Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií

ÚSTAV MATEMATIKY

MAPLE

Edita Kolářová

1 Úvod

Maple je programové prostředí vyvinuté na univerzitě ve Waterloo v Kanadě pro snažší používání matematiky. Jeho název je odvozen z anglické fráze **Ma**thematics **ple**asure – matematika potěšením. Je to program, který používáme v počítačových cvičeních v prvním ročníku pro usnadnění pochopení matematiky.

Proč právě Maple?

- Patří mezi programy, které na rozdíl od standardních programů pro numerické výpočty modelují matematické operace se symbolickými výrazy. Může být používán jednak interaktivně (jako zobecněná kalkulačka), jednak jako programovací jazyk.
- Grafy jsou velmi důležitou pomůckou pro znázornění a lepší pochopení řešení matematických problémů. V Maplu můžeme jednoduše kreslit grafy funkcí jedné i vice proměnných i funkcí daných parametricky anebo implicitně. Plochy se dají otáčet a tak se můžeme podívat na danou plochu z různých stran.
- Maple šetří čas. Můžeme ho použít na složitější numerické výpočty a získat tak čas na vysvětlení nových pojmů a soustředit se na důležitější aspekty výpočtu.
- Maple slouží také jako okamžitá kontrola počítání. Studenti spočítají příklad v sešitě a pro kontrolu také v Maplu.
- Maple slouží učiteli k demonstraci chyb studentů. Pokud student použije pro daný příklad špatný vzorec, nebo ten správný ale chybným způsobem, učitel může nechat nesprávný výsledek nakreslit nebo jinak předvést v Maplu a přesvědčit studenta o jeho chybě. Je to mnohem účinnější než jenom říct „Máte to špatně“, protože student vidí na vlastní oči, jaký nesmysl dostal jako výsledek.
- Maple nutí studenty používat přesné značení, dodržovat syntaxi a psát správně závorky.

Na našich počítačových cvičeních neučíme studenty programovat v Maple. Učíme je matematiku a pomáháme si přitom s Maple. Vlastně ho používáme pouze interaktivním způsobem jako hodně dobrou kalkulačku.

1.1 Uživatelské prostředí a rozhraní programu Maple

Maple používá grafické pracovní a uživatelské prostředí. Existují dvě verze prostředí, jedno se nazývá classic worksheet a zavoláme ho kliknutím na žlutou ikonu, podobně červená ikona skrývá Maple s tabulkou symbolů pro zjednodušení práce.

Příkaz se píše na příkazový řádek, který začíná znaky > a písmo je na příkazovém řádku červené. Nový řádek získáme buď automaticky (na konci souboru se po vykonání posledního příkazu vygeneruje další příkazový řádek), nebo jej můžeme vložit pomocí tlačíka > na panelu nástrojů. Každý příkaz musíme ukončit buď středníkem, nebo dvojtečkou, podle toho,jestli požadujeme vypsání provedení příkazu – výstupu – na obrazovku. Předpokládejme, že jsme řádek zakončili středníkem. Po stisknutí klávesy Enter se příkaz vykoná a získáme modrý text – maplovský výstup. Výstup může být numerický, symbolický a grafický. Eventuelně se může stát, že dostaneme jako výstup fialový nebo modrý text: chybové hlášení.

Následující příklad demonstruje možné výstupy v Maplu.

Příklad 1.1.

a) Vypočítejte obvod kruhu o poloměru $r = \frac{5}{2}$.

b) Vypočítejte $(x + y)^2$.

c) Nakreslete graf funkce $z = x^2 + y^2$.

Řešení: a)

```
> r:=5/2;
```

$$r := \frac{5}{2}$$

```
> obvod:=2*Pi*r;
```

$$obvod := 5\pi$$

```
> evalf(obvod);
```

$$15.70796327$$

b)

```
> expand((x+y)^2;
```

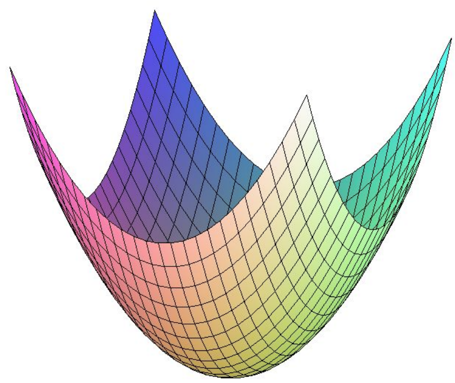
```
Error, `;` unexpected
```

```
> expand((x+y)^2);
```

$$x^2 + 2xy + y^2$$

c)

```
> plot3d(x^2+y^2,x=-2..2,y= -2..2);
```



Trojrozměný graf můžeme dokonce levým tlačítkem myši chytit a libovolně otáčet.

Program Maple pracuje se soubory typu .mws, resp. .mw, které kromě už uvedených vstupů (příkazových řádků) a výstupů mohou obsahovat i prostý text. Jednotlivé části souboru je možné sdružovat do kapitol. Možnosti programu Maple (a tím i vzhled panelu nástrojů) se liší podle toho, zda pracujeme s textem, příkazem nebo výstupem. V textovém řádku máme možnosti formátování podobné jako např. v programu Word.

1.2 Maple jako kalkulačka

Začínáme:

1. Pustíme si Maple 12, classic worksheet – žlutá ikona.
2. Po spuštění vypíše Maple ohlašovací znak > zvaný „prompt“ a umístí kurzor bezprostředně za ním.
3. Napíšeme maplovský výraz, který má být zpracován. Tento výraz musí mít přesně předepsaný, tzv. syntaktický tvar. Přípustné tvary postupně probereme. Jestliže nám na jeho zapsání nestačí řádek, stiskneme tlačítko ENTER nebo dopíšeme řádek až do konce. Maple reaguje promptem na dalším řádku a čeká na pokračování zápisu. Tak lze rozepsat výraz i na více řádků.

4. Výraz ukončíme středníkem a pokračujeme dalším maplovským výrazem. Stiskneme-li po středníku tlačítko ENTER, Maple zpracuje všechny předcházející výrazy (od posledního zpracování) a na obrazovce se objeví výsledky jejich zpracování. Chceme-li potlačit výpis některého výsledku na obrazovce, pak příslušný výraz ukončíme dvojtečkou místo středníku.

5. Po vykonání příkazu Maple vypíše prompt a čeká na další maplovský výraz.

Teď ukážeme jak používat Maple jako kalkulačku:

Za > napíšeme výraz, který chceme spočítat, ukončíme jej středníkem a stiskneme Enter. Objeví se výsledek. Jednotlivé algebraické operace zadáváme takto:

krát ... * , děleno ... /, na ... ˆ, místo desetinné čárky se píše tečka.

Všimněte si, že pokud jsou v zadaném výrazu pouze celá čísla, výsledek je zapsán symbolicky, ve tvaru zlomku, případně pomocí odmocnin, zatímco pro desetinná čísla je výsledek opět desetinné číslo. Srovnejte:

```
> (2*3-5)/3;
```

$$\frac{1}{3}$$

```
> 2^(1/2);
```

$$\sqrt{2}$$

```
> (2.0)^(1/2);
```

$$1.414213562$$

```
> 2^0.5;
```

$$1.414213562$$

Můžeme používat také mnoho různých funkcí:

$\sqrt{x}$ se zapisuje jako sqrt(x), $|x|$ se zadává jako abs(x), goniometrické funkce jako sin(x), cos(x), tan(x), cot(x) (tj. ne tg(x) a cotg(x)). Argument těchto funkcí se zadává v radiánech, ne ve stupních. Podobně ln(x) (přirozený logaritmus), log10(x) (dekadický logaritmus), log[a](x) (logaritmus o základu a) arcsin(x), arccos(x), arctan(x), arccot(x) a mnoho dalších.

Poznámka. e^x se zapisuje jako exp(x), ne eˆx . Pokud chcete pracovat s číslem e, můžete ho napsat jako exp(1), tj. e na prvou.

Argument funkce se vždy píše v závorce, např.:

```
> sqrt(16);
```

$$4$$

```
> sin(Pi/3);
```

$$\frac{\sqrt{3}}{2}$$

Maple zná konstantu π. Píše se Pi s velkým P.

Chceme-li s něčím, co jsme vypočetli, dále pracovat, je vhodné si to nějak pojmenovat. Přiřazení se provede takto: **identifikátor := výraz**, např.

```
> pes:=4;
```

$$pes := 4$$

```
> kocka:=5*1.2;
```

$$kocka := 6.0$$

Teď můžeme psa a kocku dále používat:

```
> c:=pes+kocka^2;
```

$$c := 40.00$$

```
> pes/8;
```

$$\frac{1}{2}$$

Pozor, Maple rozlišuje malá a velká písmena:

```
> A:=8;
```

$$A := 8$$

```
> a:=4;
```

$$a := 4$$

```
> a+A;
```

$$12$$

Některá písmena a slova nemůžeme použít jako identifikátory, Maple je má vyhrazena pro jiné účely. Nelze např. provést tohle:

```
> sin:=5;
```

```
Error, attempting to assign to 'sin' which is protected
```

nebo tohle (I je imaginární jednotka, ale malé i lze používat):

```
> I:=a*b;
```

```
Error, illegal use of an object as a name
```

