1. Úvod

Bakalářská práce se věnuje programu Maple. Jde o matematický program, ve kterém můžeme řešit veškeré matematické problémy. Umění řešit různé matematické úlohy není na škodu, protože v dnešní době je matematika vlastně všude. Hodně se používá například v bankovnictví a finanční praxi. Tato práce je zaměřena na využití programu Maple právě v oblasti finanční matematiky.

V jednotlivých kapitolách budou podány praktické návody, jak využít program Maple ve finanční matematice a jak lidem pomoci při řešení problémů z tohoto oboru. V úvodní kapitole seznámím čtenáře se základní charakteristikou programu Maple, s jeho uživatelským prostředím a základními příkazy a operacemi. Další kapitoly se budou věnovat konkrétním finančním funkcím, které jsou obsaženy v daném CAS programu a základním typům úloh z oboru finanční matematiky. Zmíníme se například o jednoduchém a složitém úročení, zjišťování efektivních úrokových sazeb, obligacích, umořování dluhů, počítání nákladů úvěrů, důchodech, rozhodování o investicích atd.

Výsledkem by mohla být základní orientace a práce v programu Maple a zároveň vysvětlení základních matematických postupů využívaných v bankovnictví a finanční praxi. To vše může pomoci k snadnějšímu rozhodování v podnikání či správě svých soukromých financí.

2. Základní popis systému Maple

2.1 Charakteristika systému Maple 9.5

Maple je programový systém počítačové algebry vyvinutý během uplynulých dvaceti pěti let společně na několika západních universitách, přičemž největší podíl práce vykonala skupina vědců sdružená pod názvem "Symbolic Computation Group" na universitě ve Waterloo v Kanadě a dále pak na federální technické universitě ETH Zürich ve Švýcarsku, kam část této skupiny přešla v roce 1990. V současné době je Maple komercializován a jeho další vývoj řídí kanadská firma Maplesoft Inc., (http://www.maplesoft.com) sídlící ve Waterloo ve státě Ontario.

Jméno Maple by mohlo být odvozeno z anglického akronyma Mathematics pleasure (Matematika potěšením), neboť Maple je skutečně příjemným prostředím pro využívání matematiky na počítači. Během posledních deseti let se Maple stal jedním z nejmodernějších a nejintenzivněji se rozvíjejících systémů počítačové algebry ve světě.

Současná verze 9.5 systému Maple (zkráceně Maple 9.5) umožňuje provádět jak symbolické a numerické výpočty a vytvářet grafy, tak doplňovat je vlastními texty a vytvářet tak tzv. hypertextové zápisníky (anglicky "worksheet"). Takto vytvořené zápisníky umožňuje Maple 9.5 ukládat do souboru na počítači ve svém speciálním maplovském formátu MW, který je uložen ve formátu XML. Soubory ve formátu MW umožňuje Maple 9.5 načítat zpět ke zpracování, což umožňuje snadnou přenositelnost maplovských zápisníků mezi nejrůznějšími počítačovými platformami a operačními systémy.

Soubory lze také volitelně exportovat do formátu Latexu, HTML, RTF a nově i MathML, což je rozšíření HTML pro prezentaci matematických textů na webu. Maple 9.5 dále umožňuje automatický převod svých příkazů a procedur do programovacích jazyků C, Fortran 77 a Java.

V Maplu 9.5 se používá vlastní programovací jazyk čtvrté generace podobný Pascalu s mnoha předdefinovanými funkcemi a procedurami. Maplovské funkce pokrývají mnoho odvětví matematiky od základů diferenciálního a integrálního počtu, lineární algebry, řešení rovnic, až k řešení diferenciálních a diferenčních rovnic, diferenciální geometrii a logice.

2.2 Uživatelské pracovní prostředí

V této sekci je popisováno uživatelské pracovní prostředí programu Maple 9.5, které odpovídá pracovním prostředím v operačních systémech Windows NT/2000/XP a Linux.

Systém počítačové algebry Maple 9.5 používá grafické uživatelské prostředí. Klíčovou částí v rámci tohoto prostředí je pracovní a komunikační rozhraní označované jako zápisník.

V rámci okna programu Maple 9.5 se standardně nachází hlavní nabídka menu a ovládací lišta reprezentovaná řadou ikon, viz obr 1.1. Pod ovládací lištou je kontextově závislá pracovní lišta s nabídkou aktuálně přístupných funkcí. Na levé straně je okno s takzvanými "paletami" umožňujícím jednoduché a rychlé vkládání výrazů, symbolů řecké abecedy, matic a vektorů. Toto okno lze libovolně dělit a přesouvat i na všechny ostatní strany plochy, jak je zjevné z obr. 1.1.

Hlavní část okna programu Maple 9.5 je vyhrazena pro pracovní plochu, na které je umístěn právě zpracovávaný zápisník, při spodním okraji okna programu je stavový řádek informující o časové a paměťové náročnosti právě zpracovávaného zápisníku.

Nabídka menu Maple 9.5 obsahuje vedle běžných menu pro práci se soubory (File, Edit, View, Window, Help) též některé kontextově závislé položky, jako například Style, Color, Axes, Projection, s aktuální nabídkou akcí a parametrů funkcí, použitelných v rámci aktivního objektu v zápisníku. Nejdůležitější z těchto funkcí jsou dostupné pomocí ikon na kontextově závislé pracovní liště. Většinu takto interaktivně nabízených parametrů funkcí (jako například způsob zobrazení souřadných os, barvu a tloušťku čar používaných v grafech atd.) je však možno definovat přímo jako parametry maplovských funkcí a příkazů použitých v zápisníku.

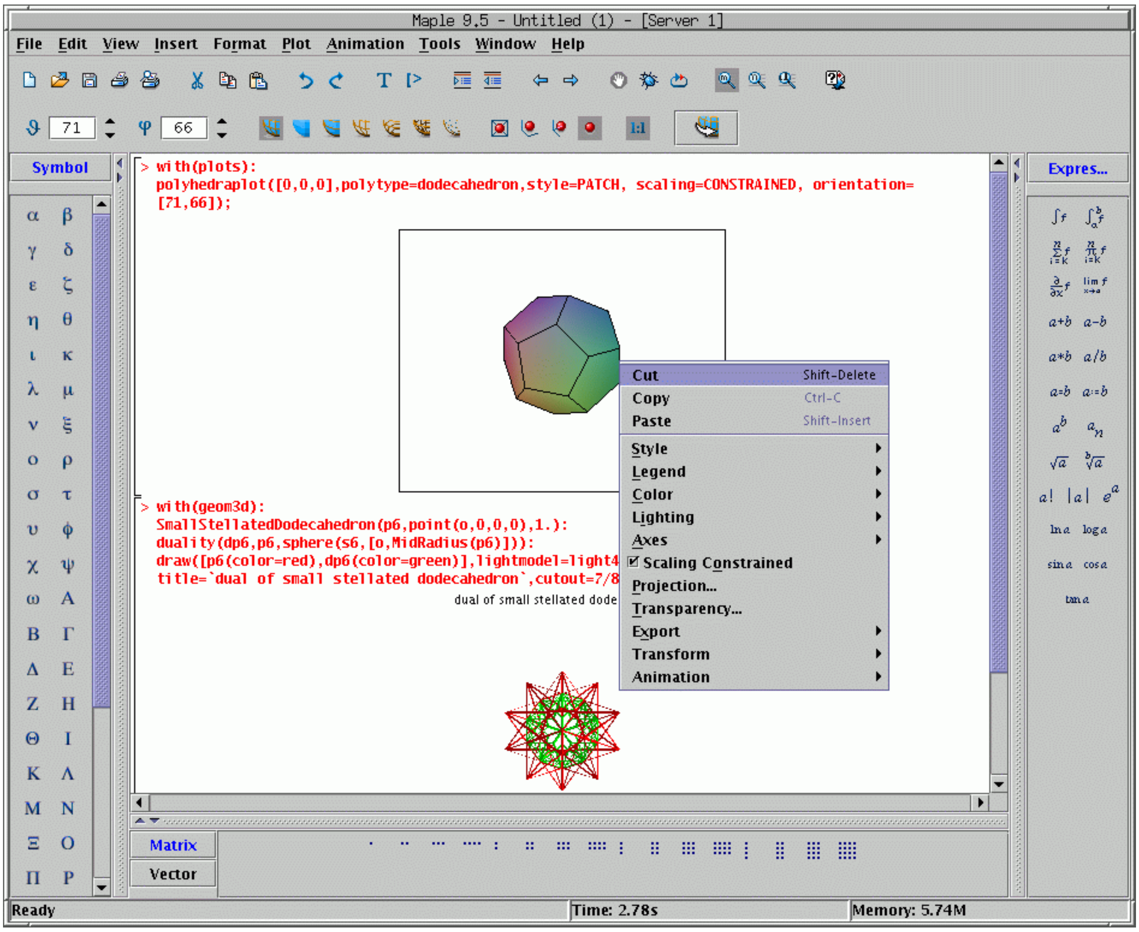
Na ovládací liště Maple 9.5 jsou dostupné běžné ikony usnadňující práci se zpracovávanými zápisníky, jako například otvírání, ukládání a tisk souborů zápisníku, práce se schránkou, funkce “zpět” (undo), přepínání mezi zapisováním do zápisníku formou prostého textu nebo formou aktivních maplovských příkazů, zastavení aktuálního výpočtu (na méně výkonných počítačích velice užitečná funkce), lupa a některé další.

