

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ v PRAZE

Fakulta elektrotechnická

Katedra řídicí techniky

**Přehled volně dostupného programového vybavení pro
řešení parciálních diferenciálních rovnic**

BAKALÁŘSKÁ PRÁCE

Vedoucí bakalářské práce: Ing. Petr AUGUSTA

Oponent bakalářské práce: Ing. Radek FRÍZEL

Studijní program: Elektrotechnika a informatika

Obor: Kybernetika a měření

2007

Jelena TOMICOVÁ

ČVUT Praha

Katedra řídicí techniky

Školní rok: 2006/2007

Z a d á n í b a k a l á ř s k é p r á c e

Student: Jelena T o m i c o v á

Obor: Kybernetika a měření

Název tématu: Přehled volně dostupného programového vybavení pro řešení
parciálních diferenciálních rovnic

Z á s a d y p r o v y p r a c o v á n í :


1. Vypracujte přehled volně dostupných výpočetních balíků určených k řešení parciálních diferenciálních rovnic.
2. Z vypracovaného přehledu vyberte několik balíků, nainstalujte si je a naučte se základní práci s nimi. Vytvořte návod, jak pomocí těchto balíků vyřešit parciální diferenciální rovnici.
3. Vyřešte zadanou parciální diferenciální rovnici.

Seznam odborné literatury: Dodá vedoucí práce

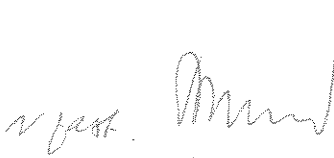
Vedoucí bakalářské práce: Ing. Petr Augusta

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007


Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 6. 3. 2007

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 24. 8. 2007

.....
Blanka Jomová

podpis

Poděkování

Ráda bych poděkovala vedoucímu práce, Ing. Petru Augustovi, za odborné připomínky, trpělivost a ochotu při zpracování tohoto textu, svým rodičům a přátelům za psychickou podporu v tomto pro mě nelehkém období. Mé díky také Báře a Michaele za celkové pročtení i připomínky k celkovému vzhledu práce samotné.

Anotace

Název práce: Přehled volně dostupného programového vybavení pro řešení parciálních diferenciálních rovnic

Autor: Tomicová Jelena

Katedra: Katedra řídicí techniky

E-mail: tomicj1@fel.cvut.cz

Vedoucí bakalářské práce: Ing. Augusta Petr

Abstrakt: Cílem práce je najít volně dostupné výpočetní balíky pro řešení parciálních diferenciálních rovnic metodou FEM (Metoda konečných prvků), tyto programy vyzkoušet, sestavit návod na jejich použití a tento návod následně aplikovat na zadaných rovnicích.

Klíčová slova: freeware, parciální diferenciální rovnice, FEM

Annotation

Title: An overview of free available software for partial differential equations solution

Author: Tomicová Jelena

E-mail: tomicj1@fel.cvut.cz

Department: Department of Control Engineering

Supervisor: Ing. Augusta Petr

Abstract: The aim of my work is to find freely available computing packages for analysis of partial differential equations by using a method of FEM (Finite Element Method), to test these programs out, to put together instructions for their application and then to apply the instructions on scheduled equations.

Keywords: freeware, partial differential equations, FEM

Obsah

Obsah.....	6
1. Úvod.....	8
2. Parciální diferenciální rovnice	9
2.1. Základní definice.....	9
2.2. Počáteční a okrajové podmínky	9
2.3. Řešení rovnic.....	10
2.4. Metoda konečných prvků – FEM.....	10
2.5. Zadané rovnice	11
3. Freeware využívající metodu FEM	12
3.1. Getfem++	12
3.1.1. Systémové požadavky	13
3.1.2. MinGW.....	13
3.1.3. Python.....	13
3.1.4. GiD.....	14
3.1.5. GmSH.....	15
3.1.6. emc2	15
3.2. Elmer 5.3.0	16
3.2.1. Systémové požadavky	16
3.2.2. Překladače	17
3.2.3. Compag Visual Fortran 6.6	17
3.3. FreeFem++ 2.15-1.....	18
3.3.1. Systémové požadavky	18
3.4. FlexPDE 5.0.15	18
3.4.1. Systémové požadavky	20
3.4.2. Typy licencí.....	20
4. Návod pro FlexPDE 5.0.15	22
4.1. Základní seznámení s programem.....	22
4.2. Řešení zadané rovnice (1.1)	25
4.3. Řešení zadané rovnice (1.2)	28
5. Závěr.....	32
6. Použité zdroje.....	34

6.1.	Literatura	34
6.2.	Elektronické zdroje	34
7.	Příloha A	36
8.	Příloha B.....	37
9.	Příloha C.....	38
10.	Příloha D	40
11.	Příloha E.....	41

1. Úvod

Cílem mé bakalářské práce je najít volně dostupné výpočetní balíky pro řešení parciálních diferenciálních rovnic metodou konečných prvků (tzv. FEM metoda), nalezené programy otestovat, sestavit návod na jejich použití a tento návod následně aplikovat na zadaných rovnicích.

Svou práci jsem se snažila pojmout jako ucelený soubor poznatků, který by měl poskytnout nejen obecný náhled na problematiku parciálních diferenciálních rovnic, tj. co vůbec termín parciální diferenciální rovnice znamená, kam se v matematice zařazuje, atd., ale současně by měl čtenáři detailně přiblížit použití metody konečných prvků při řešení zmiňovaných rovnic včetně jejího softwarového využití. Kromě toho čtenář dostává k dispozici seznam nalezených freewarových zdrojů, včetně jejich instalací a základního popisu, který by měl usnadnit výběr takového programu, jež by splňoval podmínky, které jsou na program kladeny. Vzhledem ke skutečnosti, že se jedná o freewary, dovoluji si již v úvodu upozornit, že nikdy nelze plně uspokojit veškeré uživatelské nároky. V druhé části práce uvádím stručný návod pro řešení dvou vzorových parciálních diferenciálních rovnic.

Práce samotná byla tvořena v opačném pořadí, než jsou řazeny kapitoly. Nejprve jsem se snažila s pomocí vyhledávacích automatů získat na internetu informace, zda a jaké freewarové programy by bylo možno použít. Po konzultaci s Ing. Augustou jsem zjistila, že minimální úspěšnost při vyhledávání byla zapříčiněna zadáním nevhodné kombinace dotazovacích hesel. Po jejich změně se rozsah nalezených programů sice zvýšil, avšak většinou se jednalo o sharewarové či již od počátku placené verze programů. Pro účely této práce však byl výsledek vyhledávání postačující.

V následující fázi jsem níže popsané programy odzkoušela a sepsala stručný, ale současně – jak se domnívám - výstižný popis jejich použití.

Až v samotném závěru jsem k praktické aplikaci metody konečných prvků při řešení parciálních diferenciálních rovnic na freewarových programech přidala odpovídající teoretický úvod, v jehož rámci jsem – kromě již zmiňovaných definic základních pojmů včetně systematického začlenění parciálních diferenciálních rovnic do matematiky - podrobněji rozebrala metodu konečných prvků, aby i laik měl možnost nahlédnout do teorie, která k tomuto tématu také patří. Tento teoretický rozbor je však popsán pouze v úvodu, protože cílem ani předmětem této práce nebyla detailní sumarizace veškerých vědeckých teorií a axiomů spojených s diferenciálními rovnicemi.

2. Parciální diferenciální rovnice

2.1. Základní definice

Diferenciální rovnice je rovnice, jejíž proměnné jsou derivace funkcí. Všeobecně je můžeme rozdělit do dvou skupin: na obyčejné diferenciální rovnice (ODR) a parciální diferenciální rovnice (PDR). Obyčejné diferenciální rovnice obsahují derivace hledané funkce jen podle jedné proměnné, oproti tomu parciální diferenciální rovnice obsahuje derivace hledané funkce podle více proměnných. Můžeme také říci, že daná rovnice obsahuje parciální derivace této funkce.

