

INFORMATIKA

Pascalův trojúhelník

REDAKCE



Sestavení Pascalova trojúhelníku patří mezi vděčné úlohy v počáteční fázi výuky programování. Úkolem je sestavit (tj. vytisknout) tento „trojúhelník“ pro zadané přirozené číslo N . Např. pro $N = 6$ dostaneme:

				1						
				1		1				
			1		2		1			
		1		3		3		1		
	1		4		6		4		1	
1		1	5		10		10		5	1
	1	6		15		20		15	6	1

Při řešení této úlohy samozřejmě nejde o trojúhelníkový tvar výsledku, ale o výpočet čísel v jednotlivých řádcích. V n -tém řádku (počínaje nulým) tato čísla jsou, jak víme, $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}$. Dále si ukážeme přístup k řešení zadané úlohy u tří středoškolských studentů.

První z nich vyšel právě z té skutečnosti, že stačí počítat kombinační čísla $\binom{n}{k}$. Použil proměnné typu Real a až při tisku se vrátil do celočíselného typu Longint. Pro výpočet kombinačního čísla definoval funkci *KombC*.

Zadáva se N jako počet řádků (program funguje pro N od 1 do 34, tedy maximální Pascalův trojúhelník má 34 řádků a je vytvořen pro n od 0 do 33; na další řádek už rozsah datového typu Longint nestačí).

Poznamenejme ještě, že by bylo vhodnější vůbec nevnášet do řešení tohoto ryze celočíselného problému datový typ Real a provádět celý výpočet přímo v proměnných typu Longint, byť v takovém případě vzhledem ke střídajícímu se násobení a dělení proměnné KC můžeme program použít pouze do hodnoty $N = 30$. Za poznámku dále stojí, že proměnná $Q1$, podmínka $Q2 < Q1$ i větve else uvedená uvnitř for-cyklu jsou naprosto zbytečné.

```
program PasTrojA;
var
  I, J, N: Integer;

function KombC(N, K: Integer): Real;
var
  M: Integer;
  Q1, Q2, KC: Real;
begin
  Q1 := N - K;
  Q2 := 0;
  if N > K then
    begin
      KC := 1;
      for M := K + 1 to N do
        if Q2 < Q1 then
          begin
            Q2 := Q2 + 1;
            KC := KC * M / Q2
          end
        else
          KC := KC * M;
        end
      end
      KombC := KC
    end
  else
    KombC := 1
  end;
end;
```

```

begin
  Write('Pascaluv trojuhelnik; pocet radku: ');
  ReadLn(N);
  for I := 0 to N - 1 do
    begin
      for J := 0 to I do
        Write(Trunc(KombC(I, J)), ' ');
      WriteLn
    end;
  ReadLn
end.

```

Druhý řešitel využil známé vlastnosti kombinačních čísel: $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$ a tvořil následující řádek sečítáním dvojic prvků v řádku předchozím. Přitom ve dvourozměrném poli A uchovává všechny vypočtené hodnoty a nakonec toto pole A vytiskne. Toto řešení je vcelku elegantní a výrazně rychlejší než předchozí, vytknout bychom mu mohli zbytečně velké paměťové nároky a také zbytečné nulování všech prvků pole A na začátku výpočtu.

```

program PasTrojB;
var
  I, J, N: Integer;
  A: array [0..40, 0..40] of Longint;

begin
  Write('Pascaluv trojuhelnik; pocet radku: ');
  ReadLn(N);
  for I := 0 to N do
    for J := 0 to N do
      A[I, J] := 0;
  A[0,0] := 1;
  for I := 1 to N do
    for J := 1 to I do
      A[I, J] := A[I - 1, J - 1] + A[I - 1, J];
  for I := 1 to N do
    begin

```

```

    for J := 1 to I do
        Write(A[I, J], ' ');
    WriteLn
end;
ReadLn
end.

