5 Skala logarytmiczna

W przypadku analizy wykresu funkcji, której wartości mają bardzo duże różnice, warto wtedy zastosować skalę logarytmiczną. Skalę logarytmiczną stosujemy dla wartości dodatnich. Aby wyrysować wykres w skali logarytmicznej zamiast otoczenia `axis` należy użyć któregoś z poniższych otoczeń:

`loglogaxis` obie osie w skali logarytmicznej
`semilogxaxis` jedynie oś OX w skali logarytmicznej
`semilogyaxis` jedynie oś OY w skali logarytmicznej

Jeśli chcemy dostosować zasięg rysowania funkcji warto w poleceniu `addplot` określić dziedzinę funkcji `domain=zakres` podając zakres liczb przedzielony symbolem dwukropka ":", ponieważ automatycznie dobrana wartość nie zawsze może nam odpowiadać.



```

1 \begin{tikzpicture}
2   \begin{semilogyaxis}[
3     width=\textwidth,
4     xlabel=$x$,
5     ylabel=$y$,
6     ymin=0,
7     xmin=1,
8     xmax=20,
9     legend pos = south east
10  ]
11  \addplot[red, domain=0:20] {x^2};
12  \addplot[blue, domain=0:20] {x^3};
13  \addplot[orange, domain=0:20] {2^x};
14  \legend{$f(x)=x^2$, $g(x)=x^3$, $h(x)=2^x$}
15  \end{semilogyaxis}
16 \end{tikzpicture}

```

2 Zaawansowane formatowanie wykresów

2.1 Wykresy z pliku

Typowym przykładem wykorzystania wykresów w dokumencie jest prezentacja danych eksperymentalnych. Niezależnie od sposobu ich przechowywania, prawie zawsze da się je wyeksportować do formatu `csv`. Ten z kolei może posłużyć do narysowania wykresu. Format `csv` jest dość luźno zdefiniowany, stąd podane niżej informacje dotyczą jego obsługi w paczce `pgfplots`. Plik `csv` przypomina tabelę, której kolumny rozdzielone są *separatorem*, a wiersze znakiem nowej linii. Plik może — ale nie musi — zawierać nagłówek, czyli specjalnie interpretowany pierwszy wiersz. Domyślnym separatorem dla `pgfplots` jest jeden lub więcej znaków białych. Zawartość przykładowego pliku `data.csv` pokazano poniżej:

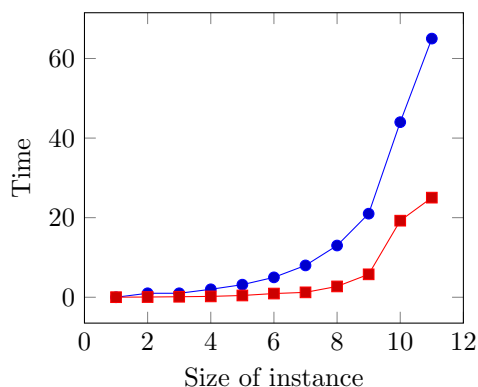
lp	name	tn	ti	ts	prdi	prds
1	20x100	0	0.0186	0.0177	-2.74	-2.19
2	20x10	1	0.0567	0.0567	-2.64	-2.66
3	20x20	1	0.1306	0.1285	-3.83	-3.14
4	50x5	2	0.2092	0.2156	-1.56	-1.69
5	50x10	3.1630	0.4528	0.4459	-3.25	-3.86
6	50x20	5	0.9229	0.9302	-3.48	-3.82
7	100x5	8	1.2632	1.2317	-0.88	-0.99
8	100x10	13	2.7882	2.7226	-1.26	-1.37
9	100x20	21	5.7525	5.7632	-2.29	-2.73
10	200x10	44	19.158	19.243	-0.84	-1.02
11	20x20	65	1.346	25.033	-1.50	-1.73

Plik składa się z nagłówka, 11 wierszy z danymi i 7 kolumn. Dane liczbowe wykorzystują kropkę jako separator dziesiętny. Jest to bardzo istotne, ponieważ **w systemie L^AT_EX separatorem dziesiętnym jest kropka, a nie przecinek**.

Aby wczytać dane z pliku i wykorzystać je do narysowania wykresu, należy skorzystać z polecenia `table[argumenty]{ścieżka_pliku}`. Jego argumenty definiują sposób w jaki dane są przetwarzane i interpretowane. Do podstawowych kluczy należą:

- x nazwa kolumny interpretowanej jako współrzędna x punktów wykresu.
- y nazwa kolumny interpretowanej jako współrzędna y punktów wykresu.
- z nazwa kolumny interpretowanej jako współrzędna z punktów wykresu.
- col sep separator użyty w pliku z danymi, domyślnie jeden lub więcej znaków białych.

Przykład użycia polecenia pokazano poniżej:



```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=0.9\textwidth,
4     xlabel=Size of instance,
5     ylabel=Time
6   ]
7     \addplot table[x=lp, y=tn] {./data.csv};
8     \addplot table[x=lp, y=ts] {./data.csv};
9   \end{axis}
10 \end{tikzpicture}

```

Nie należy rysować wykresów składających się z więcej niż kilkunastu–kilkudziesięciu tysięcy punktów. Jeżeli mimo to chcemy próbować szczęścia, w razie problemów z pamięcią można spróbować zmienić kompilator na `xelatex`, kompilować plik lokalnie, usunąć część punktów, lub po prostu skorzystać z innych metod wizualizacji.