Obecný tvar takové rovnice řádu k můžeme vyjádřit takto:

$$F\left(x, u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1^2}, \dots, \frac{\partial^k u}{\partial x_n^k}\right) = 0 \quad (1.0)$$

kde u je neznámá funkce proměnné $x \in R^n$, F je funkce příslušného počtu proměnných. Řád nejvyšší derivace, kterou v rovnici nalezneme, nám určuje řád rovnice samotné. Z tohoto hlediska tak můžeme diferenciální rovnice dále rozdělit na rovnice prvního řádu a řádu vyššího. (citace z [1])

Z pohledu matematiky je důraz kladen na existenci řešení, jednoznačnost a závislost řešení na počátečních a okrajových podmínkách, oproti tomu fyzika hledá analytické řešení, tj. takové funkce $u(t)$, které jsou řešením dané rovnice. V případě, že tuto funkci nelze vyjádřit analyticky, přistupuje se k numerickému řešení rovnice.

2.2. Počáteční a okrajové podmínky

Pomocí počátečních podmínek určíme, jak má daná funkce nebo její derivace vypadat na celé oblasti v určitém okamžiku, pro který danou rovnici řešíme. Tyto úlohy jsou označovány jako Cauchyho úlohy.

Okrajové podmínky jsou ty podmínky, jež daná funkce nebo její derivace splňuje pouze v určitých bodech. Jelikož jsou tyto body většinou na krajích oblasti, pro níž rovnici řešíme, je odsud odvozen i název pro podmínky samotné. Typy těchto úloh nazýváme okrajové problémy (úlohy). Lze také zaměnit pořadí slov a úlohy označovat jako úlohy s okrajovými podmínkami.

Pro řešení parciálních diferenciálních rovnic jsou obě uvedené podmínky nezbytné. Společně s rovnicí totiž tvoří problém, jehož řešení hledáme. Pokud se nám podaří tyto podmínky splnit, docílíme vždy jednoznačného řešení. (informace čerpány ze [18],[19],[20])

2.3.Řešení rovnic

Při řešení diferenciálních rovnic je vždy nutné specifikovat, o kterém řešení hovoříme.

Řešení samo o sobě není jen jedno, ale dělí se na:

- a) *Obecné* – řešení obsahuje libovolnou integrační konstantu; u většiny případů parciálních diferenciálních rovnic toto řešení nenastává (výjimkou je d'Alambertovo řešení vlnové rovnice),
- b) *Partikulární (částečné)* – k takovému řešení dospějeme, pokud každé konstantě v obecném řešení přiřadíme číselnou hodnotu,
- c) *Singulární (výjimečné)* – jde o takové řešení, které platí pouze v některých bodech oboru; toto řešení mají jen některé rovnice. (*informace čerpány z [18]*)

2.4.Metoda konečných prvků – FEM

Metoda konečných prvků je založena na variačním počtu - hledá minimum nějakého zobrazení z množiny funkcí do množiny čísel. Můžeme hovořit o pravidle, dle kterého přiřadíme funkci na jejím definičním oboru (nebo části) nějakou číselnou hodnotu.

Praktický výpočet v softwaru se skládá ze tří kroků.

- a) *Preprocessing* – příprava vstupních dat. Vytvoří se geometrický model tělesa nebo soustavy, který se celý rozdělí na malé prvky konečných rozměrů. Rohy těchto prvků se nazývají uzlové body. Hustota sítě je rozhodující pro přesnost výsledku.
- b) *Solver* – spouštění řešiče. Na základě zadaných vstupních hodnot se sestaví a vyřeší soustava rovnic. Je nutné zadat všechny vstupní údaje, jinak by nešla metoda konečných prvků použít. Nepřímé úlohy tedy touto cestou řešit nelze.
- c) *Postprocessing* – zpracování výsledků. Dojde k znázornění kterýchkoli neznámých parametrů v tělese, či oblasti v různých podobách, které program nabídne.

Pokud se v zadané úloze objeví nelineární chování, systém rozloží tuto nelinearitu na řadu malých kroků tak, aby je bylo možné s dostatečnou přesností v daném úseku linearizovat. (*informace čerpány z [4],[8],[9],[12],[17]*)

2.5. Zadané rovnice

Obě rovnice spadají do tzv. rovnic matematické fyziky. Byly odvozeny z fyzikálních principů a v určitém rozsahu a s určitou přesností tyto fyzikální děje také popisují.

a) Rovnice (1.1) popisuje šíření tepla,

$$\frac{\partial u(x, y, t)}{\partial t} = \kappa \left(\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} \right) + q(t, x, y) \quad (1.1)$$

kde u je teplota $[K]$,

t je čas $[s]$,

x, y jsou prostorové souřadnice $[m]$,

κ je konstanta $[m^2 s^{-1}]$,

q je vstupní zahřívání $[Ks^{-1}]$.

b) Rovnice (1.2) popisuje výchylku desky,

$$\frac{\partial^4 w(x, y, t)}{\partial x^4} + 2 \frac{\partial^4 w(x, y, t)}{\partial x^2 \partial y^2} + \frac{\partial^4 w(x, y, t)}{\partial y^4} + \frac{\rho}{D} \frac{\partial^2 w(x, y, t)}{\partial t^2} = \frac{q(x, y, t)}{D} \quad (1.2)$$

kde w je výchylka desky ve směru z $[m]$,

ρ je hustota desky $[kgm^{-2}]$,

q je vnější síla působící na desku $[Nm^{-2}]$,

$\frac{\partial^2 w}{\partial t^2}$ je zrychlení výchylky ve směru z $[ms^{-2}]$,

$$D = \frac{Eh^3}{12(1-\nu^2)}, \quad (1.3)$$

kde ν je Poissonovo číslo $[-]$,

h je tloušťka desky $[m]$,

E je Youngův modul $[Nm^{-2}]$.

3. Freeware využívající metodu FEM

3.1. Getfem++

Projekt Getfem++ se soustředí na vývoj generické a výkonné C++ knihovny pro početní využívání metody konečných prvků. Cílem projektu je poskytnout knihovnu dovolující provádět výpočty v jakékoliv základní matici (včetně společných konečných metod) na velkém množství druhů metod, prvků a libovolných dimenzích (např.: ne jen 2D či 3D dimenze).

Getfem++ nabízí kompletní oddělení mezi integračními metodami (přesné nebo přibližné), geometrickými transformacemi (lineární nebo nelineární) a konečnými metodami prvků na libovolné stupnici. Díky tomuto převodu na základní problémy lze účinně docílit výkonnějšího řešení technických úloh.

Vývoj dalších metod je postupný. Musí zahrnovat odkaz na svůj základní prvek (v mnoha případech je zde jen popis základních funkcí bez jakýchkoli dalších informací). Pracuje se na rozšíření Hermitových prvků, částečných polynomů, nepolynomialických a vektorových prvků.

Knihovna v sobě obsahuje obvyklé nástroje pro konečné prvky, jakými jsou sestavovací proces pro klasické parciální diferenciální rovnice, interpolační metody, výpočet hranic, vykreslovací operace (zahrnující automatické vylepšení), hraniční podmínky, nástroje po zpracování jako je rozdělování na díly z vykreslené sítě a další.

Getfem++ může být použit k vytvoření kódů pro mnoho základních úloh, kde se vyskytují konečné prvky, integrační metody i síťové zobrazení. K dispozici je tak mnoho parametrů, které se mohou snadno měnit, což dovoluje široké spektrum experimentálních pokusů.

Getfem++ sám osobě nemá vykreslovací schopnosti (kromě pravidelných mřížek), proto je nezbytné si tyto mřížky naimportovat.

Podporované formáty:

- GiD
- GmSH
- emc2

(informace čerpány z [13])

3.1.1. Systémové požadavky

Operační systém	Verze
Linux/x86	g++ 3.x, g++ 4.x
Linux/Uranium	g++
Intel	C++ Compiler 8.0
Windows	s MinGW a MSys
MacOS X Tiger	s rozhraním Python nebo Matlab

Tab.3.1: Getfem++ - systémové požadavky

(informace čerpány z [13])

3.1.2. MinGW

Aby bylo možno docílit stejných výsledků a chování programu pod systémem Windows, doporučují tvůrci nainstalování MinGW = Minimum GNU pro operační systém Windows 32 bit.