Definujeme nyní výraz „nasobek“ jako součin p1 a p2. Změní-li se p1 nebo p2, změní se i „nasobek“:

```
> nasobek:=p1*p2;
```

$$nasobek := p1\ p2$$

```
> p1:=3;
```

$$p1 := 3$$

```
> nasobek;
```

$$3\, p2$$

```
> p2:=2;
```

$$p2 := 2$$

```
> nasobek;
```

$$6$$

```
> p1:=5;
```

$$p1 := 5$$

```
> nasobek;
```

$$10$$

Můžeme se vrátit k původním proměnným p1 a p2. Tím zařídíme, aby v p1 a p2 už nebyla konkrétní čísla. Uděláme to takhle:

```
> p1:='p1';
```

$$p1 := p1$$

```
> p2:='p2';
```

$$p2 := p2$$

```
> nasobek;
```

$$p1\ p2$$

Chceme-li do našeho „nasobku“ dosadit něco za p1 nebo p2, aniž by se změnila hodnota p1, p2, lze použít příkaz **subs** (substitute = dosadit), který má tvar

subs(co se dosazuje, kam se to dosazuje);

```
> subs(p1=10,nasobek);
```

$$10\ p2$$

```
> subs(p2=3*(aa+bb),nasobek);
```

$$p1\ (3\ aa + 3\ bb)$$

```
> subs(p1=3,p2=5,nasobek);
```

$$15$$

Dosadíme-li do výrazu za proměnnou nějaké číslo, Maple mnohdy výsledek zapíše symbolicky:

```
> y:=sin(3*x)+1;
> d:=subs(x=Pi,y);
```

$$y := \sin(3\,x) + 1$$

$$d := \sin(3\,\pi) + 1$$

Někdy je to vhodné, jindy by nás zajímala spíše numerická hodnota výsledku. Tu můžeme vypočítat pomocí příkazu evalf (evalf je zkrátkou výrazu evaluate = vyhodnotit) takto:

```
> evalf(d);
```

$$1.$$

```
> odmocnina_ze_dvou:=sqrt(2);
```

$$odmocnina\_ze\_dvou := \sqrt{2}$$

```
> evalf(odmocnina_ze_dvou);
```

$$1.414213562$$

```
> evalf(11/15);
```

$$0.7333333333$$

Výsledek předchozího příkazu lze použít pomocí symbolu %. Podobně symboly %% a %%% označují výsledky vyhodnocení předposledního a před - předposledního předcházejícího výrazu.

```
> 4*5-2*4;
```

$$12$$

```
> (%+1)*2;
```

$$26$$

Nechceme-li, aby se vypisovaly výsledky provedeného příkazu, ukončíme jej dvojtečkou a ne středníkem. Příkaz se provede, ale nic to nenapíše. Např.

```
> a:=4*5:
```

Přesvědčíme se, že se to opravdu provedlo:

```
> a;
```

$$20$$

Někdy je potlačit vypisování výsledků opravdu vhodné, zvlášť pokud jsou to mezivýsledky, které jsou dlouhé a zcela nezajímavé.

2 Diferenciální počet funkce jedné proměnné

2.1 Jak se zadává funkce a kreslí graf

Funkci $f : y = x^2 + 1$ můžeme zadat:

```
> f:=x->x^2+1;
```

$$f := x \to x^2 + 1$$

a pak její hodnoty v různých bodech počítat

```
> f(1);f(t);f(sqrt(x));f(Pi);
```

$$2$$
$$t^2+1$$
$$x+1$$
$$\pi^2+1$$

Numerickou hodnotu posledního výrazu vypočítáme pomocí příkazu **evalf**:

```
> evalf(f(Pi));
```

$$10.86960440$$

Nebo lze funkci zadat jako výraz, ale pak nelze počítat funkční hodnotu funkce pouhým dosazením do výrazu, ale musí se použít příkaz **subs**. Předvedeme to na příkladě funkce $g : y = x^2 - 1$:

```
> g:=x^2-1;
```

$$g := x^2 - 1$$

```
> g(2);
```

$$\mathrm{x}(2)^2 - 1$$

```
> subs(x=2,g);
```

$$3$$

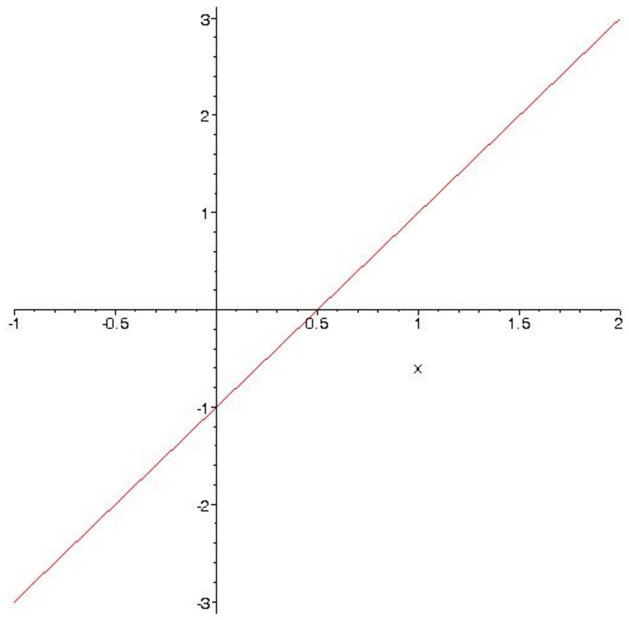
Grafy jsou velmi důležitou pomůckou pro znázornění a lepší pochopení řešení matematických problémů. Většina příkazů pro kreslení grafů je ve speciální knihovně plots. Popíšeme nejdříve používání Maplu bez knihovny plots.

Budeme používat nyní přímo zabudovaný příkaz plot, jehož základní tvar je:

plot(funkce, proměnná = dolní mez .. horní mez);

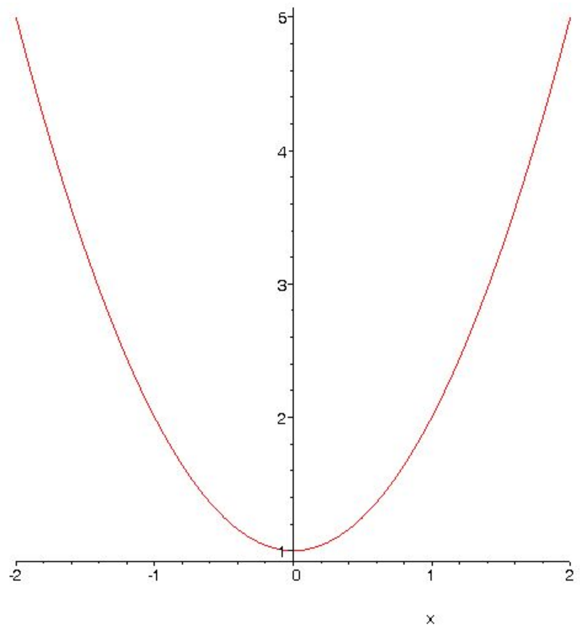
Pomocí následujícího příkazu se vykreslí graf funkce $y = 2\,x - 1$ v mezích od -1 do 2 :

```
> plot(2*x-1,x=-1..2);
```

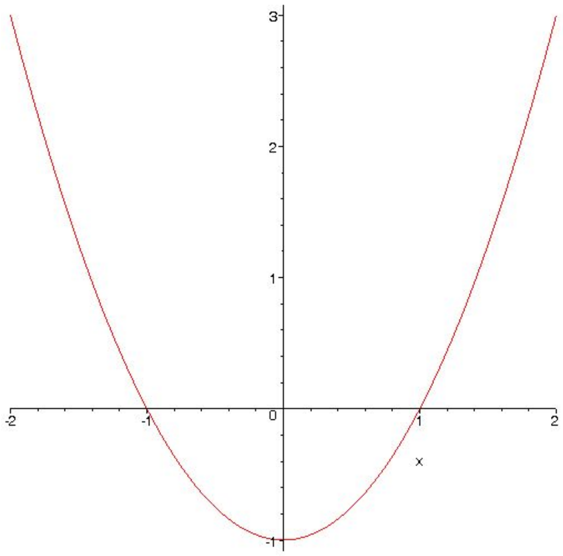


Můžeme také nakreslit graf funkce, která byla dříve zadána, např. v tomto souboru zadané funkce $\mathrm{f}(x) = x^2 + 1$ nebo funkce $g : y = x^2 - 1$:

```
> plot(f(x),x=-2..2);
```

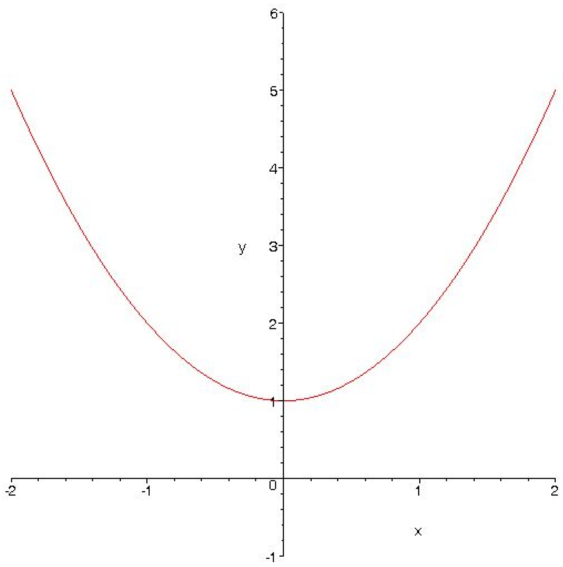


```
> plot(g,x=-2..2);
```



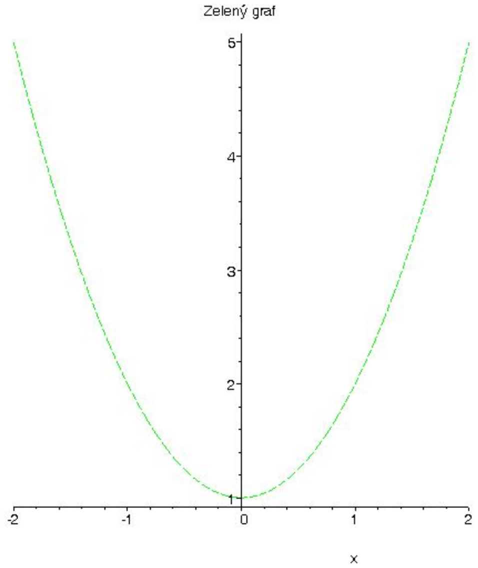
Osa y na prvním obrázku nezačíná nulou, ale jedničkou – proto graf vypadá, že prochází počátkem. Chceme-li se zbavit tohoto matoucího jevu, můžeme zadat i meze pro y:

```
> plot(f(x),x=-2..2,y=-1..6);
```



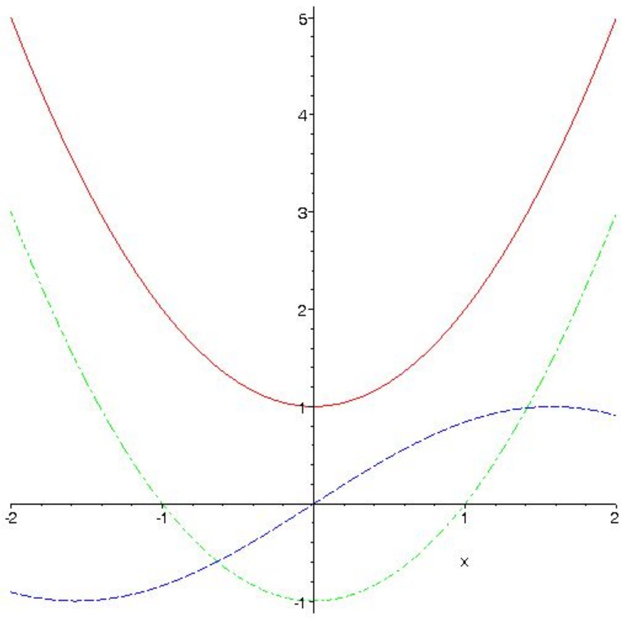
Při kreslení grafu můžeme použít mnoho různých parametrů, barvu, styl linky, nadpis, měřítko (**scaling=constrained** znamená, že na obou osách bude stejné měřítko):

```
> plot(f(x),x=-2..2,color=green,linestyle=3,title="Zelený graf");
```