Kontextově závislá pracovní lišta odvíjí svoji podobu od právě aktivního objektu v zápisníku. Na obr. 1.1 je vidět pracovní lišta s tlačítky vztahujícími se k 3D grafu funkce. Konkrétně nabízí otáčení 3D objektu kolem středu ve dvou směrech, přepínání mezi různými způsoby znázornění plochy, od vybarvené sítě až po tečkovanou interpretaci, různé způsoby znázornění souřadných os až po vyžádání konstantního měřítka na všech osách. Měřítka jednotlivých os automaticky přizpůsobují tvaru a rozsahu grafu.

Pro jednoduché interaktivní vkládání a editaci matematických výrazů jsou v levé liště umístěny navíc čtyři takzvané palety nazvané Výrazy, Symboly, Matice a Vektory. Umístění těchto menu lze změnit a umístit je nahoru, vlevo, vpravo nebo dolů, přičemž jednotlivé palety lze rozmístit na různé strany nebo případně zcela vypnout.

Práce s paletami je jednoduchá. Stačí kliknout na ikonu reprezentující naši volbu, a do zápisníku je na aktuální pozici vložen výraz, který již stačí pouze naplnit. Hodnoty, které je potřeba naplnit jsou přitom zvýrazněny. V prostředí Windows navíc lze ikonu “chytit” myší a umístit přímo na správnou pozici.

Pro úplné začátečníky, kteří se neorientují ve významu jednotlivých ikon, je tu možnost zapnout si interaktivní nápovědu, tzv. “balloon help”, automaticky vypisující názvy a stručnou charakteristiku funkcí spojených s danou ikonou.



Obrázek č. 1.1: Okno programu Maple 9.5 s dvěma otevřenými zápisníky

2.3 Práce se zápisníky

Zápisníky jsou hlavním uživatelským pracovním prostředím pro ovládání Maple 9.5. Umožňují uživateli pohodlné zadávání prováděcích příkazů, zároveň též slouží k okamžité prezentaci výstupů systému Maple 9.5. Po spuštění programu Maple 9.5 se na jeho pracovní ploše automaticky otevře nový prázdný zápisník.

Práce v novém zápisníku spočívá v zapisování vstupních příkazů Maple 9.5 do maplovské vstupní oblasti. Tyto příkazy jsou uvozeny symbolem “>” (větší než) a zobrazují se červeně. Ukončují se buď středníkem “;” nebo dvojtečkou “:”. Může se jich zapsat více na jeden řádek. Chceme-li je zapsat do samostatných řádků, tak po jejich ukončení musíme stisknout současně klávesy [Shift + Enter].

Po stisknutí klávesy [Enter] se všechny příkazy z maplovské vstupní oblasti provedou. Pokud je příkaz ukončen středníkem jeho výsledky se modře zobrazí, je-li však ukončen dvojtečkou, tak se jeho výsledky nezobrazí.

Množina vstupních oblastí s jejich odpovídajícími výstupy se v zápisníku Maplu nazývá prováděcí skupina (anglicky execution group). Zápisník dále může obsahovat samostatné textové oblasti v matematické notaci (anglicky paragraphs) a hypertextové odkazy (anglicky hyperlinks) a tabulkové kalkulátory (anglicky spreadsheet). Pro zpřehlednění lze zápisník rozdělit do sekcí (anglicky sections) a podsekcí.

2.4 Základní příkazy a operace

2.4.1 Způsob zápisu příkazů v zápisnících

Nejprve uvedeme několik obecných informací o způsobu zapisování příkazů na řádky zápisníku.

- Každý příkaz v zápisníku, obsahující příkaz Maple 9.5, musí být ukončen středníkem “ ; ” nebo dvojtečkou “ : ”.
- Po stisknutí klávesy [Enter] je celá aktuální prováděcí skupina předána jako vstup "výpočetnímu jádru" programu Maple 9.5, které ji zpracuje.

Pokud je řádek se zpracovávaným mapleovským příkazem zakončen středníkem, znamená to, že výsledek provedené operace se zobrazí na dalším řádku. Je-li řádek zakončen dvojtečkou, Maple 9.5 vyhodnotí zadaný příkaz, ale výsledek nezobrazí a očekává další příkaz. Tohoto se využívá zejména k potlačení tisku mezivýsledků v průběhu delších výpočtů.

Při zápisu posloupnosti příkazů či dlouhých algebraických výrazů je zapotřebí mít možnost přecházet na další řádek bez toho, že by se prozatím napsaný příkaz nějakým způsobem vyhodnocoval, jak se v zápisníku děje po prostém stisku klávesy [Enter]. To je možné provádět následujícím způsobem:

- Posunout kurzor na nový řádek, bez vyhodnocení dosud napsané části příkazu, nám v zápisnících umožňuje kombinace kláves [Shift] + [Enter].

2.4.2 Aritmetické operace a přiřazení hodnoty proměnné

V Maple 9.5 v aritmetických operacích pracuje se dvěma typy jmen: indexovaným jménem a neindexovaným jménem. Tato jména musí splňovat následující podmínky:

- musí začínat písmenem, ať už malým či velkým (POZOR! Maple 9.5 rozlišuje mezi malými a velkými písmeny)
- mohou být složeno z následujících znaků:
  - písmena,
  - číslice,
  - znaku podtržítko “ _ ”

- maximální délka jména je závislá na počítačové platformě, na 32-bitových systémech (případ Windows 9x) je to 524 271 znaků, na systémech 64-bitových je maximální délka jména proměnné 34 359 738 335 znaků.

Poznámka:

Znak podtržítko “_” na začátku jména proměnné je vyhrazen pro globální systémové proměnné, proto se doporučuje taková jména nepoužívat. Jména obsahující lomítko (/) jsou obecně rezervována pro kódy v knihovnách Maple a neměla by se používat, pokud to není explicitně uvedeno v nápovědě. Jména, která končí vlnovkou (~) označují v Maple proměnné předepsanými předpoklady (např. jejich hodnota je kladná, záporná, atd.) a rovněž by neměla být používána.

Každé jméno a obecně i výraz má v Maple přiřazen typ. Těchto typů je velké množství a nebudeme je zde všechny probírat. U výrazu se jménem \`vyraz\` zjistíme jeho typ pomocí příkazu whattype(vyraz), který nám dá jako výsledek, že se jedná o:

- Aritmetický, relační nebo logický operátor(\`*\`, \`+\`, \`.\`, \`..\`, \`::\`, \`<\`, \`<=\`, \`<>\`, \`=\`, \`^\`, \`and\`, \`not\`, \`or\`),
- Celé, racionální, reálné číslo v pohyblivé řádové čárce nebo komplexní číslo (integer, fraction, float, complex),

- Strukturu typu pole, matice, sloupcový nebo řádkový vektor, tabulka, možina, seznam, textový řetězec (array, Matrix, Vector[column], Vector[row], table, set, list, string),
- Funkci, proceduru, posloupnostní výraz nebo nekonečnou řadu (function, procedure, exprseq, series),
- Nevyhodnotitelný výraz (uneval).