- *MinGW* – je soubor volně dostupných a volně šiřitelných specifikací a knihoven pro Windows. Díky MinGW dojde k importu knihoven kombinovaných s nástroji GNU, jež umožní chod programům, které by za jiných podmínek vyžadovaly existenci třetí strany s C běžící pod DLLs.
- *MSYS* – minimální systém, který poskytuje konfigurační skript POSIX/Bourne za účelem vytvoření Makefile s užitím make. (informace čerpány z [15])

3.1.3. Python

Jak bylo již na počátku kapitoly zmíněno, Getfem sám o sobě je vyvíjen jako knihovna, kterou následně vyvolává vyšší program. Ten pak díky tomuto řešení zpracovává požadované úkoly. Jako vyšší program lze využít např. Matlab, který však není freewarovým programem, a nebo právě Python, který je možné použít jako freeware, pokud se nejedná o komerční účely. Dle tvůrců Getfem++ se liší pouze grafickou prezentací, jinak se chová velmi podobně jako prostředí Matlab. Python je dynamicky objektově orientovaný jazyk, který je používán pro vývoj mnoha druhů softwaru. Nabízí silnou podporu pro integraci s dalšími jazyky a nástroji, které používají rozsáhlé standardní knihovny, a lze se jej naučit v několika málo dnech. Mnoho uživatelů popisuje značný zisk z práce v tomto prostředí a přikládají pythonu zásluhu za rychlejší vývoj jejich projektů.

Operační systém	Verze
Windows	95, 98, NT, 2000, ME, XP
Mac OS X	10.3 (Panther), 10.3.9, 10.4
DOS	DOS/DPMI
Linux	Python je již zahrnut, lze jej aktualizovat přímo ze zdroje

Tab.3.2: Python – systémové požadavky

(informace čerpány z [17])

3.1.4. GiD

GiD byl vytvořen jako univerzální, adaptivní a uživatelsky přátelské grafické rozhraní pro geometrické modelování, zadávání vstupních dat a následnou vizualizaci výsledků mnoha rozličných typů číselných simulací v různých programech. Typické úlohy, které jsou GiD snadno zobrazovány, pocházejí z konstrukční mechaniky, dynamiky kapalin, elektromagnetismu, přeměny tepla, geomechaniky atd. Tyto úlohy využívají metody konečných prvků, konečnou délku, hraniční prvky...

GiD může ideálně sloužit jako pomůcka na univerzitách, výzkumných centrech, vývojářských stanicích nebo jako doplněk pro další programy pracující s číselnou simulací.

Výhody GiD můžeme popsat následujícím způsobem:

- a) *Univerzální* – GiD je ideální pro generování všech informací (strukturní a nestrukturní sítě, hranice a vstupní podmínky, druh materiálu, vizualizace výsledků) potřebných pro analýzu vědecké úlohy a využívaných inženýry při jejich číselných simulačních metodách.
- b) *Adaptivní* – GiD se přizpůsobuje jakémukoliv číselnému kódu. Uživatel může použít bezpočet číselných formátů pro zápis a čtení. Vstup a výstup může být uzpůsoben a je kompatibilní s domácím softwarovým vybavením uživatele. Rozličná menu, datové vstupy a grafická vizualizace mohou být nastaveny dle zvyklostí uživatele.
- c) *Uživatelsky přátelské* – Vývoj byl zaměřen na potřeby uživatele a na jednoduchost, rychlost, efektivitu a přesnost. To vše závisí jen na zadaných vstupních datech a na zvoleném výsledném stupni samotné vizualizace.

Zkušební verze GiD je plně funkční a může být použita jako freeware. Pouze zobrazovaná geometrie je limitována (25 ploch, síť na 1000 uzlů).

Operační systém	Verze
Windows	95, 98, NT, 2000, ME, XP
Silicon Graphics	IRIX 6.2
Linux	Kernel 2.2 nebo vyšší
Linux 64 bits	(AMD64) Kernel 2.6

Tab.3.3: GiD – systémové požadavky

(informace čerpány z [14])

3.1.5. GmSH

GmSh je 3D mřížkový generátor s vestavěným CAD a post-procesorem. Byl sestaven za účelem poskytování jednoduchých nástrojů pro tvorbu mřížek k teoretickým úlohám, kde je vstup zadáván pomocí parametrů. Uživatelé umožňuje pokročilejší zobrazování.

GmSH byl vytvořen na základě čtyř modulů: jde o popis geometrie, síť, řešič a post-procesing. Změnu jakéhokoliv vstupu kteréhokoliv z modulů lze provést interaktivně s využitím uživatelského rozhraní nebo pomocí ASCII textového souboru, který využije vlastní GmSH skriptovací jazyk.

Operační systém	Verze
Windows	NT, 2000, XP
Linux	Intel, glibc 2.3
Mac OS X	Universal, 10.4

Tab.3.4: GmSH – systémové požadavky

(informace čerpány z [5])

3.1.6. emc2

Program emc2 je přenosný, interaktivní a grafický software zabývající se 2D geometrií a tvorbou mřížek. S jeho pomocí je možno vytvořit nebo modifikovat geometrii (CAD), definovat jednotlivé čáry, definovat podtvary a definovat některé referenční hodnoty, které slouží pro popis hranic a druhu materiálu. Elementy sítě jsou trojúhelníky a čtyřúhelníky.

emc2 využívá dva druhy sítí:

- mřížkovou síť
- Delaunay Voronoï (automatickou síť)

Operační systém	Verze
sun4 -Bstatic SunOS	4.1.2
Linux	RedHad 5.0

Tab.3.5: emc2 – systémové požadavky

(informace čerpány z [7])

3.2.Elmer 5.3.0

Program Elmer je freeware vytvořený společností CSC pro modelování fyzikálních úloh. Vývoj začal v roce 1995 ve spolupráci s finskými univerzitami, výzkumnými ústavy a průmyslem.

Elmer obsahuje například modely dynamiky kapalin, konstrukční mechaniky, elektromagnetismu, tepelné přeměny a zvukové přeměny. Tyto úlohy jsou popsány parciálními diferenciálními rovnicemi, které jsou v programu řešeny pomocí metody konečných prvků.

Elmer je převážně vyvíjen pro operační systémy Unix/Linux. (informace čerpány z [11])

3.2.1. Systémové požadavky

OS	Architektura
Linux	x86, x86-64
FreeBSD	x86
Sun Solaris	sparcv7, sparcv9
IBM AIX	Power4
Apple OS X 10.3	PPC
Windows 2000,XP	x86

Tab.3.6: Elmer 5.3.0 – systémové požadavky

(informace čerpány z [11])

3.2.2. Překladače

Překladač	Poznámky
GNU(gcc,g++,g77,gfortran,g95)	
Intel7.1,8.1,9.0(icc,ifort ifc)	
Pathscale(pathcc,pathCC,pathf90)	
Portlangroup6.0-4(pgcc,pgCC,pgf90)	
SunStudio	
IBMFortran	
HPfortranandVisualC++	
Absoft	má potíže s překladem
CompaqVisualFortran	doporučeno pro OS Windows
Gfortran	doporučeno pro OS Windows

Tab.3.7: Elmer 5.3.0 – seznam překladačů

(informace čerpány z [11])

Dobrym pravidlem je překládat vše programem od jednoho výrobce. Výjimkou je „ifort“ a „pathscale“, kteří pracují korektně s překladači „gnu c“ i „c++“.

3.2.3. Compaq Visual Fortran 6.6

Vyzkoušený překladač, se kterým program Elmer správně pracuje pod operačním systémem Win x86 System.

Jedná se o dokončený vývojový systém, který zahrnuje kompilátor a provozní systém, knihovny a nástroje kombinované s vývojovým studiem Microsoft, jež jsou zahrnuty v jejich systémech, a další příslušné nástroje.