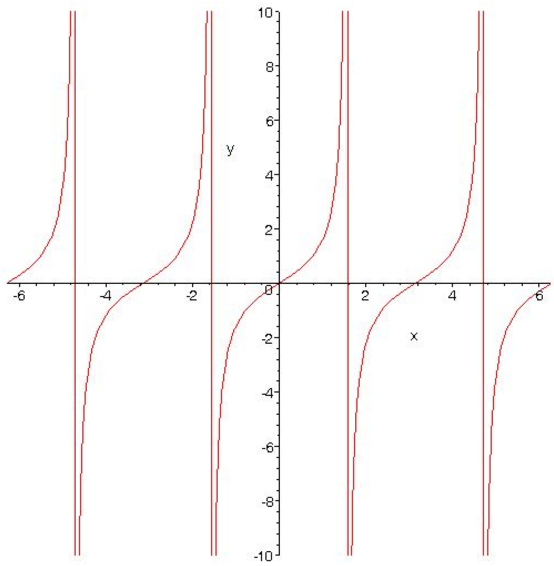
Více grafů v jednom obrázku se kreslí tak, že místo jediné funkce zadáme v příkazu plot seznam funkcí, jejichž grafy chceme kreslit, v hranatých závorkách:

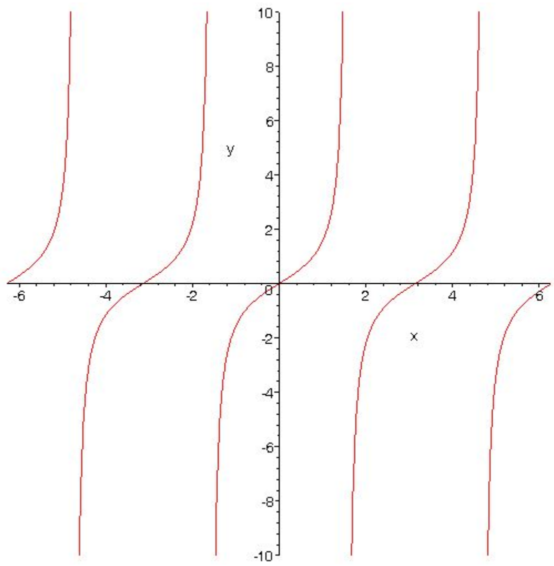
```
> plot([f(x),sin(x),g],x=-2..2,color=[red,blue,green],linestyle=[1,3,4]);
```



Maple kreslí graf funkce $y = f(x)$, $x \in (a, b)$ vždy jako spojitou křivku. V bodech nespojitosti, v nichž neexistuje limita, doplňuje graf vertikálními úsečkami. Má však i prostředek pro potlačení vertikálních úseček. Je to konstrukce tvaru **discont = true**, která se přidává v příkazu jako nepovinný parametr. Srovnejte

```
> plot(tan(x),x=-2*Pi..2*Pi,y=-10..10);
> plot(tan(x),x=-2*Pi..2*Pi,y=-10..10,discont=true);
```





Jak jsme už uvedli, Maple má speciální knihovnu plots, která obsahuje asi desítky různých příkazů. Knihovna se volá příkazem

with(plots):

Po jeho provedení můžeme kreslit obrázky funkci daných implicitně, parametricky, vícerozměrné a i celkem složité animace, viz úplně poslední animaci v těchto materiálech.

Příklad 2.1. Vypočítejte hodnoty funkce $f:\ y = 1 + \sin(2x + \frac{\pi}{2})$ v bodech $0, \pi, t, \frac{4}{2}\pi$ a nakreslete graf této funkce.

Příklad 2.2. Nakreslete graf funkce $f(x) = \frac{2x-1}{x+3}$ spolu s jejími asymptotami; graf funkce modře, asymptoty zeleně.

Příklad 2.3. Nakreslete do jednoho obrázku grafy následujících funkcí:

$$f:\ y = 2e^x - 3, \quad g:\ y = \ln\frac{x+3}{2}, \quad h:\ y = x.$$

Na základě tohoto obrázku rozhodnete, zda jsou funkce f a g navzájem inverzní.

Příklad 2.4. Nakreslete do jednoho obrázku grafy navzájem inverzních funkci $f(x) = \sin(x)$ a $g(x) = \arcsin(x)$; graf funkce f červeně, graf funkce g zeleně.

2.2 Počítání limit v Maplu

Příkaz k výpočtu limity funkce f v bodě má tvar

limit (f(x), x = a); limit (f(x), x = a, right); limit (f(x), x = a, left);

V případě, že příkazy začínají velkým písmenem L (místo malého l), výpočet limity se neprovede. Výsledek příkazu je pak pouze zápis limity v matematické symbolice.
Za příkazem Limit může ale následovat příkaz

value (%) ;

a výpočet limity, pokud existuje, se opět provede. Ve všech příkazech pro výpočet limity bod může být infinity (∞) anebo - infinity (- ∞). Uvedeme si příklady.

```
> Limit(sin(x)/x,x=0);
```

$$\lim_{x\to 0}\frac{\sin(x)}{x}$$

```
> value(%);
```

$$1$$

```
> Limit(sin(x)/x,x=0)=limit(sin(x)/x,x=0);
```

$$\lim_{x \to 0} \frac{\sin(x)}{x} = 1$$

```
> Limit(1/x,x=0)=limit(1/x,x=0);
> Limit(1/x,x=0,right)=limit(1/x,x=0,right);
> Limit(1/x,x=0,left)=limit(1/x,x=0,left);
```

$$\lim_{x \to 0} \frac{1}{x} = \mathit{undefined}$$

$$\lim_{x \to 0+} \frac{1}{x} = \infty$$

$$\lim_{x \to 0-} \frac{1}{x} = -\infty$$

```
> Limit((x+1)^(2/3)-(x-1)^(2/3),x=infinity)=limit((x+1)^(2/3)-(x-1)^(2/
> 3),x=infinity);
```

$$\lim_{x \to \infty} (x+1)^{(2/3)} - (x-1)^{(2/3)} = 0$$

```
> Limit(sin(5*x)/sin(2*x),x=0)=limit(sin(5*x)/sin(2*x),x=0);
```

$$\lim_{x \to 0} \frac{\sin(5\,x)}{\sin(2\,x)} = \frac{5}{2}$$

```
> f:=x->x^2; limit(f(x),x=2);
```

$$f := x \to x^2$$

$$4$$

Ne vždy je výsledek správně. Pokud limita existuje v daném bodě, výpočet limity se provede v pořádku. Pokud ale limita v daném bodě neexistuje, může Maple vypsat i takový nesmyslný výsledek:

```
> Limit(sin(x)+cos(x),x=infinity)=limit(sin(x)+cos(x),x=infinity);
```

$$\lim_{x \to \infty} \sin(x) + \cos(x) = -2..2$$

Příklad 2.5. Vypočítejte v Maplu $\lim\limits_{x \to 2^-} \dfrac{3-2x^2}{x-2}$ a $\lim\limits_{x \to \infty} \dfrac{5x^2-7x+3}{2x-1}$.

2.3 Výpočet derivace v Maplu

Příkazy pro derivaci funkce $f(x)$ v bodě x v Maplu jsou

diff (f(x), x); diff (f(x), x$n);

které umožňnují spočítat první a n-tou derivaci dané funkce a matematicky tyto derivace zapsat. V případě, že příkazy začínají velkým písmenem **D** místo malého písmene **d**, výpočet derivace se neprovede. Výsledkem příkazu je pak pouze zápis derivace v matematické symbolice (používané se symbolem pro derivaci vzhledem k proměnné x) . Za příkazem Diff však může následovat podobně jako u limit příkaz **value(%)** a výpočet derivace se opět provede.

```
> diff(x*sin(x),x);
```

$$\sin(x) + x\cos(x)$$

```
> Diff(x*sin(x),x);
```

$$\frac{d}{dx}\,(x\sin(x))$$

```
> Diff(x*sin(x),x)=diff(x*sin(x),x);
```

$$\frac{d}{dx}\,(x\sin(x)) = \sin(x) + x\cos(x)$$

Funkci můžeme předem zadat jako maplovskou funkci anebo maplovský výraz:

```
> f:=x->exp(2*x);
> derivace_f:=diff(f(x),x);
```

$$f := x \to e^{(2\,x)}$$

$$\mathit{derivace\_f} := 2\,e^{(2\,x)}$$

```
> g:=sin(x/3);
> derivace_g:=diff(g,x);
```

$$g := \sin(\frac{x}{3})$$

$$\mathit{derivace\_g} := \frac{1}{3}\cos(\frac{x}{3})$$

Výsledkem příkazu **diff** je výraz, ne funkce. Proto chceme-li vypočítat hodnotu derivace v konkrétním bodě, nelze to udělat takto

```
> derivace_f(0);
```

$$2\,(e^{(2\,x)})(0)$$

ale musí se dosadit pomocí příkazu **subs**

```
> subs(x=0,derivace_f);
```

$$2\,e^0$$

```
> simplify(%);
```

$$2$$

```
> Diff(x^3,x$2)=diff(x^3,x$2);
```

$$\frac{d^2}{dx^2}\,(x^3) = 6\,x$$

Použili jsme velice užitečný příkaz pro zjednodušení předchozího výsledku

simplify(%);

Maple sice automaticky krátí zlomky, ne však automaticky zjednodušuje všechny výrazy. K tomu slouží příkaz simplify, kde do závorky můžeme napsat přimo výraz anebo jedno či více %, podle toho, jestli potrebujeme zjednodušit předchozí, předpředchozí, předpředpředchozí, ...výsledek.