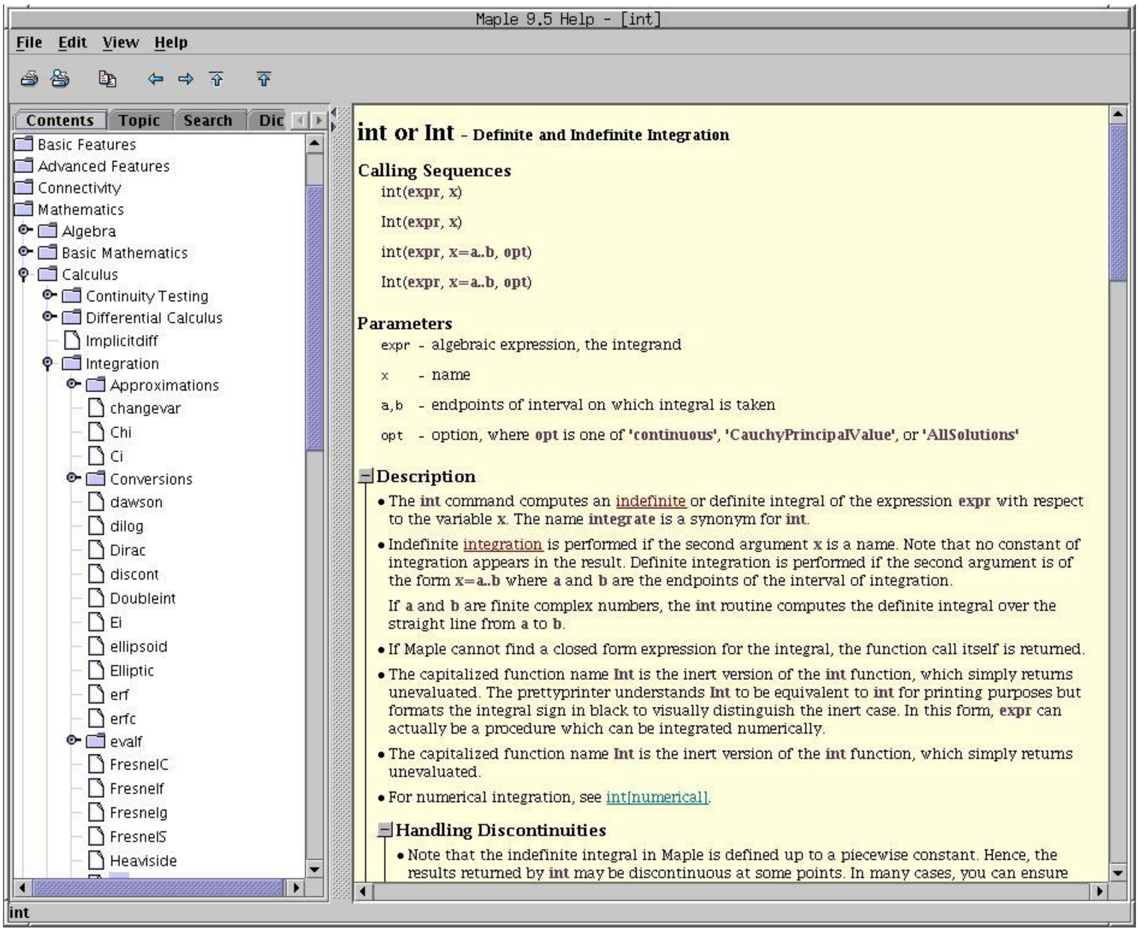
2.5 Jednoduchá nápověda k maplovskému příkazu

Velmi důležitým pomocníkem ve využívání Maplu je jeho nápověda. Tato interaktivní nápověda v Maple 9.5 slouží ke snadné a rychlé orientaci v tisících maplovských příkazů, funkcí, knihoven a balíků a jejich parametrů. V Maple 9.5 je nápověda řešena jako systém textových dokumentů propojených hypertextovými odkazy. Každá standardní funkce (příkaz) Maple 9.5 má zpracovánu vlastní stránku s nápovědou. Jednotlivé stránky nápovědy k maplovskému příkazu mají pevnou strukturu, skládající se z následujících po sobě jdoucích částí:

- název a charakteristika příkazu
- popis volání příkazu
- definice parametrů příkazu
- podrobný popis vlastností příkazu
- příklady použití příkazu
- seznam hypertextových odkazů na příbuzná témata v nápovědě Maple 9.5

Systém interaktivní nápovědy v Maple 9.5 spočívá v tom, že umožňuje zpřístupnit přímo stránku týkající se zadané funkce. Ukážeme si to na příkladu.

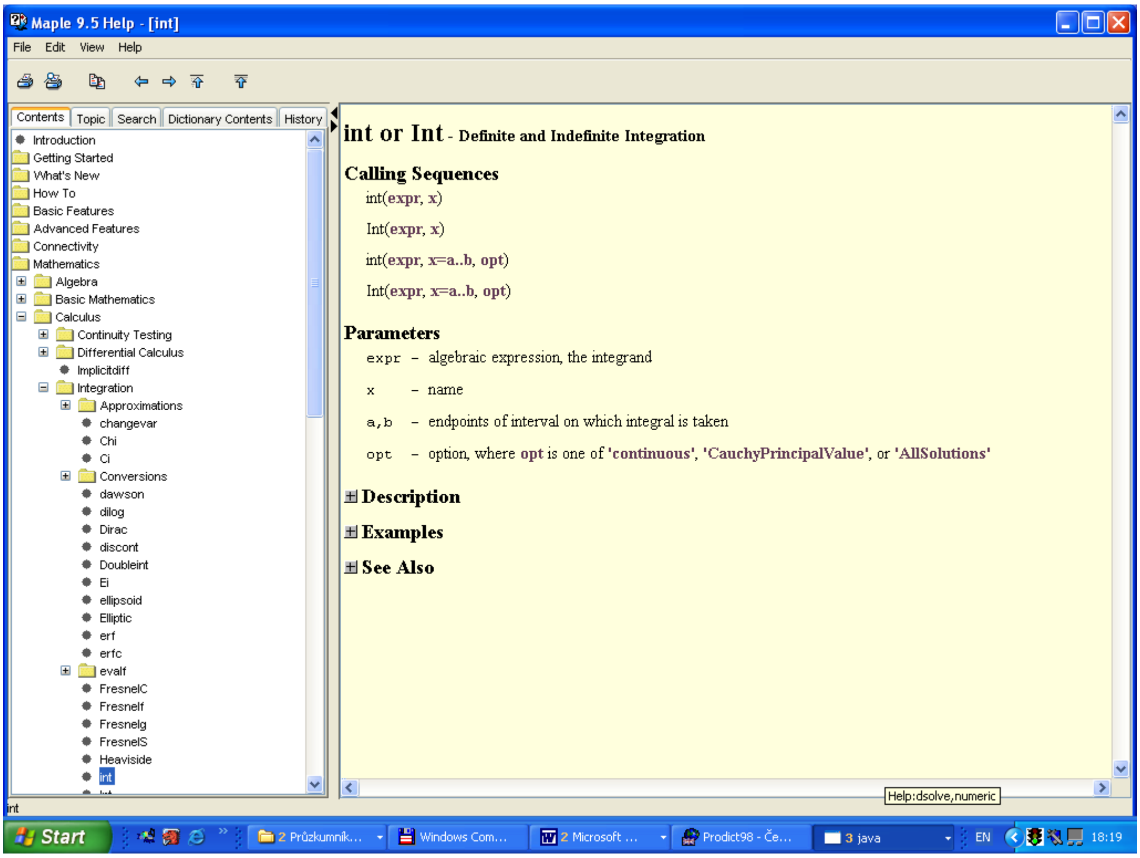
Potřebujeme-li například nápovědu k příkazu int, napišme na řádek zápisníku příkaz `?int` a stiskněme klávesu [Enter]. Tím se zobrazí v novém okně požadovaná stránka nápovědy.