Plně podporované jazykové standardy:

- FORTRAN 66 (ANSI X3.9-1966)
- FORTRAN 77 (ANSI X3.9-1978)
- FORTRAN 77 (ISO 1539-1980(E))
- Fortran 90 (ANSI X3.198-1992)
- Fortran 95 (ISO/ IEC 1539-1:1997(E))

Systémové požadavky:

- PC s 90MHz (nebo rychlejší) Intel Pentium (nebo kompatibilní) procesor
- nainstalovaný operační systém Microsoft Windows s 24MB paměti (doporučeno 32MB)

Windows

95, 98, ME, 2000, NT 4.0 (Service Pack 3), XP

- místo na pevném disku – 33 MB pro spuštění z CD-ROM a více jak 400MB pro nainstalování komponent pro standardní nebo profesionální verzi, 10 - 90 MB pro Compaq Array Visualizer (velikost závisí na vybraném nastavení), 48 MB pro Internet Explorer verze 4
- 32-bit CD-ROM mechaniku
- VGA monitor (17" SVGA doporučen)
- myš nebo kompatibilní ukazovací zařízení

Hewlett-Packard již zastavil vývoj a podporu tohoto programu. (*informace čerpány z [10]*)

3.3.FreeFem++ 2.15-1

FreeFem++ je realizací jazyka specializujícího se na metodu konečných prvků. Tak umožňuje jednoduše řešit parciální diferenciální rovnice (PDE).

Slouží k řešení problémů zahrnujících parciální diferenciální rovnice z mnoha odvětví fyziky, jakými jsou například pohled na strukturu kapalin z různých úhlů při interpolaci dat a práce s těmito daty v jednom programu. FreeFem++ obsahuje stromovou strukturu interpolačních algoritmů a využívá jazyk pro manipulaci s daty na několika vygenerovaných sítích.

FreeFem++ je psán v jazyce C++. Funguje na jakémkoliv Unixovém operačním systému jako (s g++ verzí 3 a lepším, X11R6 nebo OpenGL s GLUT) Linux, FreeBSD, Solaris 10, Microsoft Windows (95, 98, 2000, NT, XP) a MacOS X (nativní verze využívající OpenGL). FreeFem++ nahrazuje starší freefem a freefem+. (*informace čerpány z [6]*)

3.3.1. Systémové požadavky

Operační systém	Verze
Unix	g++ verze 3 a vyšší, x11R6, OpenGL s GLUT
Windows	95, 98, 2000, NT, XP
MacOS X	verze využívající OpenGL

Tab.3.8: FreeFem++ 2.15-1 – systémové požadavky

(*informace čerpány z [6]*)

3.4.FlexPDE 5.0.15

FlexPDE je kreslič modelu pro metodu konečných prvků a zároveň číselný řešič. To znamená, že z uživatelem zadané rovnice FlexPDE spustí nutné operace pro převod

samotného popisu parciální diferenciální rovnice do modelu, vyřeší takto popsaný systém a výsledek předá v grafické a číselné podobě.

FlexPDE je také nástroj pro řešení různých úloh. Zahrnuje velkou škálu funkcí nutných k řešení systémů popsaných parciálními diferenciálními rovnicemi.

Obsahuje:

- editor pro přípravu skriptu,
- generátor sítě pro vytvoření modelu pro metodu konečných prvků,
- řešič na principu metody konečných prvků k nalezení řešení,
- a grafický systém pro vykreslení výsledku.

Uživatel může skript upravovat, spouštět úlohu a pozorovat výstup, poté opravit skript a opět jej spustit, aniž by musel opustit prostředí aplikace FlexPDE.

FlexPDE nemá předdefinované úlohy ve svém hlavním okně ani na samotném úkolovém listu. Výběr parciální diferenciální rovnice je tak zcela na uživateli.

Skriptovací jazyk FlexPDE je „nativní“ jazyk. To dovoluje uživateli popsat matematiku v jeho systému parciálních diferenciálních rovnic a hlavní geometrii samotné úlohy ve tvaru podobném tomu, jaký by použil při popisu dané úlohy svému kolegovi. Pro názornost uvádím sekci ROVNICE ve skriptu, kde je popsána Laplaceova rovnice:

$$\text{Div}(\text{grad}(u)) = 0. \quad (1.4)$$

nebo sekci HRANICE ve skriptu, kde jsou popsány hranice geometrického objektu v 2 dimenzionální úloze jako chůze podél hranice tohoto objektu:

Start(x1,y1) line to (x2,y1) to (x2,y2) to (x1,y2) to close

Tato podoba skriptu má mnoho výhod:

- Skript kompletně popisuje systém rovnic a samotnou hlavní úlohu. Nemůže tedy nastat nejasnost ohledně toho, jaká rovnice je právě řešena, což je poměrně častým problémem u jiných aplikací.
- Nové proměnné, nové rovnice či nová značení jsou přidávána jako budoucí stav, tudíž nemůže nikdy nastat situace, že by program prezentoval stará (či ztracená) data, což by mělo různý fyzikální vliv na danou úlohu.
- Mnoho rozličných úloh se může řešit tím samým programem, tudíž odpadá nové učení se spojení pro každou úlohu.

Možnost využít uvedených výhod skriptu však vyžaduje jeden nezbytný předpoklad: uživatel musí být schopen zadat svou úlohu jako matematický popis.

Pro výukovou formu je vše výše popsané velmi dobré. Je to věc, kterou se studenti chtějí naučit. V průmyslové výrobě může jedinec, který ovládá zadávání úloh v podobě matematického popisu připravit skript, který bude využíván v modifikovaných podobách ostatními v této věci méně znalými pracovníky. Takto vytvořená knihovna aplikačních skriptů jim může ukázat, jak se co má provést.

Na internetu je k tomuto programu podrobně psaná uživatelská příručka a manuál ke stažení. Manuál v tištěné podobě je automaticky zaslán k licenci pro odborníky (viz. 3.5.2 Typy licencí). Internetové stránky obsahují i další příručky zaměřené na konkrétní druhy systémů popsaných parciálními diferenciálními rovnicemi. (*informace čerpány z [12]*)

3.4.1. Systémové požadavky

Operační systém	Architektura	Adresování
Windows 98/ME/NT/2000/XP	x86	32 bit
Linux86	x86	32 bit
MacIntosh OS X PPC	10.3 Panther a vyšší	32 bit
Windows XP –x64	AMD64, IA32 s EM64T	64 bit
Linux86 64	AMD64, IA32 s EM64T	64 bit
MacIntosh OS X INTEL	10.4 Tiger a vyšší	64 bit

Tab.3.9: FlexPDE 5.0.15 – systémové požadavky

(*informace čerpány z [12]*)

3.4.2. Typy licencí

FlexPDE verze 5 je dostupný ve třech licenčních konfiguracích:

- a) *Licence k posouzení programu* – Tato konfigurace je zdarma. Vyřeší pouze úlohy, které jsou uznány jako klíčové organizací PDE Solutions Inc. Všechny tyto úlohy běží v plnohodnotném módu, bez jakéhokoliv omezení (uživatel má možnost změnit jakékoliv klíčové číselné údaje v demo úlohách bez rizika znehodnocení těchto klíčů!).
- b) *Studentská licence* – Studentská konfigurace je zdarma. Spustí všechny testové úlohy, které obsahuje demonstrační verze. K tomu vyřeší jakýkoliv uživatelem zadaný problém, který má 5 a nebo méně rovnic a využívá méně jak 100, 800 nebo 1800 síťových uzlů (1D, 2D a 3D). Přes zmíněná omezení může být takto řešeno

velké množství významných problémů z vědy a techniky. Studentská konfigurace podporuje 1D, 2D a 3D zobrazení.

- c) *Odborná licence* – Profesionální konfigurace otevře plnou sílu programu FlexPDE verze 5. Má neomezenou účinnou plochu sítě a neomezený počet čísel pro simultánní rovnice (tento počet je omezen pouze samotnými zdroji PC sestavy). Profesionální konfigurace může být licencována pouze na vykreslování 1D či 1D+2D nebo 1D+2D+3D. V této verzi obdržíte tištěný manuál a instalační CD.

Volná zkušební – Pro ty, kteří si chtějí FlexPDE vyzkoušet kompletně pomocí vlastních kompletních úloh, je k dispozici 30 denní volná odborná licence.