Jiný příkaz pro výpočet první a n-té derivace funkce je příkaz tvaru

D(jméno funkce); (D@@n)(jméno funkce);

Nevýhodou tohoto příkazu je, že funkce musí být zadána opravdu jako maplovska funkce, ne jako výraz. Na druhé straně, velkou výhodou tohoto příkazu je že výsledek je opět funkce a můžeme rovnou počítat hodnoty derivace v různých bodech.

```
> f:=x->1/x^2;
> D(f);
```

$$f := x \to \frac{1}{x^2}$$

$$x \to -\frac{2}{x^3}$$

Takto ne:

```
> D(1/x^2);
```

$$-\frac{2\,\mathrm{D}(x)}{x^3}$$

V případě maplovské funkce všechno krásně funguje:

```
> f:=x->x^2;
> df:=D(f);
> print(`Hodnota derivace pro x=1 je `);
> df(1);
> print(`Hodnota derivace pro x=2 je`);
> df(2);
```

$$f := x \to x^2$$
$$df := x \to 2\,x$$
$$\mathit{Hodnota\ derivace\ pro\ x = 1\ je}$$
$$2$$
$$\mathit{Hodnota\ derivace\ pro\ x = 2\ je}$$
$$4$$

```
> f:=x->x^4;
> (D@@3)(f);
```

$$f := x \to x^4$$
$$x \to 24\,x$$

```
> (D@@3)(f)(-1);
```

$$-24$$

Příklad 2.6. Vypočtěte derivaci funkce

$$f(x) = \sqrt{x^3 + 1}$$

a hodnoty této derivace pro $x = 0, x = 1, x = 2$.

Příklad 2.7. Vypočtěte druhou derivaci funkce $f(x) = x^2 e^x$ a hodnotu této derivace v bodě $x = 0$.

Příklad 2.8. Nakreslete do jednoho obrázku graf funkce

$$f(x) = \frac{4x + 5}{2x + 1}$$

a tečnu ke grafu této funkce v bodě $T = [-1, ?]$. Pro počítání tečny využijte Maple.

2.4 Průběh funkce

Proces vyšetřování průběhu funkce ukážeme na příkladě funkce $f(x) = \dfrac{1}{x^2 - 6x + 8}$. Postupujeme podle následujících bodů:

1. Definiční obor, nulové body funkce. Začněme informacemi, které lze o funkci získat z jejího předpisu. Při hledání nulových bodů funkce využijeme příkaz na řešení rovnic a nerovnic.

solve(rovnice); solve(nerovnice);

```
> y:=1/(x^2-6*x+8);
```

$$y := \frac{1}{x^2 - 6x + 8}$$

```
> nulove_body:=solve(y=0);
```

$$nulove\_body :=$$

```
> body_nespojitosti:=solve(1/y=0);
```

$$body\_nespojitosti := 4,\ 2$$

Vidíme tedy, že funkce je definována pro $x \neq 2, 4$; nulové body nemá.

2. Funkce rostoucí, klesající, stacionární body. Další informace o funkci nám poskytne její první derivace. Vzhledem k tomu, že Maple nemusí derivaci určit v přehledném tvaru, rovnou použijeme příkaz **simplify**. Poté určíme, kde je funkce rostoucí, kde klesající a kde má stacionární bod.

```
> prvni_derivace:=simplify(diff(y,x));
> rostouci:=solve(prvni_derivace>0);
> klesajici:=solve(prvni_derivace<0);
> stacionarni_body:=solve(prvni_derivace=0);
```

$$prvni\_derivace := -\frac{2\,(x-3)}{(x^2 - 6\,x + 8)^2}$$

$$rostouci := \mathrm{RealRange}(-\infty,\ \mathrm{Open}(2)),\ \mathrm{RealRange}(\mathrm{Open}(2),\ \mathrm{Open}(3))$$

$$klesajici := \mathrm{RealRange}(\mathrm{Open}(3),\ \mathrm{Open}(4)),\ \mathrm{RealRange}(\mathrm{Open}(4),\ \infty)$$

$$stacionarni\_body := 3$$

Musíme ovšem rozlišovat, které intervaly patří do definičního oboru dané funkce. V tomto případě vidíme, že funkce roste na intervalu $(-\infty,\ 2)$ a $(2,\ 3)$, klesá na intervalech $(3,\ 4)$ a $(4,\ \infty)$ a stacionárním bodem je bod $x = 3$.

3. Konkávnost, konvexnost funkce, inflexní body. Konkávnost a konvexnost funkce určíme z druhé derivace. Postupujeme analogicky, jako v 2. bodě.

```
> druha_derivace:=simplify(diff(prvni_derivace,x));
```

$$\mathit{druha\_derivace} := \frac{2\,(3\,x^2 - 18\,x + 28)}{(x^2 - 6\,x + 8)^3}$$

```
> konvexni:=solve(druha_derivace>0);
> konkavni:=solve(druha_derivace<0);
> inflexni_body:=solve(druha_derivace=0);
```

$$\mathit{konvexni} := \mathrm{RealRange}(-\infty,\ \mathrm{Open}(2)),\ \mathrm{RealRange}(\mathrm{Open}(4),\ \infty)$$

$$\mathit{konkavni} := \mathrm{RealRange}(\mathrm{Open}(2),\ \mathrm{Open}(4))$$

$$\mathit{inflexni\_body} := 3 + \frac{1}{3}\,I\,\sqrt{3},\ 3 - \frac{1}{3}\,I\,\sqrt{3}$$

Podobně jako u prvních derivací musíme rozlišovat, které intervaly patří do definičního oboru funkce. Zde vidíme, že funkce je konvexní (tj. nad tečnou) na intervalu $(-\infty, 2)$ a $(4, \infty)$ a konkávní (tj. pod tečnou) na intervalu $(2, 4)$. Funkce nemá reálné inflexní body (I značí imaginární jednotku).

4. Lokální extrémy. Pomocí hodnot druhé derivace můžeme také rozhodnout o typech lokálních extrémů funkce. Víme o nich, že mohou nastat ve stacionárních bodech, o jejich charakteru pak rozhoduje znaménko druhé derivace v těchto bodech. Proto:

```
> s1:=stacionarni_body:
druha_derivace_v_s1:=subs(x=s1,druha_derivace);
```

$$\mathit{druha\_derivace\_v\_s1} := -2$$

5. Asymptoty grafu funkce. Asymptoty bez směrnice (svislé) budeme hledat v bodech nespojitosti.Proto budeme zkoumat jednostranné limity v těchto bodech.

```
> Limit(y,x=2,right)=limit(y,x=2,right);
Limit(y,x=2,left)=limit(y,x=2,left);
Limit(y,x=4,right)=limit(y,x=4,right);
Limit(y,x=4,left)=limit(y,x=4,left);
```

$$\lim_{x \to 2+} \frac{1}{x^2 - 6\,x + 8} = -\infty$$

$$\lim_{x \to 2-} \frac{1}{x^2 - 6\,x + 8} = \infty$$

$$\lim_{x \to 4+} \frac{1}{x^2 - 6\,x + 8} = \infty$$

$$\lim_{x \to 4-} \frac{1}{x^2 - 6\,x + 8} = -\infty$$

Kromě svislých asymptot mohou existovat i asymptoty se směrnicí.

```
> Limit(y/x,x=infinity)=limit(y/x,x=infinity);
k1:=limit(y/x,x=infinity):
Limit(y-k1*x,x=infinity)=limit(y-k1*x,x=infinity);
q1:=limit(y-k1*x,x=infinity): asymptota_1:=k1*x+q1;
```

$$\lim_{x\to\infty} \frac{1}{(x^2-6\,x+8)\,x} = 0$$

$$\lim_{x\to\infty} \frac{1}{x^2-6\,x+8} = 0$$

$$asymptota\_1 := 0$$

V tomto případě je jedinou asymptotou se směrnicí přímka $y = 0$, tj. osa x.

6. Vykreslení grafu funkce. Nyní můžeme graf funkce nechat vykreslit. Nejprve si však zopakujme, co o dané funkci víme:

– není definována v bodech 2 a 4,

– na intervalech $(-\infty, 2)$ a $(2, 3)$ roste, na intervalech $(3,4)$ a $(4, \infty)$ klesá,

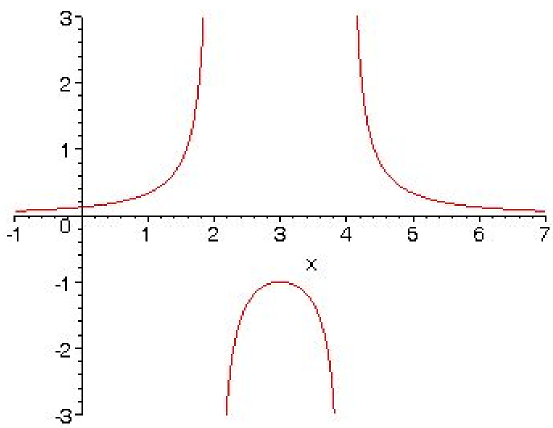
– v bodě $x = 3$ má lokální maximum $y = -2$,

– na intervalech $(-\infty, 2)$ a $(4, \infty)$ je konvexní, na intervalu $(2, 4)$ je konkávní, nemá inflexní bod,

– v okolí bodu $x = 2$ nabývá zprava velkých záporných hodnot, zleva velkých kladných hodnot; v okolí bodu $x = 4$ zprava velkých kladných, zleva velkých záporných hodnot,

– osa x je asymptotou grafu funkce.

```
> plot(y,x=-1..7,-3..3,scaling=constrained,discont=true);
```



3 Integrální počet funkce jedné proměnné

3.1 Neurčitý integrál

Příkaz pro výpočet neurčitého integrálu (primitivní funkce) z dané funkce má tvar

int(funkce, integrační proměnná);

V případě, že příkazy začínají velkým písmenem I (místo malého i), provede se pouze matematický přepis dané úlohy bez výpočtu integrálu. Na rozdíl od obecného zápisu užívaného v integrálním počtu jazyk Maple vypouští konstantu.
Maple při provádění příkazu **int** zkusí integrál spočítat. Na rozdíl od výpočtu derivace není zaručena úspěšnost výpočtu. Může se stát, že výsledek určený Maplem je komplikovanější, než bychom očekávali, nebo Maple odpoví stejným nebo zjednodušeným výrazem, který mu byl zadán k integraci.

```
> Int(x^n,x)=int(x^n,x);
```

$$\int x^n\,dx = \frac{x^{(n+1)}}{n+1}$$

```
> Int(x^n,n)=int(x^n,n);
```

$$\int x^n\,dn = \frac{x^n}{\ln(x)}$$

```
> Int(ln(x),x)=int(ln(x),x);
```

$$\int \ln(x)\,dx = x\ln(x) - x$$

Pro výpočet integrálů per-partes slouží v knihovně **student** příkaz

intparts(Int(u*dv, x), u);