2.2 Niepewności pomiarowe

Na przedmiotach takich jak miernictwo, czy fizyka, pojawi się konieczność rysowania wykresów ze słupkami błędów. O ile samo zagadnienie niepewności pomiarowych pozostaje poza zakresem przedmiotu „Technologie Informacyjne”, tak ich wizualizacja — jak najbardziej.

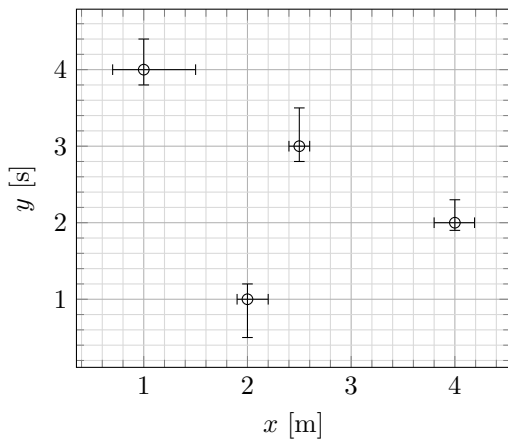
Założmy, że dokonano pomiaru wartości x [m] i y [s], a ich niepewność pomiarowa jest opisana przedziałami $(x - x_l, x + x_r)$ i $(y - y_l, y + y_r)$. Poniżej pokazano przykład pliku `errors.csv` zawierającego 4 punkty pomiarowe:

```

x; x_l; x_r; y; y_l; y_r
1; 0.3; 0.5; 4; 0.2; 0.4
2.5; 0.1; 0.1; 3; 0.2; 0.5
4; 0.2; 0.19; 2; 0.1; 0.3
2; 0.1; 0.2; 1; 0.5; 0.2

```

Jego wizualizację pokazano w następującym przykładzie:



```

1 \begin{tikzpicture}
2   \begin{axis}[
3     width=\textwidth,
4     xlabel={x$ [m]},
5     ylabel={y$ [s]},
6     grid=both,
7     minor tick num=4,
8     minor grid style={black!15},
9     major grid style={black!30},
10    xtick distance=1,
11    ytick distance=1
12  ]
13  \addplot[
14    only marks, mark=o,
15    error bars/.cd,
16    x dir=both, x explicit,
17    y dir=both, y explicit
18  ]
19  table[
20    x=x, x error minus=xl, x error plus=xr,
21    y=y, y error minus=yl, y error plus=yr,
22    col sep=semicolon
23  ]{errors.csv};
24 \end{axis}
25 \end{tikzpicture}

```

W argumencie opcjonalnym makra `addplot` pojawiły się nowe klucze:

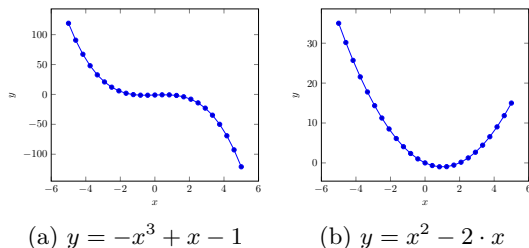
- `error bars/.cd` argument upraszczający wpisywanie kolejnych argumentów.
- `x dir` w którą stronę skierowane są słupki błędów dla osi x , `both` oznacza w *obie* strony.
- `x explicit` długości słupków błędów dla osi x zadawane są wartością bezwzględną.

Oczywiście, należy użyć również nowych kluczy w `table`:

- `x error minus` wartość x_l , czyli długość „ujemnego” słupka błędów osi x .
- `x error plus` wartość x_r , czyli długość „dodatniego” słupka błędów osi x .

3 Złożone obrazy

W celu umieszczenia dwóch lub więcej wykresów wchodzący w skład jednego rysunku. W tym celu należy skorzystać z otoczenia `subfigure` z pakietu `subcaption`. Całość będzie widoczna jako jeden rysunek, dlatego cały kod należy umieścić wewnątrz jednego otoczenia `figure`. Następnie każdy wykres należy umieścić osobno w otoczeniu `subfigure{szerokość}`, które jako argument przyjmuje szerokość dla pojedynczego rysunku. Jeśli rysunki się nie zmieszczą obok siebie to drugi przeskoczy niżej. Jednak jeśli chcemy żeby zmieściły się obok siebie warto posłużyć się szerokością bazującą na szerokości jaką dysponujemy `\linewidth`, warto też zostawić margines błędów. W celu poprawnego podpisania należy wywołać 3 razy `caption` osobno dla każdego „małego” obrazka wewnątrz otoczenia `subfigure` oraz dla całości `figure`.



(a) $y = -x^3 + x - 1$

(b) $y = x^2 - 2 \cdot x$

Rysunek 2: Dwa wykresy

```

1 \begin{figure}[h]
2   \centering\hfill
3   \begin{subfigure}{.48\linewidth}
4     \centering
5     \begin{tikzpicture}[scale=.4]
6       \begin{axis}[xlabel=$x$, ylabel=$y$]
7         \addplot{-x^3+x-1};
8       \end{axis}
9     \end{tikzpicture}
10    \caption{$y = -x^3+x-1$}
11  \end{subfigure}
12  \hfill
13  \begin{subfigure}{.48\linewidth}
14    \centering
15    \begin{tikzpicture}[scale=.4]
16      \begin{axis}[xlabel=$x$, ylabel={$y$}]
17        \addplot{x^2-2*x};
18      \end{axis}
19    \end{tikzpicture}
20    \caption{$y = x^2-2 \cdot x$}
21  \end{subfigure}
22  \hfill
23  \caption{Dwa wykresy}
24 \end{figure}

```