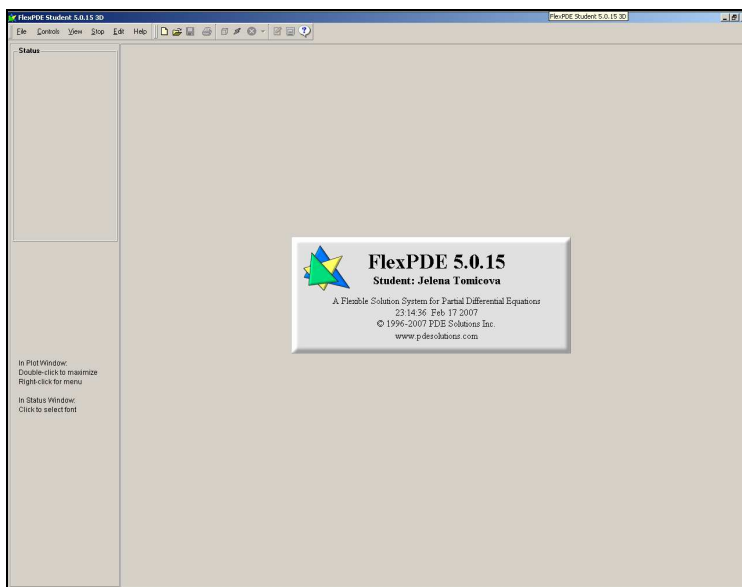
- Stáhněte a nainstalujte si verzi k posouzení k programu, spusťte a klikněte na *Help/Register*.
- Opište si počítačové identifikační číslo (ID) které se objeví v registračním dialogu.
- Napište ID do formuláře umístěného na internetu (<http://pdesolutions.com/cgi-bin/trial51>).
- Uznaná žádost bude vyřízena zasláním klíče prostřednictvím e-mailu. (*informace čerpány z [12]*)

4. Návod pro FlexPDE 5.0.15

4.1. Základní seznámení s programem

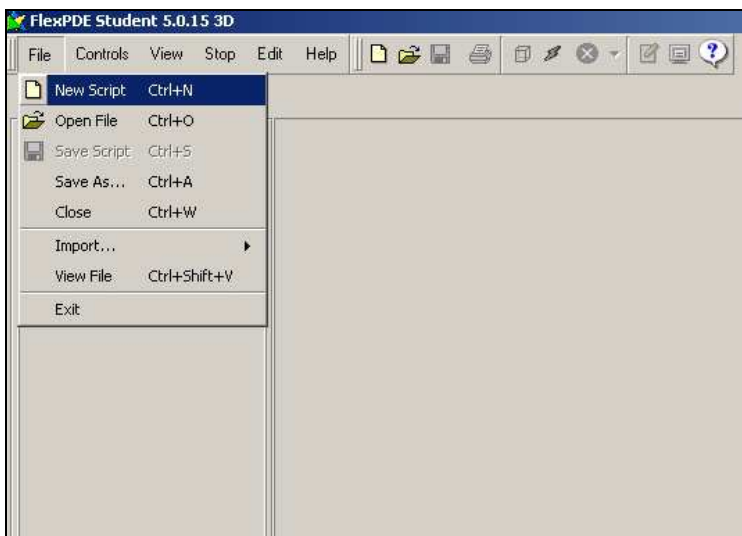
Program se nainstaluje pomocí souboru „*fpde5015wintel.exe*“ umístěném v adresáři *FlexPDE5* na příloženém CD. Po instalaci do kořenového adresáře je nutné zkopírovat soubor „*flexpde5key.txt*“, a to bez jakékoliv úpravy. Tím je program FlexPDE pro studijní účely nainstalován a plně zprovozněn.

Po spuštění programu se objeví uvítací obrazovka:



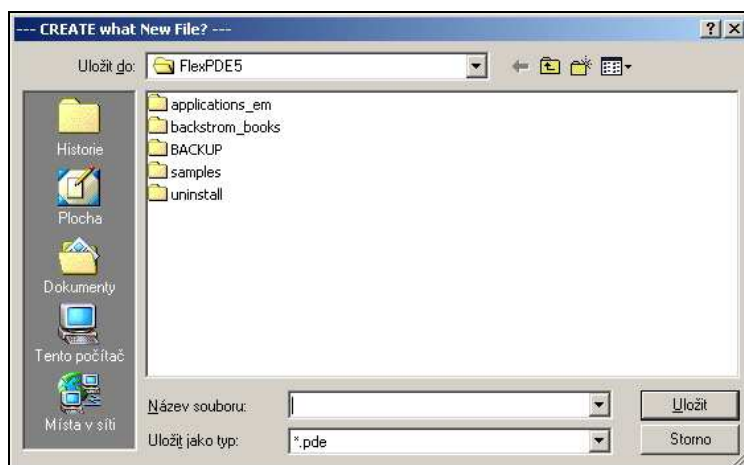
Obr.4.1.: FlexPDE 5.0.15 - Spuštění programu

Pro vytvoření našeho vlastního skriptu použijeme nabídku *File* -> *New script* a nebo klávesovou zkratkou *CTRL+N*.



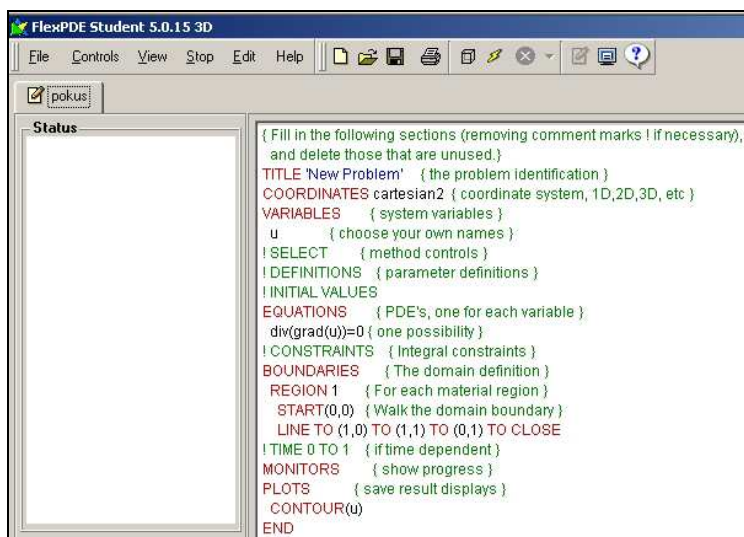
Obr.4.2.: FlexPDE 5.0.15 - Tvorba nového skriptu

Jako první je třeba skript pojmenovat a uložit jej do námi zvolené pozice. Není nutno skript ukládat do stejné adresářové struktury ve které se nacházejí vzorové skripty programu samotného.



Obr.4.3.: FlexPDE 5.0.15 - Uložení nového skriptu

Poté se objeví obrazovka rozdělená do dvou částí. Vlevo můžeme sledovat probíhající činnosti programu při spuštění skriptu a vpravo máme možnost kód editovat.



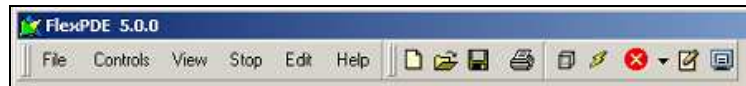
Obr.4.4.: FlexPDE 5.0.15 - Editace skriptu

Automaticky dojde k předgenerování základní struktury skriptu. Ta je členěna na následující základní části:










- *TITLE* - název skriptu (doporučuje se volit výstižný pojem charakterizující samotnou řešenou problematiku),
- *COORDINATES* - volba souřadnicového systému,
- *VARIABLES* - definice proměnných, které budou použity,
- *EQUATIONS* – oblast pro vlastní definici parciální diferenciální rovnice
- *BOUNDARIES* – definice hlavního prostoru na nějž je rovnice aplikována,

- *MONITORS* – oblast, ve které se objevuje zpracování a námi zkoumaný postup jevu samotného,
- *PLOTS* – definice výsledku, který chceme pozorovat (vykreslit),

K obsluze programu může uživatel používat nabídky pod klíčovými slovy a nebo panel nástrojů. Jednotlivá tlačítka samozřejmě odpovídají vybraným částem z textových nabídek.



