**10. Obliczeniowe jednostki wykonawcze CPU, GPU, TPU, FPU.**

**Procesor (ang. central processing unit, CPU)** – sekwencyjne urządzenie cyfrowe, które pobiera dane z pamięci operacyjnej, interpretuje je i wykonuje jako rozkazy.

Procesory wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złocone wyprowadzenia (stosowane ze względu na odporność na utlenianie) i w takiej postaci nazywa się je mikroprocesorami – w mowie potocznej pojęcia procesor i mikroprocesor używane są zamiennie. Sercem procesora jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilku miliardów tranzystorów. Jego obwody wykonywane są z metali o dobrym przewodnictwie elektrycznym, takich jak aluminium czy miedź.

Jedną z podstawowych cech procesora jest określona długość (liczba bitów) słowa, na którym wykonuje on podstawowe operacje obliczeniowe. Jeśli przykładowo słowo tworzą 64 bity, to taki procesor określany jest jako 64-bitowy. Innym ważnym parametrem określającym procesor jest szybkość, z jaką wykonuje on rozkazy. Przy danej architekturze procesora, szybkość ta w znacznym stopniu zależy od czasu trwania pojedynczego taktu[1], a więc głównie od częstotliwości jego taktowania.

**Procesor graficzny (ang. graphics processing unit, GPU)** – jednostka obliczeniowa znajdująca się w kartach graficznych.

**Tensor Processing Unit (TPU)** to specyficzny dla aplikacji układ scalony (ASIC) akceleratora AI opracowany przez Google do uczenia maszynowego sieci neuronowych przy użyciu własnego oprogramowania TensorFlow firmy Google.

**Koprocesor arytmetyczny, jednostka zmiennoprzecinkowa (ang. Floating-Point Unit, FPU)** – układ scalony wspomagający procesor w obliczeniach głównie zmiennoprzecinkowych, ale również na liczbach całkowitych. W większości współczesnych konstrukcji koprocesor arytmetyczny, a także jednostki obsługujące bardziej skomplikowane obliczenia (np. instrukcje wektorowe), zintegrowany jest z procesorem w jednym układzie scalonym. Koprocesorami nazywane bywają również układy wspomagające tworzenie i przetwarzanie grafiki (głównie wektorowej), czyli procesory graficzne (GPU). Ponadto nazwa koprocesor czasami używana jest w stosunku do układów przetwarzających sygnały (DSP) i procesorów dźwiękowych pozwalających pozycjonować dźwięki w przestrzeni (karta Sound Blaster X-Fi).

**30. Biblioteki wspierające tworzenie wykresów za pomocą języka Python**

**Matplotlib** - biblioteka do tworzenia wykresów dla języka programowania Python i jego rozszerzenia numerycznego NumPy. Zawiera ona API „pylab” zaprojektowane tak aby było jak najbardziej podobne do MATLABa, przez co jest łatwy do nauczenia przez jego użytkowników. Matplotlib został napisany i jest utrzymywany głównie przez Johna Huntera, i jest dostępny na licencji przypominającej licencję BSD.

**Najczęściej używane komendy w Matplotlib:**

plt.figure(): stwórz nowy wykres

plt.plot(): narysuj x i y jako linie i/lub markery

plt.xlabel(): oznacz oś x

plt.ylabel(): oznacz oś y

plt.title(): Dodaj tytuł dla swoich osi

plt.grid(): Skonfiguruj linie siatki

plt.legend(): Umieść odnośniki do legendy na swoich osiach

plt.savefig(): Zapisz wykres na dysku

plt.show(): Wyświetl wykres

plt.clf(): Usuń wykres (może się przydać do narysowania kilku wykresów w tym samym kodzie)

**Numpy** - Biblioteka NumPy, w Python, została stworzona, aby umożliwić szybkie i sprawne operacje na macierzach. Każdy element jest tego samego typu – zazwyczaj są to liczby. Na jej podstawie stworzono, między innymi bibliotekę Pandas.

np.array(), w której podajemy wartości,

np.arange(), w której podajemy zakres jakim, chcemy uzupełnić tablice,oraz odstęp liczbowy między nimi,

ndarray – podstawowy typ danych

**Pandas** - Możemy za jego pomocą, wczytywać dane, czyścić, modyfikować, a nawet analizować. Wszystko to co umożliwia nam SQL, Excel i dużo więcej.

Pierwszy typ danych to 'Series’. Dla analogii, możemy porównać ją do kolumny z Excela. Działa ona podobnie do listy w Python, jednak daje nam większe możliwości.

Drugi typ danych, to DataFrame. O ile 'Series’ porównywaliśmy do kolumny, o tyle DataFrame, jest odpowiednikiem tabeli, czyli zestawieniem danych typu 'Series’.