Obrázek č. 1.2: Nápověda k příkazu int

Vzhledem k tomu, že nápověda bývá u některých příkazů velmi rozsáhlá, je možné si ji zobrazit ve “sbaleném” tvaru pomocí příkazu

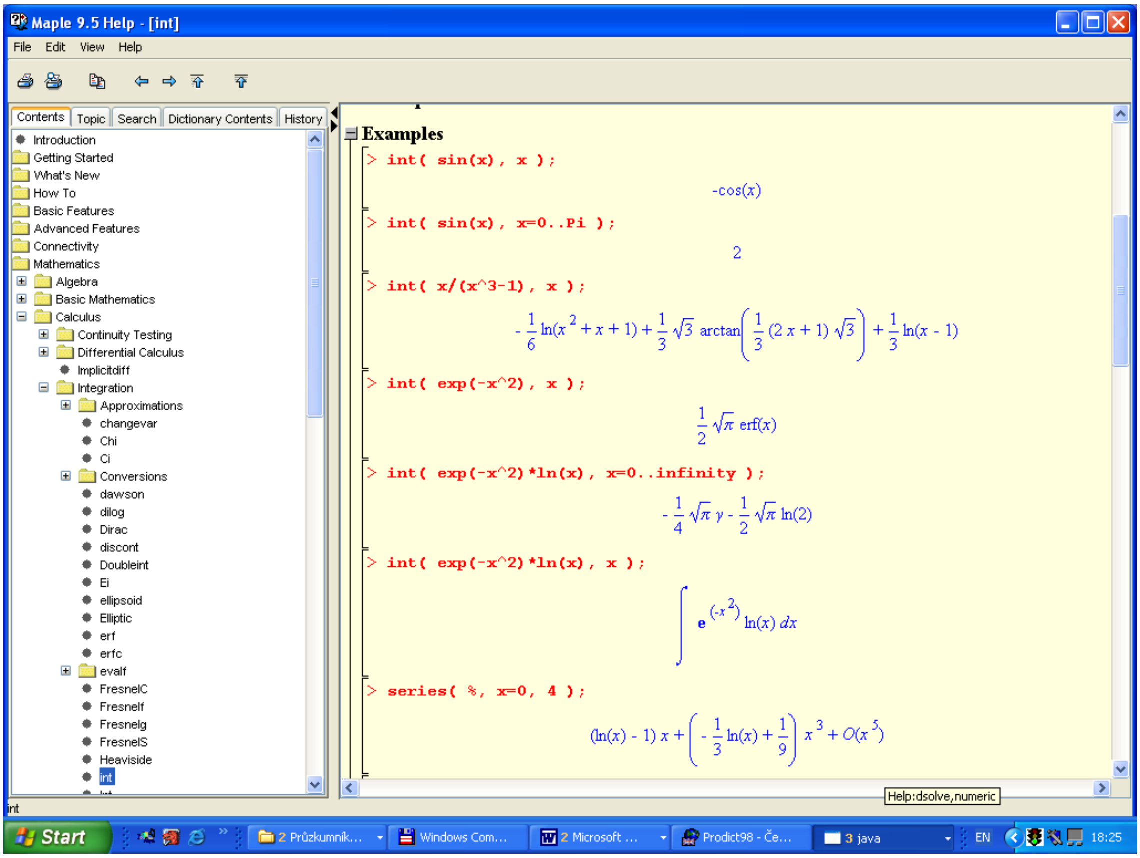
```
??int
```



Obrázek č. 1.3: “Sbalená” nápověda k příkazu int

Někdy se stačí jen podívat na příklady použití daného příkazu *int* uvedené v nápovědě. Pak stačí zapsat příkaz

```
???int
```



Obrázek č. 1.4: Příklady z nápovědy k příkazu int

Je-li již jméno funkce *int* v zápisníku použito, tak stačí na něj přenést kurzor myší či klávesovými šipkami, a poté vybrat položku \`Help on int\` z menu Help zápisníku nebo jen stisknout klávesy [Ctrl + F1].

Pokud jsme v situaci, kdy hledáte funkci, jejíž jméno neznáte, nezbývá vám, než procházet systém stránek nápovědy manuálně. Systém stránek nápovědy je však hierarchicky členěn a díky tomu se v něm dá velice dobře vyhledávat. Otevřít systém nápovědy můžeme například pomocí položky Using Help v menu Help.

2.6 Knihovny funkcí

Maple 9.5 při řešení úloh umožňuje použít obrovské množství příkazů a matematických funkcí. Ve standardní instalaci současné verze Maple 9.5 je dostupných více než 3000 funkcí. Funkce jsou uloženy v takzvané knihovně funkcí.

Z důvodu zpřehlednění práce se stovkami přístupných funkcí jsou funkce v rámci knihovny rozděleny do takzvaných balíků (packages).

Kromě standardní knihovny funkcí je možno s programem Maple 9.5 využívat též tzv. share library. Tato knihovna je sestavena z funkcí a balíků napsaných uživateli Maple 9.5 a obsahuje mnoho funkcí použitelných přímo v praxi. Tato knihovna je nyní uložena na webovském serveru Maple na adrese http://www.mapleapps.com v tzv. "Maple Application Center".

Standardní knihovní funkce – Standard library functions

Funkce (příkazy) z tohoto balíku jsou automaticky přístupné v Maple a je možné je volat jejich jménem ihned po spuštění Maple. Není nutné je inicializovat jako funkce z ostatních balíků. Sem patří hlavně běžné matematické funkce ale také příkazy Maple 9.5, umožňující manipulaci a vyhodnocování zpracovávaných výrazů. Seznam standardních knihovních funkcí je dostupný příkazem:

```
> ?index[function];
```

Knihovní balíky – Library packages

Knihovní balíky jsou určité množiny příkazů, v rámci dané knihovny příkazů systému Maple 9.5. Většinou jsou to funkce (příkazy) vztahující se k řešení určité třídy matematických úloh, které jsou společně zařazeny do jedné knihovny balíku příkazů. Potřebujeme-li například pracovat s objekty třírozměrného euklidovského prostoru, tak všechny příkazy vztahující se k této problematice jsou zařazeny do jednoho balíku, konkrétně se jedná o balík geom3d.

Pokud chceme používat příkazy z některého balíku, existují dva způsoby jejich volání:

1.volání pomocí dlouhých jmen (long names)
2.volání pomocí krátkých jmen (short names)

Volání pomocí dlouhých jmen

Dlouhá jména příkazů není třeba inicializovat. Dlouhé jméno příkazu je určeno jménem balíku, do kterého je příkaz zařazen a jménem příkazu v rámci balíku.

Chceme-li například použít příkaz *point* z balíku geom3d, pak jej voláme následovně: geom3d[point]. Tento způsob volání vylučuje kolizi jmen příkazů z různých balíků. Možnost nechtěného předefinování významu některého příkazu během výpočtu může být velmi nebezpečné. Uvědomíme-li si navíc, že v programu Maple 9.5 můžeme používat tisíce různých příkazů, vidíme, že opatření zabraňující takovéto chybě jsou zcela nezbytná.

Volání pomocí krátkých jmen

Výše uvedenému volání příkazů pomocí dlouhých jmen se můžeme vyhnout pomocí inicializace zkráceného volání příkazů. Tato možnost slouží hlavně k zjednodušení práce Maplu a usnadňuje též manipulaci s příkazy.

K inicializaci používání krátkých jmen příkazů z daného balíku slouží příkaz *with*, jehož parametrem je jméno balíku. Tímto příkazem inicializujeme současně všechny příkazy z balíku. Pokud jako druhý parametr uvedeme jméno konkrétního příkazu, tak se inicializuje pouze uvedený příkaz z balíku, např. příkazem with(plots,animate) se inicializuje příkaz *animate* z balíku *plots*.

Při inicializaci určitého knihovního balíku pomocí příkazu *with* jsou na výstupu vypsána všechna jména jeho nově inicializovaných příkazů. Pokud došlo ke shodě jmen některého inicializovaného příkazu s již definovaným jménem příkazu či proměnné, platí význam nově inicializovaného příkazu!