Obr.4.5.: FlexPDE 5.0.15 - Panel nástrojů

	New Script	- z nabídky File	- vytvoření nového skriptu
	Open Script	- z nabídky File	- otevře již existující skript
	Save Script	- z nabídky File	- uloží námi editovaný skript
	Print Script	- z nabídky Edit	- otevře nabídku k tisku
	Domain Review	- z nabídky Controls	- zkontroluje správnost hlavní části skriptu
	Run Script	- z nabídky Controls	- spustí skript
	Stop Now	- z nabídky Stop	- pomocí šipky rozbalí Stop menu a zastaví provádění skriptu
	Show Editor	- z nabídky Controls	- zobrazení kódu v pravé části obrazovky
	Show Plots	- z nabídky Controls	- zobrazení vykreslených objektů

Samotné ovládání programu je značně intuitivní. Další informace o základním nastavení lze získat v souboru *getting_started.pdf* v adresáři *FlexPDE5* na přiloženém CD.

4.2. Řešení zadané rovnice (1.1)

Celý kód je uveden v Příloze C a na příloženém CD jako soubor *1.1_FlexPDE.pde*. V této části bakalářské práce bude v požadovaném rozsahu kód vysvětlen tak, aby byl běžnému uživateli zcela pochopitelný.

```
{ 1.1_FlexPDE.PDE }
{
  *****
  The spreading of heat in the panel.
  Author: Jelena Tomicova
  *****
}

TITLE
"Straightforward heat"

COORDINATES
cartesian2                {cartesian system 'x' and 'y'}

SELECT
spectral_colors            {values from red to violet}

VARIABLES
u(threshold=1000)         {'u = temperature' with maximum of 1000 [K]}

DEFINITIONS
width = 5                  {'X = width' coordinate of the panel [m]}
long = 10                  {'Y = long' coordinate of panel [m]}
K = 1                      {constant number [m^2*s^-1]}
Q = 30                     {thermal source [K*s^-1]}
tmax = 100                 {time - plot range control [s]}
uindoor = 20               {indoor heating [K]}

INITIAL VALUE
u = uindoor

EQUATIONS
dt(u) = div(k*grad(u)) + Q { the heat equation }
```

Obr.4.6.: FlexPDE 5.0.15 – První část skriptu (1.1)

Určitou nevýhodou skriptu může být, že veškeré komentáře ke kódu jsou uváděny pouze v anglickém jazyce (vezmeme-li však v úvahu průměrnou lingvistickou vybavenost uživatelů tohoto druhu software, můžeme vycházet z předpokladu, že jednotlivé komentáře by neměly být pro uživatele nepochopitelné). V samotné hlavičce je uveden název skriptu. Poté je třeba definovat souřadnicový systém. V námi řešeném příkladu se jedná o kartézský systém souřadnic, neboť deska je popsána pouze šířkou a délkou. V části *VARIABLES* jsou definovány ty proměnné, jež jsou v rovnici řešeny. V tomto konkrétním případě se jedná o neznámou u (teplo), pro niž je stanoveno omezení, a to takové, že její hodnota nemá být vyšší než $1000K$. Zbylé zadávané proměnné jsou definovány v následující části včetně jejich číselných, případně parametrických hodnot. Pro praktické řešení zadané parciální diferenciální rovnice jsem zvolila šířku desky $5m$, délku desky $10m$, konstantu κ rovnu 1 , zdroj tepla jako $30 K/s$. V části *INITIAL VALUE* jsou dále určeny vstupní podmínky, což může být pro hodnotu u např. u_{indoor} okolní teplota vzduchu, která je rovna cca $20K$. Následuje prepis parciální diferenciální rovnice.

```

BOUNDARIES                                     {he domain definition}
REGION 1                                       {'desk' only one region}

START(0,0)                                     { specify the starting point }
VALUE(u)=uindoor                              {walk the boundary}
  LINE TO (width,0)
NATURAL(u)=normal(k*grad(u))
  LINE TO (width, long)
NATURAL(u)=normal(k*grad(u))
  LINE TO (0,long)
  LINE TO CLOSE                               {bring boundary back to starting point}

TIME 0 TO 20 BY 1                             {time dependent [s]}

MONITORS
for t=0 by 1 to 20
  elevation(u) from (0,0) to (width,0) as "axis X_bottom Temperature" range=(0,tmax)
  elevation(u) from (width, long) to (0,long) as "axis X_top Temperature" range=(0,tmax)
  elevation(u) from (0,0) to (0,long) as "axis Y_left Temperature" range=(0,tmax)
  elevation(u) from (width,0) to (width, long) as "axis Y_right Temperature" range=(0,tmax)

```

Obr.4.7.: FlexPDE 5.0.15 – Druhá část skriptu (1.1)

V rámci části *BOUNDARIES* je uveden popis celé oblasti, a to jako průchod po jejích hranicích z počátečního vrcholového bodu do následujícího do té doby, než se podaří celou oblast „obejít“ a opět skončit v počátečním bodě. Zároveň jsou zde pomocí *VALUE(u)* nadefinovány počáteční podmínky, které se shodují s již uvedenou počáteční podmínkou v části *INITIAL VALUE*, a zároveň okrajové podmínky pomocí *NATURAL(u)*. Zápis

$$NATURAL(u)=normal(k*grad(u)) \quad (1.5)$$

plně odpovídá zápisu zadané termální parciální diferenciální rovnice. Pokud by zde byl uveden pouze číselný údaj, znamenalo by to, že číselný počet jednotek bude při vykreslení přičten ke každé časové jednotce bez ohledu na pravou hodnotu *u*. V úseku *TIME* jsem navolila časový interval 20 sekund, který je krokován po 1 sekundě. Při samotném průběhu a zároveň na konci procesu lze sledovat hodnoty teploty na obryse desky (*MONITORS*).

```

PLOTS                                           {write these plots to disk at completion}
for t=0 by 1 to 20
  CONTOUR(u) AS "u_heat - contour"             {show the solution}
  SURFACE (u) AS "u_heat - surface"           {show a surface plot}
  VEKTOR(k*grad(u)) AS "Heat Flow"

SUMMARY AS "Summary"
  REPORT(Q) AS "Thermal source"
  REPORT(width) AS "Width of panel"
  REPORT(long) AS "Long of panel"
  REPORT(k) AS "Kapa"

HISTORIES
HISTORY(u) AT (0,0) (width/2,0) (width,0)
              (width,long/2) (width,long)
              (width/2,long) (0,long)
              (0,long/2) AS "History"

END {end of descriptor file}

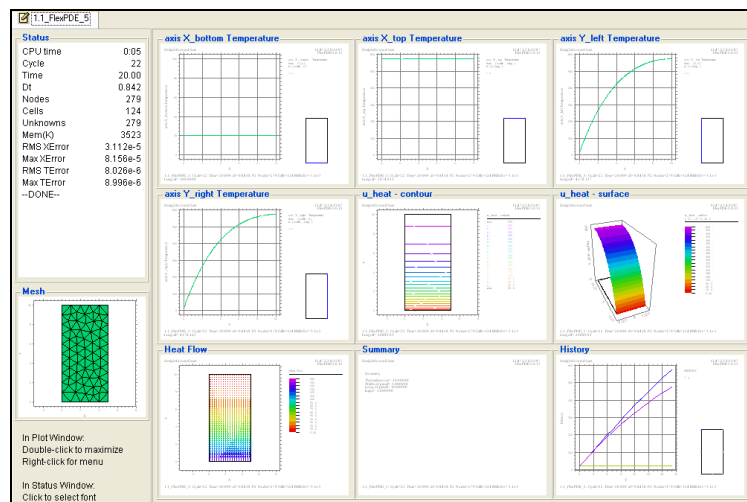
```

Obr.4.8.: FlexPDE 5.0.15 – Třetí část skriptu (1.1)

Kromě procesů probíhajících v průběhu procesu na okrajích desky je žádoucí, aby uživatel po skončení algoritmu viděl výsledné *u*, které je v programu zobrazeno pomocí vrstevnic (*contour*) a hladin (*surface*). Funkce *VEKTOR* následně dovoluje vykreslit pomocí vektorů směr šíření tepla uvnitř desky.

S pomocí sekce *SUMMARY* můžeme vygenerovat textový výpis, který ale není vždy nezbytně nutný. V rámci našeho řešení je však pro názornost programových funkcí takový popis uveden. V jeho rámci tak lze vypsát veškeré hodnoty – zdroj tepla, šířku a délku desky, a hodnotu κ .