**50. Własności funkcji: miejsca zerowe, ciągłość, pochodna.**

**Miejsce zerowe** - to taki argument x dla którego funkcja przyjmuje wartość 0.

**Definicja: Funkcja** jest ciągła jeśli jest ciągła w każdym punkcie swojej dziedziny, czyli w zbiorze A. Funkcja jest ciągła w zbiorze C zawartym w jej dziedzinie jeśli jest ciągła w każdym punkcie tego zbioru.

**Definicja Cauchy’ego**

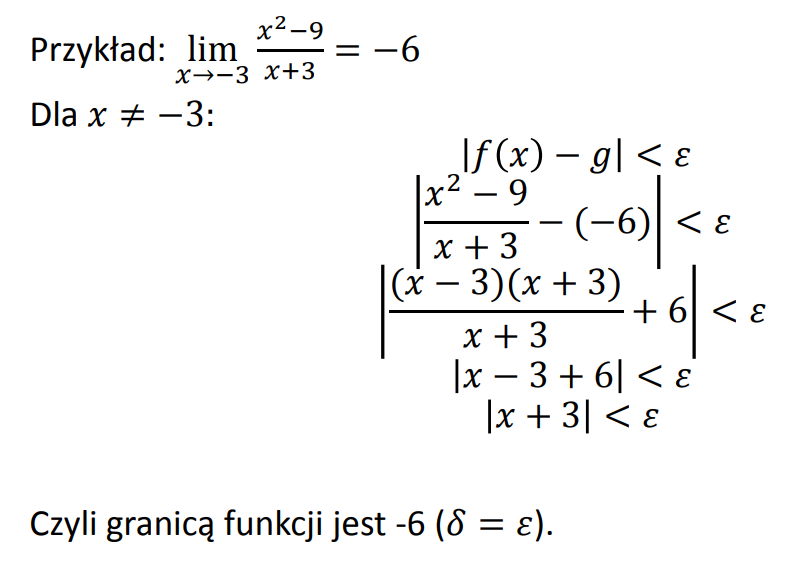
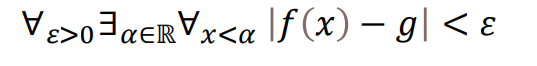
Mówimy, że funkcja 𝑓 posiada w punkcie 𝑎 granicę 𝑔, jeżeli spełniony jest warunek



Definicja dla granica w +∞:

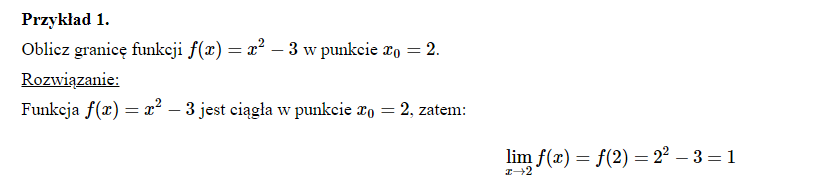


lub odpowiednio dla −∞:



**Definicja Heinego:**

Funkcja f(x) ma granicę g w punkcie x0, jeśli dla każdego ciągu (xn) zbieżnego do x0, ciąg (f(xn)) jest zbieżny do g. Zapis matematyczny:

**Ciągłość jednostronna**

Rozpatruje się czasami funkcje ciągłe jednostronnie: lewo- i prawostronne. Dla definicji Cauchy’ego należy dodać warunek dla x, mianowicie x<x0 aby otrzymać funkcję ciągłą lewostronnie. Definicja funkcji ciągłej prawostronnie wymaga zmiany powyższej nierówności na przeciwną. Definicja Heinego wymaga wybrania dowolnego ciągu zbliżającego się do x0 wyłącznie punktami z lewej lub prawej strony.

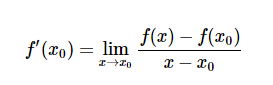
**Przykłady funkcji ciągłych:**

Funkcje: Trygonometryczne, Cyklometryczne, Wielomianowa, Potęgowa, Logarytmiczna, Wymierna,

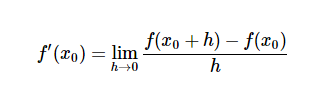
**Definicja pochodnej funkcji:**

nieformalnie: miara szybkości funkcji, czyli tempa zmian jej wartości względem zmian jej argumentów. Dokładna definicja pochodnej zależy od kontekstu, ponieważ pojęcie to stosuje się do funkcji różnego typu; jednak w każdym z tych przypadków pochodna to granica ilorazu różnicowego dla zerowego mianownika.