Ukážeme to na příkladu zkráceného volání příkazů z balíku *plots*:

```
> with(plots);
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *odeplot*, *pareto*,

pointplot, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *replot*, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

Seznam všech dostupných knihovních balíků i s jejich stručným popisem je dostupný pomocí příkazu:

```
> ?index[packages];
```

3. Výpis základních finančních funkcí v programu Maple

3.1 finance[annuity] - pravidelná platba ve stejné výši (Př. 11)

Vyvolání

annuity(cash, rate, nperiods)

Parametry

cash	…	peněžní tok
rate	…	úroková sazba
nperiods	…	počet období

Popis

- Funkce annuity (splátka, platba) udává současnou hodnotu v čase = 0 z pravidelných plateb ve stejné výši za n období = peněžní tok, startující v čase = 1.
- Hypotéka, úvěr, dluh, důchod jsou základní příklady, kde se pravidelné platby, splátky objevují.
- příkaz with(finance,annuity) dovolí použít zkrácenou formu příkazu.

3.2 finance[amortization] - umořovací tabulka z úvěru (Př. 5)

Vyvolání

amortization(amount, payments, rate, nperiods)

Parametry

amount	…	dlužná částka
payments	…	velikost jednotlivých plateb
rate	…	úroková sazba
nperiods	…	maximální počet plateb
	…	zastaví se, když zůstatek je roven 0

Popis

- Výsledek je složen ze dvou složek: umořovací tabulka a cena dluhu.
- Umořovací tabulka je seznam zahrnující 5 základních prvků
    1) počet období
    2) výše splátek
    3) výše úroku z dluhu
    4) výše úmoru dluhu
    5) stav dluhu (zbývající dlužnou částku).
- Cena dluhu je součet splátkového sloupce.
- Odkázat se můžeme na finance(annuity) pro více detailů ze závazků nebo na finance(effectiverate) pro výběr vhodné hodnoty úrokové sazby k budoucímu použití.
- Příkaz with(finance, amortization) dovolí použít zkrácenou formu příkazu.

3.3 finance[cashflows] - současná hodnota ze seznamu peněžních toků (Př. 2)

Vyvolání

cashflows(flows,rate)

Parametry

flows … seznam peněžních toků, jeden za období, startující v období = 1
rate … úroková sazba za období

Popis

- Funkce cashflows() počítá současnou hodnotu ze seznamu peněžních toků. Toky jsou určeny jeden za období, startující v období = 1.
- Příkaz with(finance, cashflows) dovolí použít zkrácenou formu příkazu.

3.4 finance[effectiverate] - určí efektivní úrokovou sazbu (Př. 1, 11)

Vyvolání

effectiverate(stated_rate, nperiods)

Parametry

state_rate … úroková sazba, vyjádřená jako desetinné číslo
nperiods … počet připisování úroků v období (rok)

Popis

- Při připisování úroků m-krát do roka, kde m → ∞ (úroková intenzita), dosadíme za parametr *nperiods* hodnotu *infinity*, což je nekonečno.
- Příkaz with(finance, effectiverate) dovolí použít zkrácenou formu příkazu.

3.5 finance[futurevalue] - budoucí hodnota z určité částky (Př. 11, 13)

Vyvolání

futurevalue(amount, rate, nperiods)

Parametry

amount	…	částka za období = 0
rate	…	úroková sazba za období
nperiods	…	počet období

Popis

- Tato funkce počítá hodnotu za období = *nperiods* z částky dané v období = 0, kde úroková sazba je daná.
- Funkce futurevalue() nese částku směrem k budoucnosti, funkce presentvalue() ji nese směrem k minulosti.
- Příkaz with(finance, futurevalue) dovolí použít zkrácenou formu příkazu.

3.6 finance[levelcoupon] - současná hodnota obligace s pevnou úrokovou sazbou (Př. 3)

Vyvolání

levelcoupon(face, rate, couponrate, maturity)

Parametry

face	…	nominální hodnota obligace
rate	…	tržní úroková sazba

couponrate … úroková sazba udávaná u obligace

maturity … počet období do splatnosti obligace

Popis

- Funkce levelcoupon() počítá současnou hodnotu všech budoucích plateb plynoucích z daného dluhopisu.
- Současná hodnota dluhopisu, u kterého je jen nominální hodnota se splatností, ale žádným kuponovým placením (dluhopis s nulovou úrokovou sazbou – nulovým kuponem) je počítán pomocí funkce finance(presentvalue).
- Příkaz with(finance,levelcoupon) dovolí použít zkrácenou formu příkazu.

3.7 finance[presentvalue] - současná hodnota z určité částky (Př. 4, 6, 8)

Vyvolání

presentvalue(amount, rate, nperiods)

Parametry

amount … daná částka

rate … úroková sazba

nperiods … počet období

Popis

- Funkce presentvalue() udává současnou hodnotu v čase = 0 z částky *amount*, dané v čase = *nperiods*. Za *rate* dosadíme úrokovou sazbu jako desetinné číslo.
- Příkaz with(finance,presentvalue) dovolí použít zkrácenou formu příkazu.

4. Příklady se základními typy úloh z finanční matematiky

<u>Příklad 4.1</u> *Efektivní úroková sazba, úroková intenzita*

Rozhodli jsme si založit termínovaný účet u banky, která nabízí dva typy účtů:

a) **Účet s 8 % roční úrokovou sazbou s denním připisováním.**

b) **Účet s 8.5 % roční úrokovou sazbou se čtvrtletním připisováním.**

Určete co je pro nás výhodnější.

<u>Řešení</u>

Jednoduše si můžeme spočítat efektivní úrokové sazby u jednotlivých účtů. Efektivní úroková sazba nám umožňuje porovnat různé úrokové sazby za stejné období, avšak s různou četností připisování úroků. Např. roční efektivní úroková sazba nám říká, jak velká roční úroková sazba při ročním připisování úroků odpovídá roční úrokové sazbě při častějším připisování úroků. Zde ideálně poslouží funkce *effectiverate*.

Účet a)

```
> with(finance):
> effectiverate( 0.08, 12 );
```

$$.083277539$$

První účet má efektivní úrokovou sazbu přibližně 8,33 %.

K tomuto výsledku též dojdeme, jestliže hodnoty dosadíme do vzorečku pro efektivní úrokovou sazbu $i_e = \left(1 + \frac{i}{m}\right)^m - 1$, tzn. bez použití funkce *effectiverate*. Za i dosadíme velikost úrokové sazby a za m počet připisování úroků za rok.

Pro ukázku si vzoreček můžeme zkusit zobrazit i v programu Maple.

```
> ie:=(1+i/m)^m-1;
```

$$ie := \left(1 + \frac{i}{m}\right)^m - 1$$

Pro přehlednost si velikost úrokové sazby a počet připisování úroků za rok vypíšeme.

```
> i:=0.08; m:=365;
```

$$i := .08$$

$$m := 365$$

Nyní už konečný výsledek efektivní úrokové sazby.

```
> ie;
```

$$.083277539$$

Účet b)

```
> effectiverate( 0.085, 4 );
```

$$.087747962$$

Druhý účet má přibližně 8,77 %. Tento účet je tedy pro nás výhodnější, protože je zde větší efektivní sazba.

<u>Doplněk:</u>

Můžeme si představit, že počet úrokových období, ve kterých se připisují úroky, poroste až do nekonečna, jejich délka se tudíž zkracuje a klesá k nule. Efektivní úroková sazba odpovídající tomuto případu se nazývá **úroková intenzita.**

Pro ukázku si zkusme u druhého účtu spočítat úrokovou intenzitu pomocí programu Maple. To uděláme tak, že za počet připisování úroků za rok dosadíme nekonečno, což vyjádří slovíčko *ifinity.*

```
> effectiverate( 0.085, infinity);
```

$$.088717067$$

Úroková intenzita vyšla přibližně 8,87 %.

<u>**Příklad 4.2**</u> *Vnitřní výnosové procento*

Jako ředitel malé společnosti zvažujete dvě možné alternativní investice na dobu 6 let, přičemž se můžete rozhodnout pouze pro jednu z nich. Očekávané peněžní toky, které jsou s nimi spojeny, jsou následující (uváděné částky jsou v Kč):

	Vložený kapitál	Peněžní toky v jednotlivých letech
Investice 1	100 000	25 000 ročně
Investice 2	100 000	24 000; 25 000; 27 000; 27 000; 26 000; 22 000

<u>**Řešení**</u>

Při rozhodování o více investičních variantách můžeme postupovat i tím způsoben, že budeme hledat takovou úrokovou míru, při které se bude rovnat součet současných hodnot budoucích toků z investice hodnotě právě vloženému kapitálu. Tuto úrokovou míru nazýváme **vnitřní výnosové procento.**

<u>**Investice 1**</u>

Nejdříve si spočteme výši všech toků dohromady za 6 let.

```
> with(finance):
> 6*25000;
```

$$150000$$

Jestliže tyto peněžní toky jsou vytvořeny s počáteční investice 100 000 Kč, pak čistá současná hodnota je

```
> csh:=150000-100000;
```

$$csh := 50000$$

Základ pro přijetí tohoto projektu je, aby vnitřní výnosové procento bylo kladné.

Zobrazíme si základní rovnost $K = \frac{u_1}{(1+i)} + \frac{u_2}{(1+i)^2} + \cdots + \frac{u_n}{(1+i)^n}$, podle které se vnitřní výnosové procento počítá. Hodnota *K* představuje počáteční investici, u_1 až u_n jsou jednotlivé peněžní toky a neznámá *i* je vnitřní výnosové procento, které hledáme. Uděláme tak pomocí funkce *cashflows*.