```

Třetí řešitel přibral do řešení souměrnost Pascalova trojúhelníku: $\binom{n}{k} = \binom{n}{n-k}$. Narozdíl od předchozího řešení pracuje jen se dvěma řádky (jednorozměrnými poli) *StaryR* a *NovyR* a údaje nacházející se v *NovyR* se ihned tisknou, přičemž druhá polovina řádku se tiskne v opačném pořadí vzhledem k 1. polovině. Musí se ovšem rozlišovat, jestli je v řádku sudý nebo lichý počet čísel. Po tisku přesune program pole *NovyR* do pole *StaryR* a provádí se výpočet první poloviny nového řádku touž metodou jako u předchozího programu.

```

program PasTrojC;
var
    I, J, K, N: Integer;
    StaryR, NovyR: array [0..40] of Longint;
    SudyPocet: Boolean;

begin
    Write('Pascaluv trojuhelnik; pocet radku: ');
    ReadLn(N);
    for I := 0 to N do
        NovyR[I] := 0;
    K := 1;
    SudyPocet := false;
    NovyR[1] := 1;
    for I := 1 to N do
        begin
            for J := 1 to K do
                Write(NovyR[J], ' ');
            for J := K + Ord(SudyPocet) - 1 downto 1 do
                Write(NovyR[J], ' ');
            WriteLn;
            if SudyPocet then

```

```

    K := K + 1;
    SudyPocet := not SudyPocet;
    StaryR := NovyR;
    for J := 1 to K do
        NovyR[J] := StaryR[J - 1] + StaryR[J];
    if not SudyPocet then
        NovyR[J] := NovyR[J] + StaryR[J - 1]
    end;
    ReadLn
end.

```

V následujícím redakčním řešení je za N volen nikoli počet řádků, ale n posledního řádku, což je přirozenější, a volba je pojištěna, aby nedošlo k zadání N mimo přípustný rozsah (předchozí programy v tomto případě tak či onak havarují). Výpočet řádků se provádí stejnou metodou jako v programu *PasTrojB*, ale s tím, že vypočtené hodnoty se ihned tisknou (jako v programu *PasTrojC*). Tím se vzhledem k programu *PasTrojB* velmi sníží paměťová náročnost. Na rozdíl od *PasTrojC* se však počítají celé řádky, protože organizace zpětného tisku 2. poloviny řádku a hlídání sudosti a lichosti počtu členů zbytečně komplikují program, aniž by vznikla jakákoli zjištěitelná úspora. Pracuje se také se dvěma řádky jako v programu *PasTrojC*, ale každý z těchto řádků je střídavě „starý“ a „nový“, takže se ušetří přesouvání nového řádku do starého.

```

program PasTrojR;
const
    MaxN = 33;
var
    I, J, K, N: Integer;
    KC: array [0..1, 0..MaxN+1] of Longint;

begin
    Write('Pascaluv trojuhelnik; radek 0 az: ');
    repeat
        ReadLn(N)
    until N in [1..MaxN];
    for I := 0 to 1 do
        for J := 0 to N + 1 do

```

```

        KC[I, J] := 0;
    KC[0,1] := 1;
    K := 0;
    for I := 1 to N + 1 do
    begin
        for J := 1 to I do
        begin
            KC[1 - K, J] := KC[K, J - 1] + KC[K, J];
            Write(KC[1 - K, J], ' ')
        end;
        WriteLn;
        K := 1 - K;
    end;
    ReadLn
end.

```

Předložené programy mohou být vhodným materiálem pro posouzení jejich předností a nedostatků. (Po stránce formálního zápisu zdrojového textu byly však redakčně upraveny, takže původní formální nedostatky už zde vidět nejsou.) Všimněme si, že v programech *PasTrojB* a *PasTrojC* byl proveden jen hrubý odhad rozsahu N (v *PasTrojA* prováděn nebyl), že v programech pracujících s celými čísly je tu časová složitost $O(N^2)$ a v *PasTrojA* je to $O(N^3)$, všimněme si i paměťové složitosti, zvláště u *PasTrojB*.