Ke stejnému účelu slouží také funkce *HISTORY*, která vykresluje změnu tepla v krajích desky v průběhu celého časového intervalu. Celý skript je zakončen funkcí *END*.



Obr.4.9.: FlexPDE 5.0.15 – Výsledek (1.1)

Po spuštění skriptu pomocí funkce *Run Script* má uživatel k dispozici již zmiňované výstupy. Každé okno lze dvojklikem levého tlačítka myši maximalizovat do celého okna v pravé části programu, stejným postupem lze okno opět zmenšit do původní velikosti. Na prvních čtyřech obrázcích je pozorovatelný postup teploty na jednotlivých okrajích desky, dále výsledná teplota u ve vrstevnicovém i hladinovém znázornění, následuje *SUMMARY* s vypsáním námi nadefinovanými položkami a *HISTORY* s vývojem teplot v jednotlivých bodech desky.

Skript je napsán tak, aby bylo možné vždy zaměřit pouze jeden číselný údaj a pozorovat vliv jeho změny na výsledné hodnoty. Při zadávání údajů však byl dáván maximální pozor na to, aby nebylo nutné jednotlivé údaje přepisovat i v dalších částech skriptu.

4.3.Řešení zadané rovnice (1.2)

Celý kód je uveden v Příloze D a na přiloženém CD jako soubor *1.2_FlexPDE.pde*. Řešení rovnice (1.2) je o něco komplexnější problém, neboť FlexPDE, zrovna tak jako ostatní v práci zmiňované programy, neumí jednoduše řešit parciální diferenciální rovnice čtvrtého řádu. Nejprve je tedy nutné vysvětlit princip zapsání samotné rovnice. Další text bude zaměřen na jednotlivé části kódu, které jsou opět okomentovány v anglickém jazyce (tj. opěr se předpokládá dostačující jazyková vybavenost uživatele).

Rovnici (1.2) je třeba nejprve přepsat do tvaru srozumitelného pro program, přestože je známo, že zadat ji k samotnému řešení v takovémto tvaru nelze.

$$dxxxx(w)+2*dxyy(w)+dyyyy(w) = del2(del2(w)). \quad (1.6)$$

V první řadě je nezbytné nadefinovat, že:

$$V=del2(W). \quad (1.7)$$

Díky uvedené definici se získají dvě rovnice:

$$del2(W)=V, \quad (1.8)$$

$$del2(V) + (rho/D)*dt(W) = q/D. \quad (1.9)$$

Jestliže U je definováno jako:

$$U=dt(W), \quad (1.10)$$

výsledkem jsou tyto 3 následující rovnice:

$$W: dt(W)=U, \quad (1.11)$$

$$V: del2(W)=V, \quad (1.12)$$

$$U: del2(V) + (rho/D)*dt(U) = q/D. \quad (1.13)$$

V tomto tvaru je program FlexPDE schopen rovnice vyřešit.

```

{ 1.2_FlexPDE.PDE }

{
*****
The deflection of the panel.
Author: Jelena Tomicova
***** }

TITLE 'Deflection of panel'

SELECT
errlim = 0.005           {error tolerance - error in any variable
                          is less than 0.1% over every cell of the
                          mesh}
cubic                   {Use Cubic Basis}

VARIABLES
U(0.1)
V(0.1)
W(0.1)

DEFINITIONS
width = 5                {width of panel [m]}
long = 10                {long of panel [m]}
h = 2                    {thickness of panel [m]}
L = 1.0e4                {large number}
E = 29e6                 {Youngs modul [N/m^2]}
Q = 14.7                 {load distribution [N/m^2]}
nu = 0.3                 {Poissons ratio [-]}
D = E*h^3/(12*(1-nu^2))
rho = 1                  {density of panel [kg/m^2]}

INITIAL VALUES
W = 0
U = 0
V = 0

```

Obr.4.10.: FlexPDE 5.0.15 – První část skriptu (1.2)

Jako první je definován *TITLE* název samotného skriptu. V sekci *SELECT* se využívá příkaz *ERRLIM*. Jeho základní hodnota, která je nastavena na *0,001* říká, že základní síť objektu bude průběžně modifikována, dokud se nedocílí odhadované chyby v jakékoliv hodnotě menší než *0,1%* přes každou buňku sítě. V řešeném případě byla zvolena hodnota *0,005*, tj. *0,5%* chyba. Dále byl navolen kubický souřadnicový systém a nadefinovány vstupní veličiny *U*, *V* a *W*, pro něž byly vytvořeny výše uvedené tři rovnice. V sekci *DEFINITIONS* je třeba definovat zbylé hodnoty, ať už přesným číslem, nebo dalším parametrickým zadáním. Pro řešení zadaného příkladu byla definována šířka panelu *5* metrů, délka *10* metrů a tloušťka *2* metry. Vzhledem ke skutečnosti, že panel je pevně upevněn, musí pro jeho okrajové podmínky platit:

$$dU/dn = 0, \tag{1.14}$$

$$dV/dn = L*U, \tag{1.15}$$

$$dW/dn = 0. \tag{1.16}$$

Z tohoto důvodu je definováno číslo *L*, a to tak, aby jeho hodnota byla velmi vysoká. Dále jsou určeny Youngův modul $E=29e6$ a Poissonovo číslo rovno *0,3*. Při řešení se vychází z toho, že panel je prezentován jako prvek z kovu (pro kov je Poissonovo číslo definováno jako *0,3*, pro gumu neboli téměř nestlačitelný materiál má hodnotu *0,5* a pro materiál podobný korku je rovno *0*). Počáteční hodnoty vstupních veličin jsou rovny *0*.

```

EQUATIONS
W: dt(W)=U
V: del2(W)=V
U: del2(V) + (rho/D)*dt(U) = q/D

BOUNDARIES
Region 1                                {panel' just one region}
start (0,0)
natural(W) = 0
natural(U) = 0
natural(V) = L*U
line to (width,0)
to (width,long)
to (0,long)
to close

TIME 0 to 5 by 0.01                      {time dependent [s]}

MONITORS
for cycle = 1                            {refreshing for every number of time step}
!for t=0 by 0.01 to 5                    {for each number, the best step was found 0.01}
elevation(U) from (0,long/2) to (width,long/2) as "Displacement_in_the_middle"

PLOTS
for cycle = 1                            {refreshing for every number of time step}
!for t=0 by 0.01 to 5                    {for each number, the best step was found 0.01}
contour (U) as "Displacement_contour"
surface(U) as "Displacement_surface"

SUMMARY as "Summary"
report(width) as "Width of panel"
report(long) as "Long of panel"
report(h) as "Thickness of panel"
report(rho) as "Density of panel"
report(nu) as "Poissons ratio"
report(U) as "Displacement"

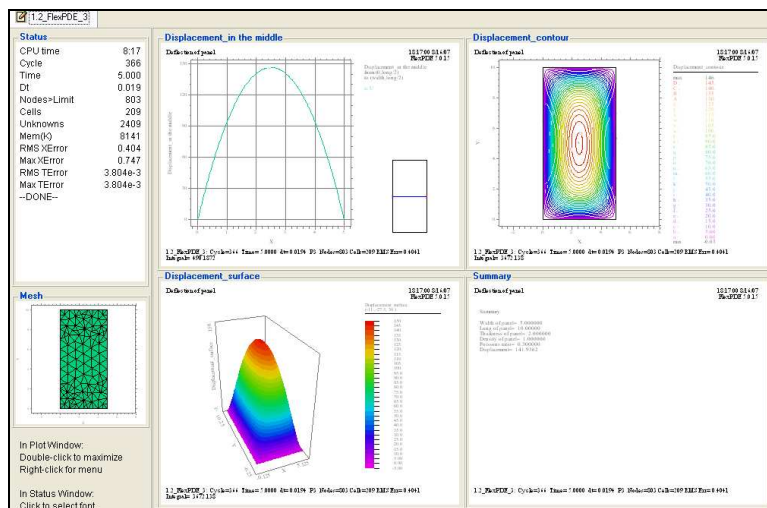
END                                       {end of descriptor file}

```

Obr.4.11.: FlexPDE 5.0.15 – Druhá část skriptu (1.2)

V sekci *EQUATIONS* vidíme zápis tří v úvodu odvozených rovnic. V sekci *BOUNDARIES* je definován náš panel, a to již popsáním způsobem (průchod po jeho obvodu). Funkcí $NATURAL(U)=0$, $NATURAL(V)=L*U$ a $NATURAL(W)=0$ jsou definovány okrajové podmínky.