```
> X:=-100000+cashflows(
[25000,25000,25000,25000,25000,25000],i);
```

$$X := -100000 + \frac{25000}{1+i} + \frac{25000}{(1+i)^2} + \frac{25000}{(1+i)^3} + \frac{25000}{(1+i)^4} + \frac{25000}{(1+i)^5} + \frac{25000}{(1+i)^6}$$

Nyní stačí vnitřní výnosové procento vytknout a vypočítat. Použijeme příkaz *fsolve*.

```
> fsolve(X=0, i, 0..2);
```

$$.1297800069$$

Výpočtem získáme i = 12,98 %.

<u>Investice 2</u>

Postupujeme obdobně.

```
> 24000+25000+27000+27000+26000+22000;
```

$$151000$$

```
>csh:=151000-100000;
```

$$csh := 51000$$

```
> X:=-100000+cashflows(
[24000,25000,27000,27000,26000,22000],i);
```

$$X := -100000 + \frac{24000}{1+i} + \frac{25000}{(1+i)^2} + \frac{27000}{(1+i)^3} + \frac{27000}{(1+i)^4} + \frac{26000}{(1+i)^5} + \frac{22000}{(1+i)^6}$$

```
> fsolve(X=0, i, 0..2);
```

$$.1327578890$$

Výpočtem získáme i = 13,28 %.

Podle provedených výpočtů vidíme, že lepší je druhá investice, protože její výnosové procento je vyšší.

Příklad 4.3 *Teoretická cena dluhopisu*

Máme dluhopis o nominální hodnotě 10 000 Kč a pevnou úrokovou sazbou 12 %. Doba splatnosti dluhopisu je 3 roky a tržní úroková sazba je 10 % za rok. Jaká je současná hodnota dluhopisu?

Řešení

Teoretickou cenu dluhopisu můžeme odvodit z podstaty dluhopisu jako cenného papíru, ze kterého plynou majiteli během doby do splatnosti určité výnosy a v době splatnosti jmenovitá hodnota. Teoretická cena dluhopisu potom není nic jiného, než současná hodnota všech budoucích plateb, plynoucích z daného dluhopisu.

Tuhle současnou hodnotu můžeme spočítat podle vzorečku na teoretickou cenu dluhopisu s pevnou úrokovou sazbou $P = \frac{C}{1+i} + \frac{C}{(1+i)^2} + \cdots + \frac{C}{(1+i)^n} + \frac{NH}{(1+i)^n}$, kde *C* je roční kupónová úroková platba (12 % z nominální hodnoty dluhopisu), *NH* je nominální hodnota dluhopisu, *i* je úroková sazba a *n* je doba do splatnosti dluhopisu v letech. Mnohem jednodušší však bude výpočet pomocí funkce *levelcoupon*.

```
> with(finance):
> levelcoupon(10000, 0.10, 0.12, 3);
```

$$10497.37040$$

Současná hodnota vyšla skoro 10 500 Kč.

Pozn.:

Kdybychom platili dluhopis dvakrát ročně, musíme si dát pozor na zadávané parametry. Tržní úrokovou sazbu a úrokovou sazbu dluhopisu dělíme dvakrát, protože platíme dvakrát za rok a sazby jsou zadané jako roční. Naopak poslední parametr zvětšíme, protože počet období do splatnosti dluhopisu se nám zdvojnásobí.

```
> levelcoupon(10000, 0.10/2,  0.12/2, 6);
```

$$10507.56921$$

Cena vyšla opět okolo 10 500 Kč.

Příklad 4.4 *Teoretická cena dluhopisu s nulovým kuponem*

Vypočítejte teoretickou cenu dluhopisu s nulovým kuponem se splatností dva roky, jmenovitá hodnota dluhopisu činí 1 000 Kč, při tržní úrokové míře 5,5 % p.a.

Řešení:

Tento dluhopis není spojen se žádnými úrokovými platbami během doby splatnosti. Teoretickou cenu dluhopisu s nulovým kuponem proto určíme jako současnou hodnotu jmenovité hodnoty splatné v době splatnosti. Používá se vzorec $P_{NK} = \frac{NH}{(1+i)^n}$, kde *NH* je nominální hodnota dluhopisu, *i* je tržní úroková sazba a *n* je počet let.

Vzoreček si můžeme zobrazit v Maplu.

```
> Pnk:=NH/(1+i)^n;
```

$$Pnk := \frac{NH}{(1+i)^n}$$

Vypíšeme si dané veličiny.

```
>  NH:=1000; i:=0.055; n:=2;
```

$$NH := 1000$$

$$i := 0.055$$

$$n := 2$$

```
> Pnk;
```

$$898.4524157$$

Teoretická cena (kurz) dluhopisu činí 898,45 Kč to je 89,85 %.

Vše si však jde mnohem zjednodušit použitím funkce *presentvalue,* což je v podstatě ukázka složeného úročení a nádherně se hodí na typ tohoto úkolu. Představme si, že budeme počítat, jak velký kapitál musíme dnes uložit, abychom po čase *n* a při úrokové sazbě *i,* dosáhli kapitálu rovnému *NH.*

```
> presentvalue(1000, .055, 2);
```

$$898.4524157$$

Jestliže chceme za dva roky mít 1000 Kč a budeme od banky dostávat 5,5 % ročně, musíme nyní vložit skoro 900 Kč.

Příklad 4.5 *Umořování úvěru*

Máme úvěr 1 000 Kč, neměnnou roční úrokovou sazbu 10% a naše platby budou po 350-ti Kč ročně. Sestavte umořovací plán, zjistěte, jak dlouho budeme splácet a jaký bude náklad za pořízení úvěru.

Řešení

Naším cílem bude sestrojit umořovací plán (tabulku), ze které můžeme následně všechno potřebné vyčíst. Nejjednodušší bude použití funkce *amortization*, která se přímo hodí na úlohu tohoto typu.

```
> with(finance):
> amortization( 1000.00, 350.00, 0.10 );
```

[[0, 0, 0, -1000.00, 1000.00], [1, 350.00, 100.0000, 250.0000, 750.0000],
[2, 350.00, 75.000000, 275.000000, 475.000000],
[3, 350.00, 47.50000000, 302.5000000, 172.5000000],
[4, 189.7500000, 17.25000000, 172.5000000, 0.]], 239.7500000

Teď si pro přehlednost přepíšeme výsledek (hranaté závorky) do takovéhle umořovací tabulky

Období	Splátky	Úrok	Úmor	Zůstatek úvěru
0	0	0	- 1000	1000
1	350	100	250	750
2	350	75	275	475
3	350	47.5	302.5	172.5
4	189.75	17.25	172.5	0

Nyní máme před sebou umořovací plán. Z tabulky je krásně vidět, že budeme úvěr splácet 4 roky. Dále vidíme výši splátky za každý rok, výši úroku z dluhu, výši úmoru dluhu a stav dluhu po odečtení úmoru (zbývající dlužnou částku). Náklad za pořízení úvěru je 239.75 Kč (poslední číslo ve výsledku), což znamená, že jsme si půjčili 1000 Kč, ale splátky se nám díky úrokům vyšplhaly na 1239.75 Kč.

Příklad 4.6 *Složené úročení*

Rozhodli jsme se založit svému právě narozenému dítěti terminovaný bankovní účet spojený s pevnou úrokovou sazbou 5 % ročně a uložit dnes na účet 300 000 Kč. Úroky budou připisovány k vkladu a dále úročeny stejnou sazbou. Budeme chtít účet vybrat, až se částka vyšplhá na 1 000 000 Kč. Za kolik let to bude?

Řešení

Pro znázornění postupu použijeme funkci *presentvalue.* Nebudeme však počítat základní vklad, poněvadž ten známe. Budeme počítat neznámou *n*, která představuje počet let, za které peníze z účtu vybereme.

```
> with(finance):
> pocvklad:=presentvalue(1000000, .05, n);
```

$$pocvklad := \frac{1000000}{1.05^n}$$

Nyní už jen spočítáme neznámou pomocí funkce *fsolve.*

```
> n:=fsolve(pocvklad=300000, n);
```

$$n := 24.67654751$$

Jestliže budeme chtít vybrat z účtu 1 000 000 Kč, musíme si počkat přibližně 24 let a 8 měsíců.

__Příklad 4.7__ *Jednoduché úročení*

Odběratel nezaplatil fakturu na 193 000 Kč splatnou 7. července 2005. Podle smlouvy účtujeme penále ve výší 0,05 % z faktury za každý den prodlení. Jak velké bude penále k 9. září 2005?

__Řešení__

Nejdříve si spočítáme, kolik dní trvá zpoždění platby (červenec + srpen + září).