Tento příkaz umožní studentovi procvičit si volbu funkci u a dv v metodě per partes a zjistit, co se stane, když si zvolí funkce opačně. Výsledek svého výpočtu si může ověřit v Maplu.

```
> int(x^2*cos(x),x); with(student):
Int(x^2*cos(x),x)=intparts(Int(x^2*cos(x),x),x^2);
```

$$x^2\sin(x) - 2\sin(x) + 2\,x\cos(x)$$

$$\int x^2\cos(x)\,dx = x^2\sin(x) - \int 2\,x\sin(x)\,dx$$

3.2 Integrál z racionální lomené funkce

Integrace racionální lomené funkce se převede na integraci parciálních zlomků. Rozklad na parciální zlomky je poměrně pracná záležitost a nesouvisí přímo s inegrováním jako takovým, a proto je užitečné na tento úkon použít Maple.

```
> u:=(x^2-2*x-8)*(x^2-2*x+1);
```

$$u := (x^2 - 2\,x - 8)\,(x^2 - 2\,x + 1)$$

```
> factor(u);
```

$$(x + 2)\,(x - 4)\,(x - 1)^2$$

```
> v:=(4*x+8)/u;
```

$$v := \frac{4\,x + 8}{(x^2 - 2\,x - 8)\,(x^2 - 2\,x + 1)}$$

```
> convert(v,parfrac,x);
```

$$\frac{4}{9\,(x - 4)} - \frac{4}{9\,(x - 1)} - \frac{4}{3\,(x - 1)^2}$$

```
> int(v,x);
```

$$\frac{4}{3\,(x - 1)} + \frac{4}{9}\ln(x - 4) - \frac{4}{9}\ln(x - 1)$$

Příklad 3.1. Vypočtěte následující integrály na papíře i v Maplu:

$$\int \frac{(e^x - 10)^2}{e^{3x}}\mathrm{d}x, \qquad \int \frac{\sin x}{3 - 5\cos x}\mathrm{d}x, \qquad \int e^{x^3} x^2 \mathrm{d}x.$$

Příklad 3.2. Metodou per partes vypočtěte následující integrály:

$$\int 6x\, e^{3x}\mathrm{d}x, \qquad \int (2x - 5)\,\sin x\,\mathrm{d}x, \qquad \int \arcsin 3x\,\mathrm{d}x.$$

Příklad 3.3. Vypočtěte následující integrály rozkladem na parciální zlomky:

$$\int \frac{24x}{9x^3 - 6x^2 - 5x + 2}\mathrm{d}x, \quad \int \frac{1}{x^3 - 8}\mathrm{d}x, \quad \int \frac{x^2 + 3x + 2}{(1 + x^2)^2(x^2 - 4)}\mathrm{d}x.$$

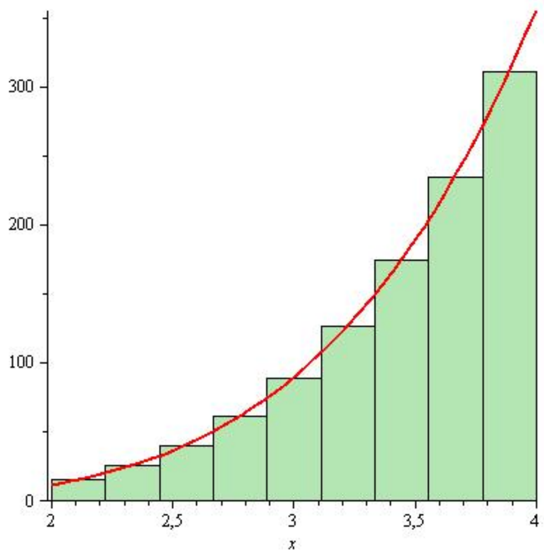
3.3 Určitý integrál

Víme, že určitý integrál je limitou vytvořujících součtů pro normu dělení jdoucí k nule. Pokusíme se o reprezentaci těchto součtů u kladné spojité funkce $f = x^4 \ln x$ pomocí animace.

```
> with(student):f:=x^4*ln(x);
```

$$f := x^4 \ln(x)$$

```
> S:=seq(middlebox(f,x=2..4,n),n=delení):with(plots):display(S,insequen
> ce=true);
```



Můžete spustit animaci přímo v Maplu kliknutím sem.

Příkaz pro výpočet určitého integrálu má tvar

int(funkce, promenna=dolní mez .. horní mez);

I tady platí, že příkazy začínají velkým písmenem I (místo malého i), provedou pouze matematický přepis dané úlohy.

```
> Int(f,x=2..4)=int(f,x=2..4);evalf(%);
```

$$\int_2^4 x^4 \ln(x)\, dx = \frac{2016}{5} \ln(2) - \frac{992}{25}$$

$$239.7969432$$

Maple umí počítat i nevlastní integrály v případě, kdy funkce je v hraničním bodě integračního intervalu neohraničená, ale jsou splněny podmínky konvergence integrálu na uzavřeném intervalu.

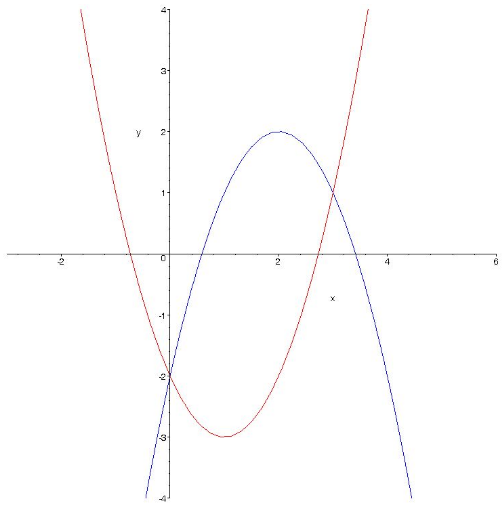
3.4 Výpočet obsahu plochy

Pro výpočet obsahu plochy nelze vždy pouze mechanicky použít určitého integrálu. Nejlepší způsob, jak se vyhnou chybě je nakreslit graf a podle toho určit meze určitého integrálu.

```
> F1:=x^2-2*x-2;
> G1:=-x^2+4*x-2;
```

$$F1 := x^2 - 2\,x - 2$$
$$G1 := -x^2 + 4\,x - 2$$

```
> plot([F1,G1],x=-3..6,y=-4..4,color=[red,blue]);
```




```
> int(F1-G1,x=0..3);
```

$$-9$$

Hodnota integrálu je záporná – je třeba odečíst funkci G od F:

```
> int(G1-F1,x=0..3);
```

$$9$$

Příklad 3.4. Vypočtěte následující integrály v Maplu:

$$\int_0^{\frac{\sqrt{2}}{2}} \frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x, \qquad \int_1^e \ln x\,\mathrm{d}x, \qquad \int_0^1 x\,e^{-x}\,\mathrm{d}x, \qquad \int_0^{\sqrt{e}} \frac{1}{x\sqrt{1-\ln^2 x}}\,\mathrm{d}x.$$

Příklad 3.5. Vypočtěte obsah oblasti omezených čárami o rovnicích:

a) $y = 6x - x^2,\ y = x;$ b) $y = 14 + 5x - x^2,\ y = x^2 - x - 6;$ c)$y = \ln x,\ y = \ln^2 x.$

4 Řady

4.1 Číselné řady

Konečné i nekonečné součty číselné řady se v Maple sčítají příkazem:

sum (výraz, sčítací index = dolní mez .. horní mez);

Jejich formální, matematický zápis se provádí příkazem **Sum**.

```
> sum(1/n^2,n=1..15);
```

$$\frac{205234915681}{129859329600}$$

```
> Sum(1/n^2,n=1..15);
```

$$\sum_{n=1}^{15} \frac{1}{n^2}$$

```
> Sum(1/n^2,n=1..infinity)=sum(1/n^2,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

```
> Sum(1/n^3,n=1..infinity)=sum(1/n^3,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{n^3} = \zeta(3)$$

Výsledek ve formě desetinného čísla lze získat příkazem **evalf**.

```
> Sum(1/n^3,n=1..infinity)=evalf(sum(1/n^3,n=1..infinity));
```

$$\sum_{n=1}^{\infty} \frac{1}{n^3} = 1.202056903$$

```
> Sum(1/n,n=1..infinity)=sum(1/n,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty$$

Pokud Maple neumí nekonečný součet vypočítat, místo výsledku opíše zadání

```
> Sum((-1)^n,n=1..infinity)=sum((-1)^n,n=1..infinity);
```

$$\sum_{n=1}^{\infty} (-1)^n = \sum_{n=1}^{\infty} (-1)^n$$

Příklad 4.1.

a) Vypočtěte v Maplu součet $\sum_{n=1}^{N} \frac{1}{n}$ pro N = 10, 100, 1000 , ... (kolik chcete). Výsledky nechte psát ve tvaru desetinného čísla, ne zlomku.

b) Udělejte něco podobného pro $\sum_{n=0}^{N} \frac{1}{n!}$, (zde radši nevolte N moc velké).

c) $\displaystyle\sum_{n=1}^{\infty} \frac{1}{n}$, d) $\displaystyle\sum_{n=0}^{\infty} \frac{1}{n!}$, e) $\displaystyle\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$, f) $\displaystyle\sum_{n=1}^{\infty} \frac{(-1)^n}{n!}$.

4.2 Taylorův rozvoj funkce

Taylorův rozvoj funkce v daném bodě x_0 se provádí příkazem

taylor(funkce, x $= x_0$**);**

Není-li uvedeno jinak, rozvoj se provede do polynomu stupně < 6 s chybou 6. řádu.

```
> sin(x)=taylor(sin(x),x=0);
```

$$\sin(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5 + O(x^6)$$

Aby chyba v Taylorově rozvoji byla řádu n, je potřeba toto n v příkazu **taylor** specifikovat jako poslední parametr.

```
> ln(1+x)=taylor(ln(1+x),x=0,9);
```

$$\ln(x+1) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 - \frac{1}{6}x^6 + \frac{1}{7}x^7 - \frac{1}{8}x^8 + O(x^9)$$

```
> 1/x=taylor(1/x,x=2,4);
```

$$\frac{1}{x} = \frac{1}{2} - \frac{1}{4}(x-2) + \frac{1}{8}(x-2)^2 - \frac{1}{16}(x-2)^3 + O((x-2)^4)$$

Příkazem mtaylor se přímo vypočítá Taylorův aproximační polynom funkce $f(x)$. Zbytek je automaticky potlačen. Příkaz má tvar

mtaylor (funkce, x = a);

Výsledek tohoto příkazu je výraz, se kterým lze dále pracovat – počítat jeho hodnotu v nějakém bodě, kreslit graf a podobně.