Studenti mohou řešit i různé úpravy zadání, např. požadavek tisku jen k posledních řádků Pascalova trojúhelníku nebo výpočet Pascalova trojúhelníku i pro větší N , kde by se místo členů přesahujících formát Longint tiskly např. hvězdičky.

(Autorkou ilustrace je Mgr. Jaroslava Čermáková z Hlinska v Čechách.)

Graf nespojité funkce s parametry v Excelu

VÁCLAV MATYÁŠ

Purkyňovo gymnázium, Strážnice

V článku [1] jsme probrali použití posuvníků při zobrazování funkcí s parametry. Nyní se podíváme na případy těch nespojitých funkcí, které mají v některém bodě nevlastní limitu nebo jednostranné nevlastní limity.

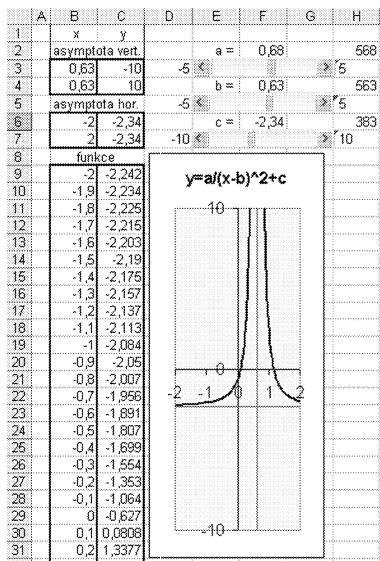
Funkce $y = \frac{a}{(x - b)^2} + c$

Vidíme, že tato funkce má bod nespojitosti b , není v něm definována a má v něm limitu $+\infty$ v případě, že $a > 0$, a limitu $-\infty$ v případě, že $a < 0$. Excel není nejvhodnější prostředek pro zobrazování neomezených (a tedy i nespojitých) funkcí. Jestliže totiž volíme bodový graf a podtyp „Bodový s datovými body spojenými pomocí hladkých spojnic a bez značek“, pak tím vlastně po Excelu požadujeme, aby všechny jemu předkládané funkce považoval za spojité.

V případě naší funkce, pokud hodnota b není v tabulce funkce, spojí Excel zadaný tabulkový bod $[x_1, y_1]$, kde $x_1 < b$ je hodnota nejbližší k b zleva, s tabulkovým bodem $[x_2, y_2]$, kde $x_2 > b$ je hodnota nejbližší k b zprava, takže graf je spojitý. V tomto případě je vhodné volit v obrázku rozsah funkčních hodnot tak, aby uvedené „nepovolené“ spojení vyšlo mimo obrázek. Ovšem vzhledem k tomu, že máme funkci s parametry, nepodaří se to pro všechny kombinace hodnot parametru a musíme být vůči Excelu trochu tolerantní. Ke zlepšení dojde, volíme-li tabulku nezávisle proměnné „hustší“, protože pak vyjdou body x_1 a x_2 velmi blízko b a onen spojovací efekt se ukáže jen velmi vzácně.

Jestliže se bod b vyskytuje v tabulce funkce jako jedna z hodnot nezávisle proměnné, pak Excel hodnotu funkce samozřejmě nevypočítá (nabídne nám „zahrádky“), ale graf vykreslí tak, jako by hodnota funkce byla nula, tedy velmi radikálně jej pozmění. I tento případ vyžaduje naši toleranci. Samozřejmě v jednotlivých případech lze zásahem do tabulky obrázek vylepšit, ale pracujeme-li s parametry, pak takový mimořádný zásah do tabulky ji může pro další práci znehodnotit.