```
> pocetdni:=24+31+9;
```

$$pocetdni := 64$$

V tomto případě se jedná v podstatě o jednoduché úročení, takže můžeme pokračovat podle vzorce pro konečný kapitál při jednoduchém polhůtním úročení $FV = PV.(1 + i.t)$, kde PV představuje počáteční kapitál, i úrokovou sazbu (výši penále v %) a t dobu splatnosti (počet dní v prodlení).

Zobrazení vzorce v Maplu.

```
> FV:=PV*(1+i*t);
```

$$FV := PV(1 + i\,t)$$

Vypsání zadaných veličin.

```
> PV:=193000; i:=0.0005; t:=64;
```

$$PV := 193000$$

$$i := .0005$$

$$t := 64$$

```
> FV;
```

$$199176.0000$$

Nyní ještě spočítáme čistou výši penále. Tu dostaneme po odečtení faktury.

```
> penale:=FV-193000;
```

$$penale := 6176.0000$$

Penále se za 64 dní vyšplhalo na 6 176 Kč.

<u>Příklad 4.8</u> *Současná hodnota budoucího kapitálu (složené úročení)*

Máme možnost koupit za 4 700 Kč investici (diskontovanou obligaci), která nám umožní získat za dva roky částku 5 000 Kč. Je to výhodná investice, uvažujeme-li útokovou sazbu 3 % ročně a roční připisování úroků?

<u>Řešení</u>

Budeme počítat současnou hodnotu budoucího kapitálu. Současná hodnota kapitálu nám říká, jak velký kapitál musíme dnes uložit, abychom po daném čase (2 roky), při dané

úrokové sazbě (3 %) a za předpokladu reinvestování úroků, to je při složeném úročení, dosáhli kapitálu 5 000 Kč.
Tohle umí bez vypisování do vzorečků spočítat funkce *presentvalue*.

```
> with(finance):
> presentvalue(5000, .03, 2);
```

$$4712.979546$$

Současná hodnota vyšla 4712,98 Kč, což je vyšší než požadovaná investice. Tedy abychom za dva roky měli 5 000 Kč, museli bychom dnes uložit částku 4 712,98 Kč. Nám však stačí dát částku 4 700 Kč a tedy investice je za daných předpokladů výhodná.

Příklad 4.9 *Krátkodobé spoření polhůtní*

Kolik uspoříme do konce roku, ukládáme-li koncem každého měsíce 1 700 Kč při úrokové sazbě 2 % p.a.?

Řešení

Použijeme vzoreček na krátkodobé spoření polhůtní $S_x = m.x.\left(1 + \frac{m-1}{2m}.i\right)$, poněvadž spoříme jen jeden rok a ukládáme vždy na konci měsíce. Budeme počítat celkovou uspořenou částku S_x na konci roku včetně úroků i, jestliže spoříme každou m-tinu roku částku x Kč.

Zobrazení vzorečku v Maplu.

```
> S:=m*x*(1+(m-1)/(2*m)*i);
```

$$S := m\,x\left(1 + \frac{\frac{1}{2}(m-1)\,i}{m}\right)$$

Zadané proměnné.

```
> m:=12; x:=1700; i:=0.02;
```

$$m := 12$$

$$x := 1700$$

$$i := .02$$

```
> S;
```

$$20587.00001$$

Do konce roku uspoříme 20 587 Kč.

__Příklad 4.10__ *Dlouhodobé spoření předlhůtní*

Za pět let plánujeme nákup nového automobilu. Značka, kterou jsme si vybrali, má dle prognóz vývoje cen stát v té době 750 000 Kč. Kolik musíme tedy spořit na počátku každého roku, abychom za pět let uspořili 750 000 Kč? Úspory dáváme na účet úročený sazbou 2,5 % p.a. a ročním připisováním úroků.

__Řešení:__

Jedná se o spoření dlouhodobé předlhůtní, jelikož budeme spořit pět let a úspory budeme vždy ukládat na začátku roku. Toto spoření se počítá podle tohoto vzorce $S' = a.(1+i).\frac{(1+i)^n-1}{i}$, kde S' je výsledná naspořená částka, n je počet úrokových období (let), ve kterých se spoří, i je roční úroková sazba a a je výše úložky, která je ukládána vždy na počátku úrokového období (roku). Právě výši této úložky potřebujeme v tomto příkladu spočítat.

Nejdříve si vypíšeme známé.

```
> S´:=750000; i:=0.025; n:=5;
```

$$S' := 750000$$

$$i := 0.025$$

$$n := 5$$

Nyní po dosazení do vzorce a následném vytknutí neznámé *a* pomocí maplovské funkce *fsolve,* dostaneme výsledek.

```
> S´:=a*(1+i)*(((1+i)^n)-1)/i;
```

$$S' := 5.387736732\ a$$

```
> fsolve(S´=750000, a);
```

$$1.392050201\ 10^5$$

Počátkem každého roku je třeba spořit 139 205,02 Kč.

Pozn.:

Bude-li úrokové období kratší než jeden rok, tj. úroky budou připisovány častěji, např. dvakrát ročně při pololetním úrokovém období, čtyřikrát ročně při čtvrtletním úrokovém období atd., bude nutno přizpůsobit úrokovou sazbu tomuto období.

Příklad 4.11 *Efektivní úroková sazba, složené úročení, bezprostřední důchod*

Zjistěte výši měsíčních plateb, které jsou potřeba k zaplacení závazku 10 000 Kč. Tento dluh budeme umořovat po dobu 25 let s úrokovou sazbou 10 % ročně. Nebudeme skládat žádnou zálohu. Jde o úročení složené polhůtní.

Řešení

Nejprve si spočteme, kolik měsíčních plateb budeme celkem platit (25 * 12 měsíců).

```
> with(finance):
> pocetplateb:=25*12;
```

$$pocetplateb := 300$$

Plateb bude celkem 300.

Jestliže roční úroková sazba je 10 %, potom půlroční sazba je 5 % (10 / 2). My však potřebujeme spočítat sazbu měsíční. Následně vypočteme pomocí příkazu *fsolve* a *effectiverate*.

```
> messaz:= fsolve(effectiverate(r,6)=0.05,r, 0.. 0.5)/6;
```

$$messaz{:=}0.008164846052$$

Tohle je tedy sazba, která vychází na jeden měsíc.

Nyní si uděláme malou zkoušku:

Budoucí hodnota z 1 Kč po šesti měsících se sazbou, kterou jsme si spočetli výše, by měla být 1,05. Spočítáme to podle základní rovnice na složené polhůtní úročení $FV = PV.(1+i)^t$. *PV* je počáteční kapitál, *i* úroková sazba a *t* je doba splatnosti.

Ukázka vzorce v Maplu.

```
> FV:=PV*(1+i)^t;
```

$$FV := PV\,(1+i)^t$$

Zadané veličiny.

```
> PV:=1; i:=messaz; t:=6;
```

$$PV := 1$$

$$i := .008164846052$$

$$t := 6$$

```
> FV;
```

$$1.050000000$$

K tomuto výpočtu též slouží funkce *futurevalue*.

```
> futurevalue(1., messaz, 6);
```

$$1.050000000$$

Dále si spočítáme hodnotu při vkládání 1 Kč měsíčně po 25 let s úrokovou sazbou, kterou jsme si už spočítali. Můžeme počítat podle vzorce na bezprostřední polhůtní důchod $D = a.\frac{(1+i)^n-1}{i.(1+i)^n} = a.\frac{1-v^n}{i}$. Zde *a* je výše vyplácené částky (1 Kč), *i* je úroková sazba a *n* je doba vyplácení (25 let). Vidíte i trochu zjednodušený vzorec, kde se objevuje proměnná *v*. Tahle proměnná se nazývá *diskontní faktor* a vypočítáme ji pomocí vzorečku $v = \frac{1}{1+i}$.

Druhá možnost jak danou hodnotu spočítat je použití maplovské funkce *annuity*.

```
> A:=annuity(1, messaz, pocetplateb);
```

$$A := 111.7958950$$

Teď už jen stačí vydělit výši dluhu hodnotou A a dostaneme velikost měsíční splátky, kterou jsme chtěli spočítat.

