Programátorská dokumentace k programu Maticové operace

Program vyžaduje kompilátor v souladu s normou c++11.

Zdrojový kód je umístěn ve třech souborech:

1. main.cpp

Obsahuje hlavní funkci *main* tato funkce. Ve funkci main se zpracovávají parametry programu.

Jediný možný parametr programu je soubor, ve kterém je uložena matice. Soubor musí být textový, řádky matice odděleny novým řádkem a čísla od sebe oddělena jakýmkoliv bílým znakem, např. mezera nebo tabulátor.

Po načtení hodnot se vypisuje načtená matice pomocí funkce *print*.

Z hodnot matice je vypočítán determinant. Matice musí mít velikost 2x2 nebo 3x3, aby šel determinant vypočítat. Pokud lze determinant vypočítat, je do konzole vypsána jeho hodnota.

Poté je vypočítána matice Gaussovou eliminační metodou. Aby matice šla vypočítat, musí mít o jeden sloupec více jak počet řádků. Přestože je při výpočtu několikanásobně ošetřeno dělení nulou, může se stát, že matice která obsahuje velký počet nul, nepůjde vypočítat. V tom případě je uživatel upozorněn na standartní výstup, že matici nelze vypočíst. Pokud matici vypočítat lze, je vypsán pomocí funkce *print* trojúhelníkový stav matice. Dále jsou vypsány hodnoty matice pojmenovány jako x_1 až $x_{pocet\text{-}radku}$

Podle vstupu uživatele, zda chce uložit výsledky se uloží vypočtená matice pomocí funkce *saveToFile*

Dále program jen uvolní veškerou paměť, ve které je matice uložena a ukončí se.

2. functions.h

V tomto hlavičkovém souboru jsou uloženy hlavičky všechny další funkce, které funkce *main* používá.

Také se zde vkládají knihovny, které program potřebuje k funkci. Jsou to knihovny *iostream, fstream, regex, cctype, cmath* a *algorithm.*

3. functions.cpp

Obsahuje všechny funkce, které funkce *main* používá.

Jejich kompletní seznam najdete v dokumentaci vygenerovanou přes Doxygen.

Tady popíšu, jak funguje výpočet gaussovy eliminační metody.

Před samotným výpočtem se spouští funkce *solve_zero_problem*, která zpřehází řádky tak, aby se nevyskytovala 0 na hlavní diagonále.

Procházejí se všechny řádky a s každým řádkem se procházejí řádky v další smyčce. Pro každý řádek se tedy projdou všechny následující řádky.

Z leva doprava se vypočítává konstant spodní řádek/horní a touto konstantou se vynásobí všechny hodnoty v aktuálním řádku. Spodní řádek je vnitřní smyčky. Horní řádek je vnější. Po vynásobení máme v matici nuly pod hlavní diagonálou.

Pokud při výpočtu někde vyjde, že by se muselo dělit nulou, je opět spuštěna funkce solve_zero_problem, která zpřehází následující řádky tak, aby se chyba s největší pravděpodobností neopakovala, zároveň se ukládá číslo řádku kde byl problém, kdyby se problém na tomto řádku opakoval, funkce výpočtu matice se ukončí s chybovým stavem, protože matici nelze vypočítat.

Máme tedy 0 pod hlavní diagonálou. V dalším kroku postupujeme odspoda nahoru. V pravém sloupci matice budeme ukládat hodnoty. Pro každý řádek vypočítáme sumu levé strany. Každé číslo musí být vynásobeno s vypočtenou hodnotou která mu odpovídá. Poté se vypočítá hodnota pro daný řádek tím, že hodnotu pravého sloupce vydělíme vypočtenou sumou. Takto postupujeme pro všechny řádky.

V pravém sloupci nyní máme výsledky matice.

Zpětně lze vygenerovat matici v trojúhelníkovém tvaru tím, že přepisujeme levou část matice aby obsahovala na hlavní diagonále 1 a ve zbytku 0.

Funkce *solve_zero_problem* funguje v principu takto: Najde řádky, kde je 0 na hlavní diagonále a zkusí k nim najít další řádek, který na této pozici má nenulovou hodnotu, zároveň ověřuje že v pozici hlavní diagonály prohledávaného řádku není 0. Pokud takový řádek najde, vymění jej za sebe.

Takto projde všechny řádky.

Vlivem výpočtů ve funkci pro gaussovu eliminační metodu se stává, že se řádky násobí a sčítají a tím mohou vznikat další 0 i na hlavní diagonále. Proto se funkce na opravu spouští při každém nálezu 0 na hlavní diagonále. Ve vzorci by se totiž muselo dělit nulou, což program vyhodnotí jako nekonečno a poté by roznásobil zbytek matice. S touto chybou by pokračoval na zbytek řádků.