```
> skorosin:=mtaylor(sin(x),x=0);
```

$$\mathit{skorosin} := x - \frac{1}{6}x^3 + \frac{1}{120}x^5$$

Odchylka funkcí sin a skorosin v bodě $\pi/2$.

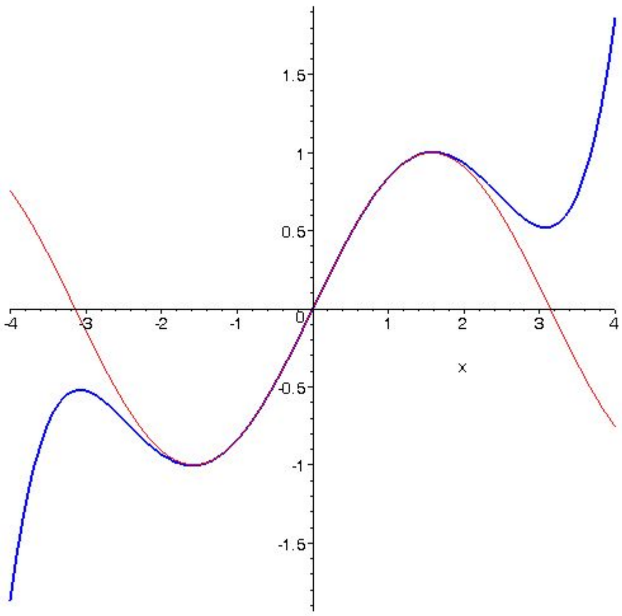
```
> sin(Pi/2)-subs(x=Pi/2,skorosin);
```

$$1 - \frac{1}{2}\pi + \frac{1}{48}\pi^3 - \frac{1}{3840}\pi^5$$

```
> evalf(%);
```

$$-0.00452485571$$

```
> plot([sin(x),skorosin],x=-4..4,color=[red,blue],thickness=[1,2]);
```



4.3 Fourierovy řady

Fourierův rozvoj funkce Maple přímo neobsahuje, takže všechny vzorce je potřeba zadat buď do nějaké vlastní procedury, kterou by pak bylo možné volat, nebo do několika příkazů. Jako příklad uvedeme rozvoj funkce $x^2 - 2\,x$ na intervalu $\langle -\pi, \pi \rangle$.

```
> f:=x->x^2-2*x;
> a:=k->1/Pi*int(f(x)*cos(k*x),x=-Pi..Pi);
> b:=k->1/Pi*int(f(x)*sin(k*x),x=-Pi..Pi);
> Fourier_f:=n->a(0)/2+sum(a(k)*cos(k*x)+b(k)*sin(k*x),k=1..n);
```

$$f := x \to x^2 - 2x$$

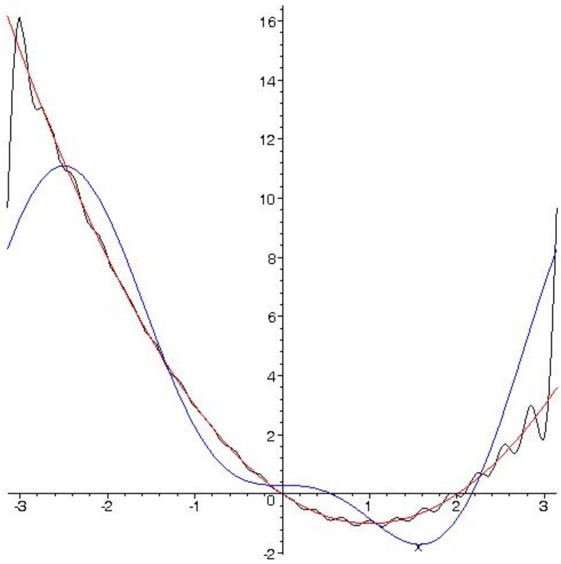
$$a := k \to \frac{1}{\pi} \int_{-\pi}^{\pi} \mathrm{f}(x) \cos(k\,x)\,dx$$

$$b := k \to \frac{1}{\pi} \int_{-\pi}^{\pi} \mathrm{f}(x) \sin(k\,x)\,dx$$

$$\mathit{Fourier\_f} := n \to \frac{1}{2}\,\mathrm{a}(0) + \sum_{k=1}^{n} (\mathrm{a}(k) \cos(k\,x) + \mathrm{b}(k) \sin(k\,x))$$

Grafické srovnání původní funkce (červeně) a jejích rozvojů do druhých (modře), resp. dvacátých (černě) harmonických složek:

```
> plot([f(x),Fourier_f(2),Fourier_f(20)],x=-Pi..Pi,color=[red,blue,black]);
```



Dle naši definice **Fourier_f(n)** je Fourierův rozvoj funkce f do n složek, např:

```
> Fourier_f(3);
```

$$\frac{\pi^2}{3} - 4\cos(x) - 4\sin(x) + \cos(2\,x) + 2\sin(2\,x) - \frac{4}{9}\cos(3\,x) - \frac{4}{3}\sin(3\,x)$$

Příklad rozvoje na jiném intervalu, než $\langle -\pi, \pi \rangle$, obecně by to mohl být jakýkoli interval. Budeme rozvíjet funkci sign(x) na intervalu $\langle 0, 1 \rangle$. Pro stručnost jsme potlačili vypisování posledních řádků.

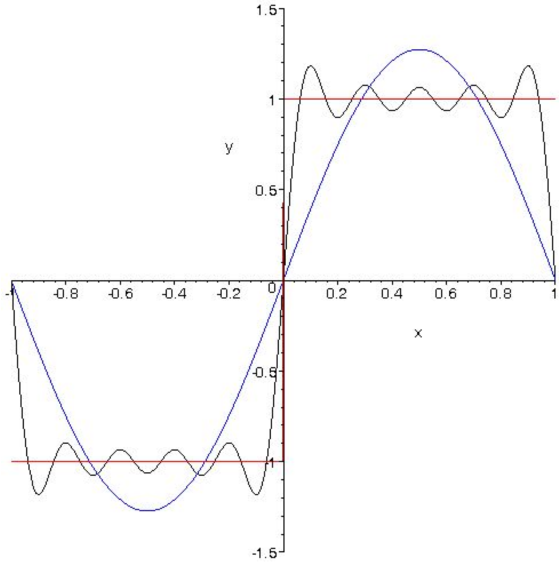
```
> f:=x->abs(x)/x;
> d:=-1;h:=1;
> T:=h-d:w:=2*Pi/T:
> a:=k->2/T*int(f(x)*cos(k*w*x),x=d..h):
> b:=k->2/T*int(f(x)*sin(k*w*x),x=d..h):
> Fourier_f:=n->a(0)/2+sum(a(k)*cos(k*w*x)+b(k)*sin(k*w*x),k=1..n):
```

$$f := x \to \frac{|x|}{x}$$

$$d := -1$$
$$h := 1$$

Nakreslime si graf funkce $\operatorname{sign}(x)$ a Fourierova rozvoje do druhých a desátých harmonických složek:

```
> plot([f(x),Fourier_f(2),Fourier_f(10)],x=-1..1,y=-1.5..1.5,color=[red
,blue,black]);
```




```
> Fourier_f(10);
```

$$\frac{4\sin(\pi\,x)}{\pi}+\frac{4}{3}\,\frac{\sin(3\,\pi\,x)}{\pi}+\frac{4}{5}\,\frac{\sin(5\,\pi\,x)}{\pi}+\frac{4}{7}\,\frac{\sin(7\,\pi\,x)}{\pi}+\frac{4}{9}\,\frac{\sin(9\,\pi\,x)}{\pi}$$

Vyzkoušejte si animaci v Maplu, která znázorňuje jak se s přičítáním dalších a dalších harmonických složek zlepšuje aproximace, kliknutím sem.

Příklad 4.2. Vypočtěte Taylorův polynom 4. řádu funkce $f(x) = \dfrac{2x}{x-3}$ v bodech $x_0 = -1$, $x_0 = 0, x_0 = 2$. Vždy nakreslete graf funkce spolu s jejím Fourierovým rozvojem v daném bodě.

Příklad 4.3. Vypočtěte Taylorovou řadu funkce $f(x) = \cos x$ v bodech $x_0 = 0$ a $x_0 = \frac{\pi}{2}$.

Příklad 4.4. Vypočtěte Fourierovy rozvoje zadaných funkcí na zadaných intervalech. Vždy nakreslete graf zadané funkce spolu s jejím Fourierovým rozvojem postupně do prvních, druhých, … (jakých chcete) harmonických složek. Pak nakreslete tytéž grafy na širším intervalu, než byl zadaný.

a) $f(x) = 1-x^2,\ x \in \langle -\pi, \pi\rangle$; b) $y = 2-|x|,\ x \in \langle 0, 2\rangle$; c) $y = \pi - x,\ x \in \langle -\pi, \pi\rangle$.

5 Lineární algebra

Všeobecně se tvrdí, že lineární algebra je jednou z nejlépe propracovaných částí Maplu. Většina příkazů pro práci s maticemi, řešení soustav lineárních rovnic a pod. je z knihovny **linalg**. Proto vždy, když chceme s maticemi pracovat, nejprve tuto knihovnu zavoláme:

```
> with(linalg):
```

5.1 Operace s vektory

Zadat vektor můžeme tak, že do příkazové závorky za příkaz **vector** napíšeme seznam složek vektoru:

vector([výpis složek]);

jednotlivé prvky v hranaté závorce jsou odděleny čárkami.

```
> v1:=vector([-2,1,3]);
```

$$v1 := [-2, 1, 3]$$

Jednotlivé složky vektoru můžeme vypsat tak, že zadáme příslušný index v hranaté závorce.Vidíme, že nejsou-li složky konkrétně definovány, vytvoří se pouze oindexované pole.

```
> v1[1];v1[3];v2[2];
```

$$-2$$
$$3$$
$$v2_2$$

```
> u:=vector([1,-2,0]);
> v:=vector([-2,1,3]);
```

$$u := [1, -2, 0]$$
$$v := [-2, 1, 3]$$

Součet, rozdíl a násobení konstantou Maple zapíše symbolicky; pro vyhodnocení výsledku použijeme příkaz (vyhodnoť matici)

evalm(vektor nebo matice);

Použití tohoto příkazu ukážeme na příkladech:

```
> soucet:=u+v;
```

$$soucet := u + v$$

```
> evalm(soucet);
```

$$[-1,\ -1,\ 3]$$

```
> rozdil:=u-v;
```

$$rozdil := u - v$$

```
> evalm(rozdil);
```

$$[3,\ -3,\ -3]$$

```
> dvakrat_u:=2*u;
```

$$dvakrat\_u := 2\,u$$

```
> evalm(dvakrat_u);
```

$$[2,\ -4,\ 0]$$

Chceme-li zadané vektory napsat znovu, musíme také použít příkaz **evalm**.