Wzory na pochodne:



iloraz różnicowy:

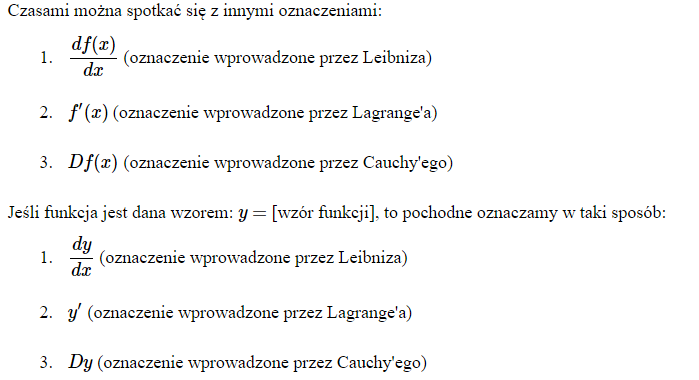


Pochodna pokazuje nam jak funkcja zmienia się w danym punkcie. Dokładniej:

* Jeśli f′(x0)>0, to funkcja f(x) **rośnie** w punkcie x0.
* Jeśli f′(x0)=0, to funkcja f(x) jest **stała** w punkcie x0.
* Jeśli f′(x0)<0, to funkcja f(x) **maleje** w punkcie x0.

Sposoby oznaczania pochodnych:

Pochodną funkcji f(x) zapisujemy najczęściej tak:



**70. Rzutowanie w grafice 3W.**

**Rzuty:**

♦ rzuty przekształcają punkty w n-wymiarowym układzie współrzędnych w punkty w układzie współrzędnych o wymiarze mniejszym niż n

♦ rzut obiektu 3D jest określony przez promienie rzutujące wychodzące ze środka rzutowania, przechodzące przez każdy punkt obiektu i przecinające płaszczyznę rzutowania

**Dowolny rzut 3D**

Rzutnia – inaczej płaszczyzna rzutowania,

Płaszczyznę rzutowania określa:

♦punkt na tej płaszczyźnie – tzw. punkt odniesienia rzutni (VRP),

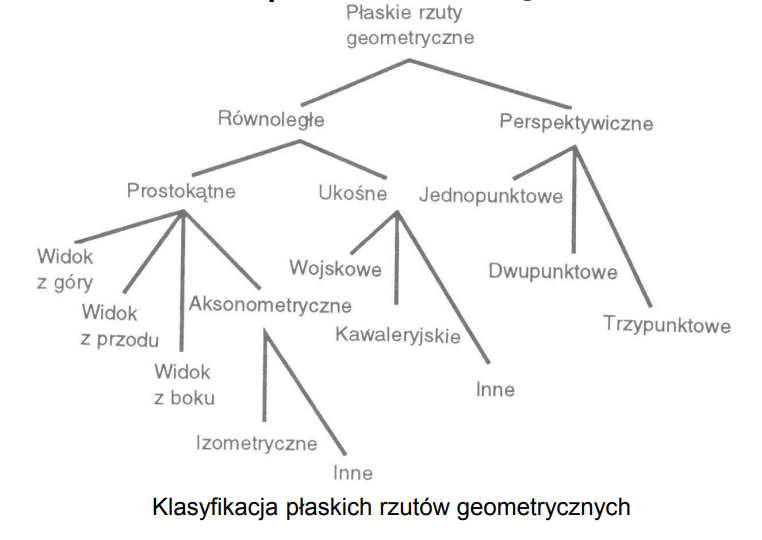
♦normalna do płaszczyzny – tzw. Normalna do rzutni (VPN),

W celu wyznaczenia okna dla rzutni należy określić:

♦ minimalną i maksymalną współrzędną okna,

♦ układ współrzędnych rzutowania (VRC), dwie osie na rzutni – jedna do niej prostopadła,

♦ punkt (VRP) będący początkiem układu (VRC)



**Rzuty perspektywiczne:**

♦rzuty perspektywiczne dowolnego zbioru linii równoległych, które nie są równoległe do rzutni, zbiegają się w punkcie zbieżności.

♦jeżeli rozważany zbiór linii jest równoległy do jednej z trzech osi, to punkt, w którym się zbiegają jest określany jako osiowy punkt zbieżności.

♦rzuty perspektywiczne są dzielone ze względu na liczbę osiowych punktów zbieżności.

**Rzuty równoległe**

♦rozróżniamy rzuty: prostokątny, skośny

♦najbardziej typowe rzuty ortogonalne: przedni, górny, boczny

**Rzuty równoległe**

♦w aksonometrycznych rzutach prostokątnych rzutnia nie jest prostopadła do głównej osi co umożliwia obserwację kilku stron obiektu.

♦często stosowanym rzutem aksonometrycznym jest rzut izometryczny. Normalna do rzutni tworzy równe kąty z głównymi osiami