```
> P:=10000/A;
```

$$P := 89.44872260$$

Měsíční splátka vyšla necelých 90 Kč. Nyní si ještě můžeme spočítat, na kolik korun nás celkově přijde postupné splácení dluhu.

```
> Tot:=P*pocetplateb;
```

$$Tot:=26834.61678$$

Tady vidíme, že celkově se cena závazku vyšplhá skoro na 27 000 Kč.

<u>**Příklad 4.12**</u> *Bezprostřední důchod vyplácený vícekrát do roka*

Kolik budeme ochotni zaplatit za investici, z níž budeme mít ke konci každého čtvrtletí výnos 4 000 Kč po dobu dvaceti let, požadujeme-li míru výnosnosti 5 % p.a. a předpokládáme-li roční úrokové období?

Tato úloha jde zformulovat alternativně následujícím způsobem:

Jaká je počáteční hodnota důchodu 4 000 Kč, který se vyplácí na konci každého čtvrtletí po dobu dvaceti let při neměnné úrokové sazbě 5 %?

<u>Řešení</u>

Jelikož budou důchody vypláceny několikrát do roka, budeme počítat podle vzorce pro výpočet bezprostředního polhůtního důchodu vypláceného m-krát do roka

, kde m je četnost vyplácení důchodu ročně, x splátka důchodu v Kč, v diskontní faktor, který jsme zmínili v předchozím příkladě, a n představuje dobu vyplácení důchodu.

Výpis zadaných veličin v Maplu.

```
> x:=4000; m:=4; n:=20; i:=0.05;
```

$$x := 4000$$

$$m := 4$$

$$n := 20$$

$$i := 0.05$$

Z těchto údajů si dále vypočítáme diskontní faktor .

```
> v:=1/(1+i);
```

$$v := 0.9523809524$$

Nyní už máme všechny potřebné veličiny pro dosazení do vzorce.

```
> Dcelkove:=m*x*(1+((m-1)/(2*m))*i)*(1-v^n)/i;
```

$$Dcelkove := 2.031340286\ 10^5$$

Počáteční hodnota důchodu, tedy cena investice, je přibližně 203 134 Kč.

Příklad 4.13 *Budoucí hodnota, složené úročení*

Jaká byla roční úroková sazba z vkladu, jestliže částka 20 000 Kč vzrostla za čtyři roky na 23 400 Kč? Úroky byly vypláceny jedenkrát ročně, ponechány na účtu a dále úročeny.

Řešení

Z úlohy je vidět, že jde o složené polhůtní úročení, které se počítá podle vzorce $FV = PV.(1+i)^t$ a v programu Maple pomocí funkce *futurevalue*. *FV* je konečný kapitál, *PV* počáteční kapitál, *i* úroková sazba a *t* je doba splatnosti. Jediná neznámá v tomto příkladě je úroková sazba, kterou hledáme.

```
> with(finance):
> FV:=futurevalue(20000, i, 4);
```

$$FV := 20000\,(1+i)^4$$

Nyní vytkneme a spočteme neznámou *i* pomocí funkce *fsolve*.

```
> fsolve(FV=23400, i, 0..1);
```

$$0.04003143349$$

Úroková sazba byla 4 % p.a.

Příklad 4.14 *Čistá roční výnosnost*

Jaké čisté roční výnosnosti dosáhne klient, jestliže uložil na počátku roku částku 100 000 Kč na šestiměsíční termínový vklad při 2 % úrokové sazbě p.a. a v polovině roku kapitál včetně vyplacených úroků znovu okamžitě uložil na šestiměsíční termínový vklad při 2,5 % úrokové sazbě p.a.? Úroky z vkladů podléhají dani z příjmů vybírané srážkou ve výši 15 %.

Všechny předcházející výpočty neuvažovaly se zdaněním. Počítali jsme tedy výši úroku (výnos) před zdaněním, neboli hrubý výnos. Úrokové výnosy ovšem podléhají zdanění. Jestliže od hrubého výnosu odečteme daň z příjmů, získáme čistý výnos, to znamená částku, kterou investor (věřitel) skutečně obdrží. **Hrubý výnos** (úrok) je vyjádřen vztahem $u = K.i.n$.

Z tohoto vztahu dostaneme **čistý výnos** odečtením daně z příjmu, který je $u_č = K_0.i.n - d.K_0.i.n = K_0.i.(1-d).n$, kde K_0 je počáteční kapitál, $u_č$ je čistý výnos (úrok), i je úroková sazba, d je daňová sazba a n je doba splatnosti vyjádřená v letech.

Součtem počáteční výše kapitálu a čistého výnosu získáme **čistou konečnou výši kapitálu**. Zapíšeme: ${}_{č}K_n = K_0 + u_č = K_0 + K_0.i.(1-d).n = K_0.[1 + i.(1-d).n]$, kde ${}_{č}K_n$ je čistá konečná výše kapitálu.

Nyní už nám jen zbývá vztah pro **čistou roční výnosnost** (úrokovou sazbu), který vypadá takto: $i_č = \frac{u_č}{K_0.n} = \frac{{}_{č}K_n - K_0}{K_0.n} = i.(1-d)$.

<u>**Řešení**</u>

Čistou výši kapitálu na konci prvního pololetí vypočteme podle vzorce na čistou konečnou výši kapitálu, kam dosadíme zadané veličiny:

```
> K0:=100000; i1:=0.02; n:=0.5; d:=0.15;
```

$$K0 := 100000$$

$$i1 := 0.02$$

$$n := 0.5$$

$$d := 0.15$$

```
> cKpul:=K0*(1+i1*(1-d)*n);
```

$$cKpul := 1.008500000\ 10^5$$

Hodnotu vkladu na konci roku vypočítáme podle stejného výrazu, kde však za počáteční kapitál dosadíme hodnotu vkladu na počátku druhého pololetí, která nám teď vyšla a úrokovou sazbu změníme na sazbu druhého pololetí.

```
> cK1:=Kpul*(1+i2*(1-d)*n);
```

$$cK1 := 1.019215312\ 10^5$$

Teď už jen zbývá dosadit do vzorce na čistou míru výnosu. Jen za *n* dosadíme 1, protože počítáme míru výnosu za celý rok.

```
> ic:=(cK1-K0)/(K0*1);
```

$$ic := 0.01921531200$$

Čistá výnosnost za rok vyšla 1,92 % p.a.

5. Závěr

Pro matematiku je program Maple velmi užitečný. Program Maple je dobrá pomůcka k vysvětlení ať ryze matematických nebo právě finančních pojmů. Může sloužit například jako pomůcka pro učitele finanční matematiky, kteří využívají v hodinách výpočetní techniku. Učitel by měl znát možnosti, jak studentům ukázat danou problematiku z jiného úhlu pohledu. Využití tohoto programu leckdy dává studentům možnost rychlejšího a snazšího řešení dané problematiky, pokud však mají určité teoretické znalosti, které pomocí počítače využijí ve svůj prospěch.

Tato práce je vhodná pro studenty vysokých, vyšších odborných či středních škol s ekonomickým zaměřením. Snadno se v ní však budou orientovat i ti, kteří si budou chtít doplnit v dnešní době nezbytné znalosti samostudiem.

Určitě se každý z nás někdy setká se zakládáním nějakého bankovního účtu, s rozhodováním zda uskutečnit či neuskutečnit danou investici, s rozložením jednotlivých plateb nějakého závazku nebo například se spořením peněz na náš důchod. Proto je v dnešní době a samozřejmě i pro budoucnost důležité, abychom se v oboru finanční matematiky alespoň trochu orientovali.

Program Maple nám k tomu hodně pomůže, protože je v něm spousta finančních pojmů a případů velice dobře vysvětlena. Mnoho finančních problémů jde v programu snadno vyřešit pomocí už předdefinovaných finančních funkcí, které nás ušetří od manuálního počítání.

Jedinou nevýhodou je určitě finanční stránka věci. Pořízení tohoto programu není zrovna levné.

Literatura

K. H. Heal: Maple V, Learning Guide

Macháček, O.: Finanční a pojistná matematika,

Nýdl,V., Kletečková, M.: Finanční a pojistná matematika, JU ZF, 2001,

Radová, J., Dvořák, P.: Finanční matematika pro každého,

Manuály ovládání a programování programu Maple 9.5 .

WWW stránky:

www.maplesoft.com

www.maplesoft.cz