Postupem popsaným v [1] lze získat graf naší funkce, ale tento graf nás neuspokojí z toho důvodu, že pro větší názornost průběhu funkce nám budou chybět asymptoty; zde je horizontální asymptota $y = c$ a vertikální asymptota $x = b$. Tyto asymptoty, protože jde o přímky, lze získat z dvouhodnotových tabulek (viz obr. 1) a to tak, že v jednom obrázku zobrazíme tři „bodové řady“: zadaná funkce nám vytváří řadu 1 a asymptoty řadu 2 a řadu 3 (tak je také zadáváme). Pro názornost je vhodné pro graf funkce volit poněkud tlustší čáru (poklepem levým tlačítkem myši na čáru grafu lze otevřít okno, v němž lze volit tloušťku i barvu grafu). Názornost zvýšíme, když pro asymptoty volíme jinou barvu.



Obr. 1

Takto zpracovaný graf lze použít při probírání učiva v části 5.5 [2] Racionální a polynommické funkce.

Funkce $y = \frac{ax + b}{cx + d}$

Tato funkce není definována v bodě $-\frac{d}{c}$ a pokud čítecel není násobkem jmenovatele, je přímka $x = -\frac{d}{c}$ vertikální asymptotou grafu funkce. Ježto

$$\lim_{x \rightarrow \pm\infty} \frac{ax+b}{cx+d} = \frac{a}{c},$$

je přímka $y = \frac{a}{c}$ horizontální asymptotou grafu této funkce. Zadanou funkci uveďme ještě na tvar, v němž lze názorněji rozpoznávat vlastnosti funkce. Upravujme:

$$\frac{ax+b}{cx+d} = \frac{a}{c} \cdot \frac{x + \frac{d}{c} + \frac{b}{a} - \frac{d}{c}}{x + \frac{d}{c}} = \frac{a}{c} + \frac{1}{x + \frac{d}{c}} \cdot \frac{a}{c} \cdot \frac{bc-ad}{ac} = y_p + \frac{k}{x-x_p}, \quad (1)$$

kde jsme označili $x_p = -\frac{d}{c}$, $y_p = \frac{a}{c}$, $k = \frac{bc-ad}{c^2}$.

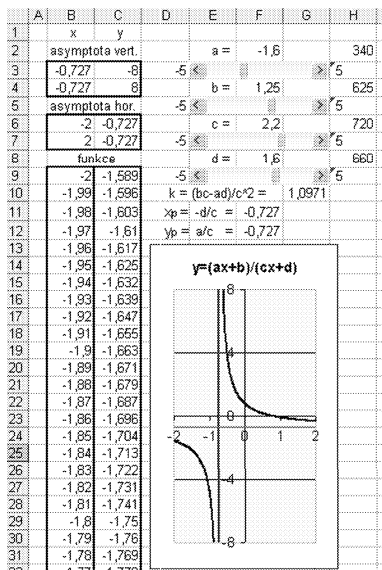
Z (1) je vidět, že grafem naší funkce je hyperbola vzniklá posunutím grafu funkce $y = \frac{k}{x}$. Jestliže by tedy novou souřadnicovou soustavu tvořily asymptoty zadané hyperboly, měla by naše hyperbola v této soustavě rovnici právě $y = \frac{k}{x}$.

Ještě se zabýváme „spojovací vlastností“ Excelu u této funkce. Pro $k > 0$ platí

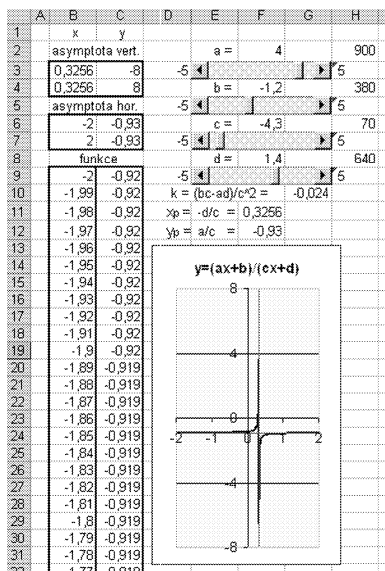
$$\lim_{x \rightarrow x_p-} \frac{ax+b}{cx+d} = -\infty, \quad \lim_{x \rightarrow x_p+} \frac{ax+b}{cx+d} = +\infty.$$