```
> u;v;
```

$$u$$
$$v$$

```
> evalm(u);
```

$$[1,\ -2,\ 0]$$

Skalární součin dvou vektorů stejných rozměrů vypočteme příkazem

dotprod(první vektor, druhý vektor);

```
> dotprod(u,v);
```

$$-4$$

Příklad 5.1. Zadejte v Maplu tři trojrozměrné vektory $\overrightarrow{u}$, $\overrightarrow{v}$ a $\overrightarrow{w}$ a vypočítejte následující vektory:

$$\overrightarrow{x} = \overrightarrow{u} + \overrightarrow{v}, \qquad \overrightarrow{y} = 2\overrightarrow{u} - 3\overrightarrow{v}, \qquad \overrightarrow{z} = \overrightarrow{u} \cdot \overrightarrow{v}.$$

5.2 Matice a operace s nimi

Zadání matice lze provést následovně (jsou i jiné možnosti):

matrix(počet řádků, počet sloupců, [výpis prvků po řádcích]);

jednotlivé prvky v závorce jsou odděleny čárkami.

```
> A:=matrix(2,3,[1,-1,0,1,2,3]);
```

$$A := \begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

Chceme-li matici znovu vypsat, nestačí napsat jméno matice, ale provede se to příkazem **evalm**:

```
> A;
```

$$A$$

```
> evalm(A);
```

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

Matici transponovanou získáme pomocí příkazu
transpose(jméno matice);

```
> transpose(A);
```

$$\begin{bmatrix} 1 & 1 \\ -1 & 2 \\ 0 & 3 \end{bmatrix}$$

Součet a rozdíl matic a násobení matice konstantou se zadá očekávaným způsobem, ale výpočet sám se provede až příkazem **evalm**:

```
> B:=matrix(2,3,[1,2,3,4,5,6]);
> C:=A+B;
> F:=A-B;
> G:=2*A;
```

$$B := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$C := A + B$$
$$F := A - B$$
$$G := 2\,A$$

```
> evalm(C);
> evalm(F);
> evalm(G);
```

$$\begin{bmatrix} 2 & 1 & 3 \\ 5 & 7 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -3 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & -2 & 0 \\ 2 & 4 & 6 \end{bmatrix}$$

Násobení matic je operaci v mnoha ohledech (například co se týče komutitativity) rozdílné od násobení čísel, proto se součin matic zadává pomocí **&*** , nestačí jen *.

```
> M1:=matrix(2,2,[-1,2,-3,4]);
> M2:=matrix(2,2,[1,3,5,7]);
> M3:=M1&*M2;
> evalm(M3);
```

$$M1 := \begin{bmatrix} -1 & 2 \\ -3 & 4 \end{bmatrix}$$

$$M2 := \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$$

$$M3 := M1 \,\&*\, M2$$

$$\begin{bmatrix} 9 & 11 \\ 17 & 19 \end{bmatrix}$$

Pro výpočet determinantu slouží příkaz:
det(jméno matice);

```
> A:=matrix(3,3,[1,0,2,-1,1,3,4,-2,2]);
```

$$A := \begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & 3 \\ 4 & -2 & 2 \end{bmatrix}$$

```
> det(A);
```

$$4$$

Pro výpočet hodnosti matice slouží příkaz:
rank(jméno matice);

```
> A:=matrix(3,4,[1,1,3,-4,2,0,5,-1,1,-1,2,3]);
```

$$A := \begin{bmatrix} 1 & 1 & 3 & -4 \\ 2 & 0 & 5 & -1 \\ 1 & -1 & 2 & 3 \end{bmatrix}$$

```
> rank(A);
```

$$2$$

„Ručně“ ji zjistíme úpravou na trojúhelníkový tvar pomocí příkazu
gausselim(jméno matice);

```
> gausselim(A);
```

$$\begin{bmatrix} 1 & 1 & 3 & -4 \\ 0 & -2 & -1 & 7 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Opět vidíme, že hodnost je 2.

Výpočet inverzní matice se provede příkazem
inverse(jméno matice);

```
> A:=matrix(3,3,[2,5,7,6,3,4,5,-2,-3]);
```

$$A := \begin{bmatrix} 2 & 5 & 7 \\ 6 & 3 & 4 \\ 5 & -2 & -3 \end{bmatrix}$$

```
> IA:=inverse(A);
```

$$IA := \begin{bmatrix} 1 & -1 & 1 \\ -38 & 41 & -34 \\ 27 & -29 & 24 \end{bmatrix}$$

Můžeme se přesvědčit, že skutečně platí A*IA=E a IA*A=E:

```
> evalm(A&*IA);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> evalm(IA&*A);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Výpočet inverzní matice provedený „ručním“ způsobem: Vedle zadané matice připíšeme jednotkovou matici. Takovouto rozšířenou matici můžeme zadat přímo. Máme-li už matici, jejíž inverzi chceme počítat, zadanou, je jednodušší k ní jednotkovou matici připojit.

Jednotková matice řádu n se dá v Maplu zadat takto:
E := array(1 . . n, 1 . . n, identity);

Spojení dvou matic do jedné se provede příkazem
augment(matice1, matice2);

```
> E:=array(1..3,1..3,identity);
```

$$E := \mathrm{array}(\mathit{identity}, 1..3, 1..3, [])$$

```
> AE:=augment(A,E);
```

$$AE := \begin{bmatrix} 2 & 5 & 7 & 1 & 0 & 0 \\ 6 & 3 & 4 & 0 & 1 & 0 \\ 5 & -2 & -3 & 0 & 0 & 1 \end{bmatrix}$$

Nyní se budeme snažit pomocí ekvivalentních úprav dostat nalevo jednotkovou matici. V Maplu se to dá udělat jediným příkazem

gaussjord(matice);

```
> gaussjord(AE);
```

$$\begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & -38 & 41 & -34 \\ 0 & 0 & 1 & 27 & -29 & 24 \end{bmatrix}$$

Vpravo je inverzní matice k matici A.

Příklad 5.2. Vypočítejte matici $\mathbf{X} = 3 \cdot \mathbf{A} - 4 \cdot \mathbf{E}$, kde

$$\text{a)}\ \mathbf{A} = \begin{pmatrix} 2 & -1 & 0 \\ 1 & 6 & 3 \\ 5 & 1 & -2 \end{pmatrix}, \quad \text{b)}\ \mathbf{A} = \begin{pmatrix} 5 & 3 & -1 & 0 \\ 4 & -1 & 5 & 1 \\ 1 & 1 & 2 & 2 \\ 0 & -2 & 2 & 1 \end{pmatrix}, \quad \text{c)}\ \mathbf{A} = \begin{pmatrix} 5 & 3 \\ -4 & -1 \end{pmatrix}.$$

Příklad 5.3. Vypočítejte matice $\mathbf{X} = 2 \cdot \mathbf{A} - \mathbf{A}^T$ a $\mathbf{Y} = \mathbf{A} \cdot \mathbf{A}^T$, kde

$$\text{a)}\ \mathbf{A} = \begin{pmatrix} 5 & -2 & 6 \\ -3 & 5 & 3 \\ 1 & -2 & -5 \end{pmatrix}, \quad \text{b)}\ \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 2 & -2 \end{pmatrix}.$$

Příklad 5.4. Vypočítejte dvěma způsoby inverzní matici k $\mathbf{A}$, kde

$$\text{a)}\ \mathbf{A} = \begin{pmatrix} -1 & -3 & 2 \\ 0 & 0 & -3 \\ -1 & -2 & 0 \end{pmatrix}, \quad \text{b)}\ \mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}.$$

Příklad 5.5. Vypočítejte matici $\mathbf{X} = \mathbf{A} \cdot \mathbf{B}^{-1},$ kde

$$\mathbf{A} = \begin{pmatrix} -2 & 1 \\ -3 & 5 \\ 1 & -2 \end{pmatrix} \quad \text{a} \quad \mathbf{B} = \begin{pmatrix} -1 & 1 & -2 \\ 1 & -1 & -2 \\ 2 & -2 & 0 \end{pmatrix}.$$

5.3 Soustavy lineárních rovnic

Soustava lineárních rovnic v balíku **linalg** se vyřeší jediným příkazem

linsolve(matice soustavy, vektor pravých stran);

Výsledkem je vektor řešení soustavy. Vektor se zadává podobně jako matice.

```
> A:=matrix(2,2,[1,-1,1,1]);
> b:=vector(2,[1,3]);
> x:=linsolve(A,b);
```

$$A := \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$b := [1, 3]$$

$$x := [2, 1]$$

Tímto příkazem ovšem získáme řešení soustavy, aniž bychom věděli, jak se k němu vlastně došlo. Pro pochopení metody řešení je lepší dojít k výsledku krok za krokem - stejně, jako by se výpočet prováděl „ručně“ pomocí Gaussovy eliminační metody. Znamená to upravit rozšířenou matici soustavy pomoci ekvivalentních úprav na trojúhelníkový tvar.

