Řešení soustav lineárních rovnic Gaussovou eliminační metodou dokumentace

Alexander Mansurov, 6.C

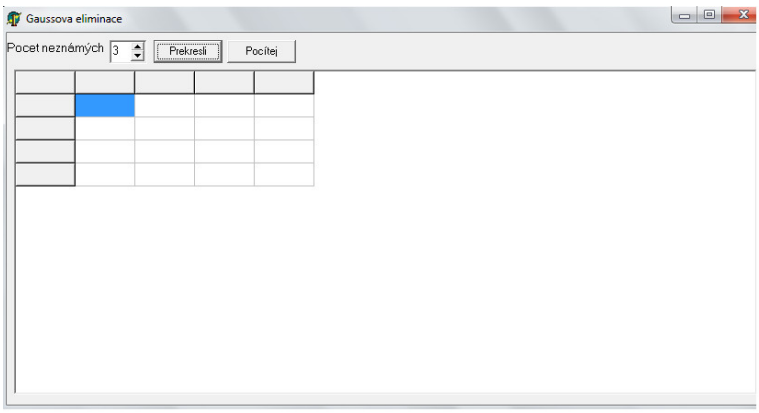
October 4, 2012

0.1 Grafické rozhraní

Popis Grafické rozhraní se skládá ze dvou komponent typu Label, jedné komponenty typu SpinEdit, dvou tlačítek, a jednoho StringGridu.

Jedna komponenta typu Label není z počátku vidět a slouží pro zobrazení řešení soustavy a druhá konstantně informuje uživatele o tom, že vedle se nacházející komponentou SpinEdit nastavujeme počet neznámých soustavy. Jedno z tlačítek slouží pro změnu počtu neznámých, druhé pro zahájení výpočtu. Samotná matice se zadává pomocí komponenty StringGrid.

Postup Nejprve pomocí SpinEditu vybereme počet neznámých a tlačítkem Prekresli nastavíme rozměry StringGridu. StringGrid dostane popisky buněk a bude do něj povoleno zapisovat. Uživatel následně zadává koeficienty rovnic a jejich pravou stranu. Nakonec odstartuje výpočet příslušným tlačítkem a dostane výsledek v příslušném Labelu.



0.2 Pohled pod pokličku

TForm1.Button2Click (řádek 137) Když uživatel odstartuje výpočet, nejprve dojde k zapsání vstupních dat do vnitřních datových struktur. V rámci programu se používají dva vlastní datové typy: Matrix a Vector. Matrix je dvojrozměrné pole reálných čísel velikosti 10x10, Vector jednorozměrné pole reálných čísel velikosti 10. Pole jsou indexovaná od 0 do 9. Koeficienty matice soustavy se ukládají po sloupcích do proměnné matice typu Matrix. Koeficienty pravé strany se ukládají do proměnné vektor typu Vector. Tyto struktury spolu s číslem n označujícím počet neznámých se předají proceduře gauss, provádějící vlastní eliminaci.

gauss (řádek 107) Gaussova eliminace s výběrem pivota. Procházíme matici po řádcích. Pro každý řádek nalezneme pivota - největší prvek ve sloupci pod hlavní diagonálou, tj. pro i-tý řádek, pod i-tým sloupcem - užitím funkce pivot_lookup. Pokud pivot není v matici na pozici [i,i], pomocí procedury swap dojde k prohození řádků v matici tak, aby se pivot posléze nacházel na pozici [i,i].

Procházíme i-tý sloupec od pozice [i,i] dolů a v každém řádku, za podmínky, že pivot není nula, procházíme od i-tého sloupce doprava a od jednotlivých prvků matice i vektoru odečítáme podíl prvku v daném řádku pod pivotem ku pivotu násobený prvkem matice ve stejném sloupci z i-tého řádku. Zároveň od vektoru (pravé strany rovnice) odečteme podíl prvku v daném řádku pod pivotem ku

V případě, že by pivot byl nula, nastane výjimka o nedovoleném dělení nulou a zároveň už z funkce pivot_lookup obdržíme hlášku o nejednoznačnosti řešení. Pokud k chybě nedojde, pustíme se do reverzní substituce pomocí procedury rev_sbst.

pivot_lookup (řádek 44) Funkce hledá největší prvek v i-tém sloupci, vrací číslo řádku tohoto prvku nebo nastává výjimka pokud by tento pivot byl rovný nule, jelikož by řešení soustavy nebylo jednoznačné.

swap (řádek 76) Procedura prohodí v i-tém sloupci matice k-tý prvek s prvkem označeným jako pivot a zároveň prohodí i-tý prvek vektoru s prvkem označeným jako pivot.

rev_sbst (řádek 113) Procedura koná zpětný chod po kterém je ve vektoru řešení soustavy, pokud existuje, nebo nastane výjimka o neexistenci řešení soustavy.

Pokud v pravém dolním rohu matice je nula a zároveň je poslední prvek vektoru nula, pak se hodí výjimka, že soustava má nekonečně mnoho řešení, pokud však poslední prvek vektoru není nula, nastane výjimka, že soustava nemá řešení.

Do i-té rovnice dosazujeme vypočítané hodnoty x_i až x_n, vyjádříme a vypočítáme neznámou x_i.

0.2.1 Další pomocné procedury

TForm1.Button1Click (řádek 65) Nastaví velikost StringGridu na $n*(n+1)$, kde n je počet rovnic zadaný uživatelem přes komponentu SpinEdit.

outprint (řádek 149) Vypíše množinu řešení na obrazovku do Labelu2

popis_stringgrid (řádek 152) Přidá popisky na kompnentu StringGrid pro uživatelskou přívětivost