To znamená, že „poslední bod vlevo nahoře“ tabulky funkce, tj. bod $[x_1, y_1]$, kde $x_1 < x_p$ a je nejbližší k x_p , se nám v grafu spojí s bodem $[x_2, y_2]$, kde x_2 je nejmenší tabulkovou hodnotou nezávisle proměnné x , pro niž $x_2 > x_p$; spojovací čarou je úsečka, která jde velmi strmě zleva shora vpravo dolů. Je-li tabulka x dostatečně hustá, je strmost prakticky vertikální a my můžeme využít tuto „úsečku navíc“ tak, že ji považujeme za vertikální asymptotu. Na obr. 2 jsou vytvořeny obě asymptoty a protože obrázek není barevný, vidíme, že nedovedeme odlišit asymptotu od falešného grafu. Při nepříznivé volbě konstant (obr. 3) však může dopadnout obrázek katastrofálně.

Takto zpracovaný graf lze použít při probírání učiva v kapitole 5 [2] Lineární lomené funkce.



Obr. 2



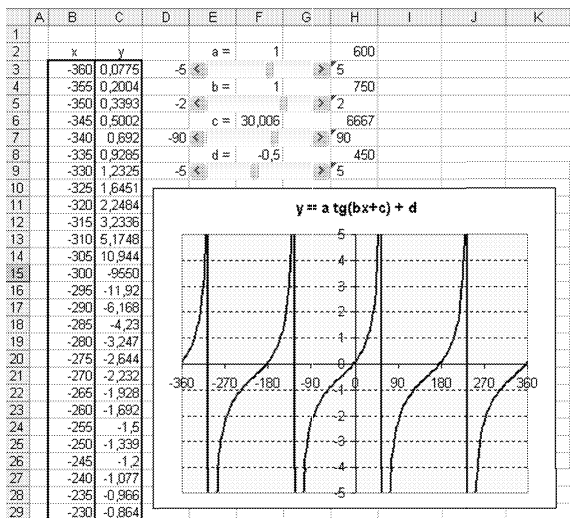
Obr. 3

Funkce $y = a \operatorname{tg}(bx + c) + d$

Je volen rozsah x od -360 do 360 s krokem 5 . Příprava posuvníků probíhá jako u předchozích funkcí, jen parametr c volíme ve větším rozsahu (zde od -90 do 90). Proto je vhodné pro tento parametr volit Formát ovládacího prvku jemněji, např. do 10000 . Při zadávání výpočtu hodnot funkce do tabulky musíme dát pozor na to, že Excel pracuje s goniometrickými funkcemi v obloukové míře, takže je třeba převést $bx + c$ do obloukové míry. Do buňky C3 tedy zadáme:

$$= \$F\$2 * \operatorname{TG}((B3 * \$F\$4 + \$F\$6) * \operatorname{PI}()) / 180) + \$F\$8$$

Spojení „konců“ jednotlivých větví opět vytváří dojem asymptot (obr. 4). Graf funkce tangens vychází troch „hrubě“, z toho důvodu jsme rozsah b omezili jen na interval $-2, 2$. Graf by bylo možno zjemnit tím, že bychom v tabulce volili krok 1 , pak ovšem rozsah tabulky velmi naroste a je užitečné zde znát příkazy *Ctrl Home*, resp. *Ctrl End* (z klávesnice) pro posun kurzoru na její začátek resp. konec.



Obr. 4

Grafy goniometrických funkcí s parametry připravené v Excelu lze využít při probírání učiva v části 2.5 a 2.6 [3].

Literatura

- [1] Matyáš, V.: Graf funkce s parametry v Excelu, MFI, roč. 17 (2007/08), č. 2, s. 106.
- [2] Odvárko, O.: Matematika pro gymnázia (Funkce). Prometheus, Praha 2002.
- [3] Odvárko, O.: Matematika pro gymnázia (Goniometrie). Prometheus, Praha 2002.