V Maplu existují příkazy pro ekvivalentní úpravy matic

Výměna dvou řádků matice s indexy r1 a r2 se provede příkazem
swaprow(jméno matice, r1, r2);

Vynásobení r-tého **řádku konstantou** k se provede příkazem
mulrow(jméno matice, r, k);

Přičtení k-násobku řádku r1 k řádku r2 se provede příkazem
addrow(jméno matice, r1, r2, k);

Příklad 5.6. Postup předvedeme na řešeni soustavy rovnic:

$$\begin{array}{rrrcr} x & +4y & -2z & = & 2 \\ 2x & -y & +3z & = & -1 \\ -x & -6y & & = & 4 \end{array}$$

Řešení: Nejprve zadáme rozšířenou matici soustavy:

```
> A:=matrix(3,4,[1,4,-2,2,2,-1,3,-1,-1,-6,0,4]);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 2 & -1 & 3 & -1 \\ -1 & -6 & 0 & 4 \end{bmatrix}$$

Nyní budeme upravovat na trojúhelníkový tvar.
Ke druhému řádku přičteme první řádek vynásobený -2:

```
> A:=addrow(A,1,2,-2);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & -9 & 7 & -5 \\ -1 & -6 & 0 & 4 \end{bmatrix}$$

Nyní ke třetímu řádku přičteme první:

```
> A:=addrow(A,1,3,1);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & -9 & 7 & -5 \\ 0 & -2 & -2 & 6 \end{bmatrix}$$

Teď se hodí např. vydělit třetí řádek minus dvěma

```
> A:=mulrow(A,3,-1/2);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & -9 & 7 & -5 \\ 0 & 1 & 1 & -3 \end{bmatrix}$$

vyměnit druhý a třetí řádek

```
> A:=swaprow(A,2,3);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & 1 & 1 & -3 \\ 0 & -9 & 7 & -5 \end{bmatrix}$$

a ke třetímu řádku přičíst devítinásobek druhého

```
> A:=addrow(A,2,3,9);
```

$$A := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & 1 & 1 & -3 \\ 0 & 0 & 16 & -32 \end{bmatrix}$$

Tím máme soustavu v trojúhelníkovém tvaru a již ji lze snadno i „ručně“ vyřešit. Máme $16\,z = -32$, potom $z = -2$. Dosazením do předchozích rovnic dostaneme $y = -1$ a $x = 2$.

Úpravu matice na trojúhelníkový tvar lze provést i naráz, a to příkazem

gausselim (matice);

```
> B:=matrix(3,4,[1,4,-2,2,2,-1,3,-1,-1,-6,0,4]);
```

$$B := \begin{bmatrix} 1 & 4 & -2 & 2 \\ 2 & -1 & 3 & -1 \\ -1 & -6 & 0 & 4 \end{bmatrix}$$

```
> gausselim(B);
```

$$\begin{bmatrix} 1 & 4 & -2 & 2 \\ 0 & -9 & 7 & -5 \\ 0 & 0 & \frac{-32}{9} & \frac{64}{9} \end{bmatrix}$$

Je vidět, že jsme došli k jinému výsledku, ale řešení bude stejné.

Řešení soustavy v trojúhelníkovém tvaru lze také najít jediným příkazem, a to

backsub (rozšířená matice soustavy v trojúhelníkovém tvaru);

```
> backsub(A);
```

$$[2,\ -1,\ -2]$$

V Maplu lze ze zadané soustavy rovnic sestavit matici, resp. rozšířenou matici soustavy a naopak. Pro přechod od soustavy k matici slouží příkaz

genmatrix(soustava, neznámé, v apostrofech jméno pravé strany);

Výsledkem je matice soustavy a vektor pravých stran.
Anebo

genmatrix(soustava, neznámé, 'flag');

Výsledkem je rozšířená matice soustavy.

Soustavu rovnic lze zadat jako seznam v hranatých závorkách. Stejně tak jména neznámých.

```
> rovnice:=[x+y=4,x-y=3];
```

$$rovnice := [x + y = 4,\ x - y = 3]$$

```
> A:=genmatrix(rovnice,[x,y],'b');
```

$$A := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

```
> evalm(b);
```

$$[4,\ 3]$$

```
> Ab:=genmatrix(rovnice,[x,y],'flag');
```

$$Ab := \begin{bmatrix} 1 & 1 & 4 \\ 1 & -1 & 3 \end{bmatrix}$$

Pro přechod od matici k soustavě slouží příkaz

geneqns(matice, neznámé, pravá strana);

Použití tohoto příkazu nejlépe uvidíme na příkladě.

```
> A:=matrix(2,2,[1,1,2,-1]);
```

$$A := \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}$$

```
> b:=vector(2,[0,3]);
```

$$b := [0,\ 3]$$

```
> geneqns(A,[x,y],b);
```

$$\{x + y = 0,\ 2\,x - y = 3\}$$

```
> geneqns(A,t,b);
```

$$\{t_1 + t_2 = 0,\ 2\,t_1 - t_2 = 3\}$$

Viděli jsme, že Maple umožňuje několik způsobů řešení lineárních soustav rovnic. Knihovna **Linalg** má opravdu hodně možnosti, které můžeme využit. Můžeme si například ověřit větu Frobeniovu tím, že vypočítáme hodnosti matice soustavy a matice rozšířené. Regulární soustavy můžeme řešit Cramerovym pravidlem nebo pomocí inverzní matice. Je možné také prověst zkoušku správnosti řešení pomocí příkazu **subs**.

Příklad 5.7. Řešte následující soustavu rovnic v Maplu

$$\begin{array}{rcrcrcr} x & + & y & - & z & = & -1 \\ 2x & - & y & + & z & = & 4 \\ 3x & - & 7y & - & 2z & = & -1 \\ -2x & + & 5y & + & z & = & 1 \end{array}$$

Příklad 5.8. Řešte následující soustavy rovnic

$$\begin{aligned} 3x_1 + 2x_2 + 5x_3 + 2x_4 + 7x_5 &= 1 \\ 6x_1 + 4x_2 + 7x_3 + 4x_4 + 5x_5 &= 8 \\ 3x_1 + 2x_2 - x_3 + 2x_4 - 11x_5 &= 13 \\ 6x_1 + 4x_2 + x_3 + 4x_4 - 13x_5 &= 20 \end{aligned}$$

$$\begin{aligned} 3x_1 + 2x_2 + 5x_3 + 2x_4 + 7x_5 &= 0 \\ 6x_1 + 4x_2 + 7x_3 + 4x_4 + 5x_5 &= 0 \\ 3x_1 + 2x_2 - x_3 + 2x_4 - 11x_5 &= 0 \\ 6x_1 + 4x_2 + x_3 + 4x_4 - 13x_5 &= 0 \end{aligned}$$

Příklad 5.9. Užitím Cramerova pravidla najděte x_1 a x_2 vyhovující soustavě rovnic

$$\begin{array}{rrrrl} 2x_1 + & x_2 & & & = 1 \\ x_1 + & 2x_2 + & x_3 & & = 0 \\ & x_2 + & 2x_3 + & x_4 & = 0 \\ & & x_3 + & 2x_4 & = 0 \end{array}$$

6 Maplety

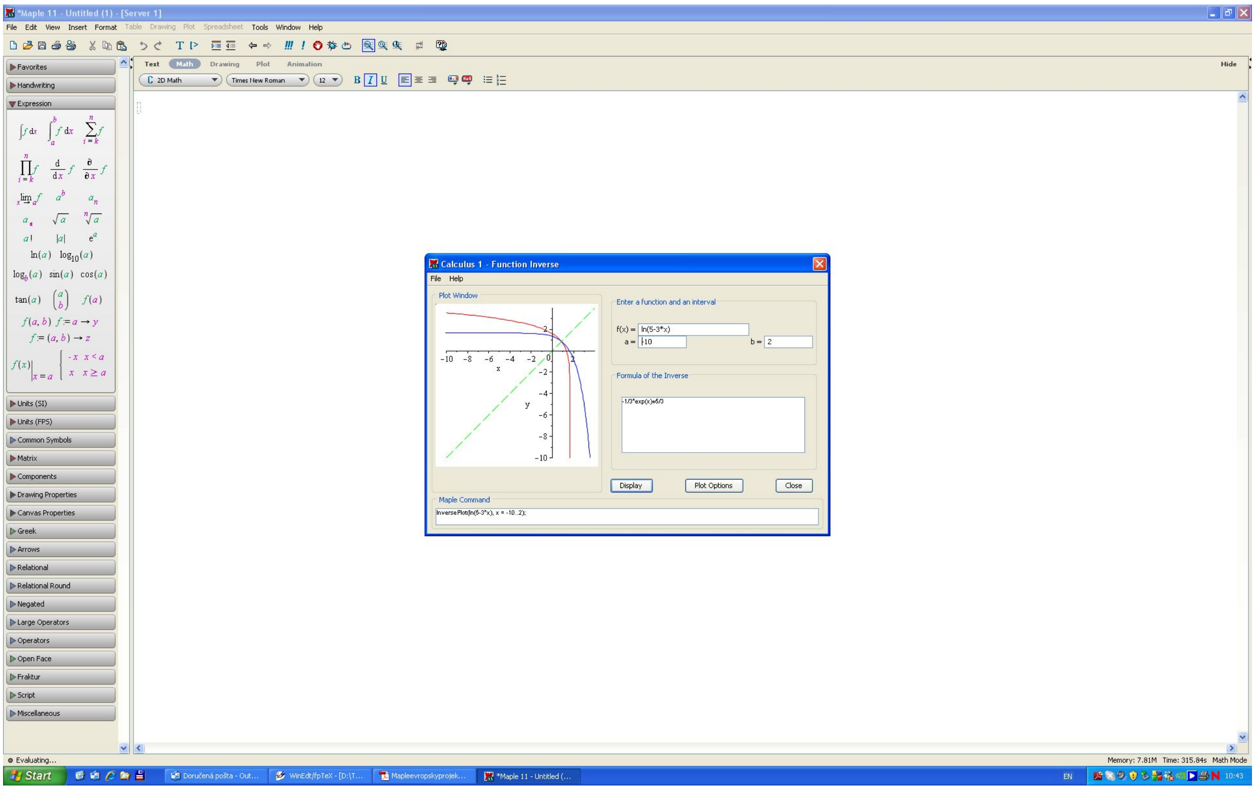
Maplets jsou nově definovaná grafická uživatelská rozhraní pro demonstrace a výpočty. Ovládají se většinou tlačítkovými lištami a obsahují okna pro zadání vstupních a zobrazení výstupních hodnot. Mají také samostatný Help.

Většinu Maplets můžete spustit přímo z **Menu Bar** volbou. Například

Tools – > Tutors – > Calculus - Single Variable – > Function inverse

pro výpočet a grafickou demonstraci inverzní funkce.

Ukázka rozpracovaného Mapletu se zobrazením výpočtu inverzní funkce s vykreslováním obrázků.



7 Závěr

Maple je počítačové prostředí, které bylo vyvinuto, pro zjednodušení a zrychlení výpočtů v matematice. Na rozdíl od klasických programů pro numerické výpočty (např. MATLAB - také obsahuje nástroj pro symbolické výpočty) modeluje matematické operace se symbolickými výrazy. Umožňuje provádět jak symbolické a numerické výpočty, tak vytvářet grafy funkcí, programovat vlastní funkce či procedury, ukládat data v několika formátech (např. LaTeX , HTML , RTF , MATHML, . . .) a dokonce provádět export do programovacích jazyků (např. C, Fortran 77, . . .). Funkce implementované v Maplu pokrývají širokou oblast matematiky od základů lineární algebry, diferenciálního a integrálního počtu, přes diferenciální rovnice, geometrii až k logice. Systém je primárně určen pro: symbolické operace v matematice, numerické výpočty, zobrazování grafů.

Obsah