Úloha je řešena v časovém intervalu 20 sekund, kde se jako nejvhodnější krok zjištěný postupnými experimenty jeví hodnota 0,01 (pokud zvolíme jiný, ne zcela vhodný interval, program poběží a zobrazí chybu *STEP FAILED. RETRY*. Jeho chod se nezastaví, ale software začne hledat novou, vhodnější hodnotu. Po jejím nalezení pracuje právě s touto hodnotou a nikoliv s hodnotou definovanou uživatelem). V sekci *MONITORS* je možno sledovat sílu U na křivce, která je vedena středem panelu, a v sekci *PLOTS* je pak nadefinováno výsledné zobrazení U (opět jak ve vrstevnicích, tak v hladinách). V tomto skriptu je použit vykreslovací krok po každém cyklu, pomocí funkce *for cycle = 1*. Číslo odpovídá cyklům, které mají uběhnout, než se vykreslený objekt zobrazí. V komentované části je také uveden zápis pomocí funkce *for t=0 by 0,01 to 5*. Na závěr skriptu je opět využito sekce *SUMMARY* pro výpis zadávaných hodnot.



Obr.4.12.: FlexPDE 5.0.15 – Výsledek (1.2)

Po spuštění skriptu může uživatel sledovat postupné vykreslování jednotlivých grafů. Doba vykreslování je o něco delší, což je důsledkem vyšší náročnosti rovnice na výpočet.

Skript je napsán v co nejjednodušší formě, tedy tak, aby bylo možné vždy zaměnit pouze jeden číselný údaj a pozorovat vliv jeho změny na výsledné hodnoty. Při zadávání údajů byl dáván maximální pozor na to, aby nebylo nutné jednotlivé údaje přepisovat i v dalších částech skriptu samotného.

5. Závěr

Pro ověření možností nalezených freewarových programů v praxi jsem se omezila na výběr čtyř z nich: software Getfem++, Elmer, FreeFem++ a FlexPDE. Všechny programy jsem nainstalovala a následně provedla testování řešení jednoduchých matematických operací. Touto cestou jsem zjistila, že programy Getfem++ a Elmer jsou značně náročné na využití paměti běžného uživatelského PC. V případě programu Getfem++ byla uvedená skutečnost zapříčiněna především jeho provázaností s prostředím Matlab, což mělo za důsledek nadměrnou zátěž systému - program v době jeho použití zabíral největší množství fyzické paměti počítače (bráno včetně programu Matlab).

Před konečným výběrem programu pro výpočet zadaných rovnic jsem sledovala dění na domovských stránkách autorů softwaru, abych zjistila, zda a jak jsou tyto programy nadále přizpůsobovány potřebám studentské, vědecké i technické veřejnosti. Po porovnání prvních získaných verzí programů s jejich následným upgrade mohu konstatovat, že jednoznačně nejslabší vývojový pokrok byl zaznamenán u programu Elmer, naopak nejvíce dopředných vývojových kroků bylo dosaženo u programu FlexPDE.

V průběhu června jsem ověřila funkčnost nabízeného poradenství, a to pomocí e-mailové komunikace (odpovídající adresy jsem získala v kontaktních odkazech na stránkách jednotlivých producentů testovaného softwaru). Nejkratší čekací dobu na odpověď vykazala společnost PDE Solutions Inc (program FlexPDE), a to 4 hodiny od odeslání žádosti. Výjimkou byl program Getfem++, od jehož technické podpory nedošla odpověď žádná. V této souvislosti považuji za vhodné připomenout, že program FlexPDE má i své vlastní diskusní fórum, v jehož rámci je na jakékoliv problémy reagováno do 24 hodin.

Mám-li shrnout uvedená fakta, je nesporné, že – při prakticky stejné úrovni požadovaných výstupů – z hlediska nároků na uživatelské prostředí, vývoje software i z hlediska technické programové podpory je pro účely této práce i pro běžnou výuku nejvíce použitelný program FlexPDE, pro který jsem v rámci bakalářské práce sestavila manuál (v rozsahu, který je pro tuto práci nezbytný).

Bohužel, z objektivních důvodů se mi nepodařilo sestavit dostatečně kvalitní materiál pro popis zbylých programů. V případě programu Getfem++ vznikly potíže s konfigurací PC, neboť náročnost na využití zejména paměti byla natolik vysoká, že jakékoliv delší testování bylo jednak nemožné (program dokonce několikrát zablokoval funkčnost celého počítače), jednak z časového hlediska neproveditelné. U programů FreeFem++ i Elmer se podařilo

vyřešit zadanou rovnicí (1.1). Řešení této úlohy jsem docílila díky využití manuálů a ukázkových úloh pro oba dva druhy testovaného softwaru. Při aplikaci programu Freefem++ jsem využila především kapitolu 3.4 *Thermal Conduction* v manuálu *freefem++doc.pdf*, který lze nalézt na přiloženém CD, u programu Elmer je pak využitelná zejména kapitola *Tutorial 1 – Temperature distribution* v manuálu *ElmerTutorials.pdf*, který je rovněž součástí sestavy souborů na přiloženém CD. Jako problematické je třeba vyhodnotit celkové řešení zadané rovnice (1.2), která je čtvrtého řádu a jejíž zjednodušení se nedařilo aplikovat. Vzhledem ke skutečnosti, že návod pro zbylé dva programy byl neúplný, takže i řešení diferenciální parciální rovnice bylo neúplné, do této práce jsem popis využitelnosti aplikovaného softwaru nezařadila.

Všechny testované programy vedly ke správnému vyřešení zadaných parciálních diferenciálních rovnic, proto je pro praktické využití softwaru skutečně nezbytně nutné zvážit zejména nároky na uživatelské prostředí.

V této práci, má čtenář k dispozici manuál ke čtyřem programům, s jejichž pomocí lze řešit parciální diferenciální rovnice, přičemž již výše uvedený program FlexPDE disponuje detailním návodem k použití. Součástí práce je také CD, na kterém jsou - kromě práce samotné a spouštěcích skriptů k vyřešení zadaných rovnic – umístěny i potřebné soubory a manuály pro všechny uváděné softwary. CD obsahuje i v textu zmiňované doplňky pro systém Windows – jejich nahrání na CD vychází ze skutečnosti, že převážná část veřejnosti stále pracuje v prostředí tohoto operačního systému.

6. Použité zdroje

6.1.Literatura

- [1] Miroslav Dont (2001): Úvod do parciálních diferenciálních rovnic, Vydavatelství ČVUT, Praha, ISBN 80-01-01676-5
- [2] Pavel Kočička, Filip Blažek (2004): Praktická typografie, Computer Press, Brno, ISBN 80-722-6385-4

6.2.Elektronické zdroje

- [3] BACKSTROM, Gunnar. Incomparable books for demystifying mathematics and the fields of physics [online]. Dostupný z WWW: <<http://web.telia.com/~u40124320/index.htm>>.
- [4] doc. Ing. Jiří BURŠA PhD. – Metoda konečných prvků
- [5] GEUZAINÉ, Christopher, REMACLE, Jean-François. Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities [online]. July 13 2007. Dostupný z WWW: <<http://www.geuz.org/gmsh/>>.
- [6] HECHT, Frédéric, LE HYARIC, Antoine. Freefem++ : Home Page [online]. Sat Aug 4 22:53:03 2007. Dostupný z WWW: <<http://www.freefem.org/ff++/index.htm>>.
- [7] HECHT , Frédéric, SALTEL, Eric. Emc2 : (english) [online]. May 19 1998. Dostupný z WWW: <<http://pauillac.inria.fr/cdrom/www/emc2/eng.htm>>.
- [8] P.W.HEMKER - Discretisation of PDEs, Finite Element Method
- [9] Ing. Miroslav ŠPANIEL, CSc. – Metoda konečných prvků
- [10] Compaq Visual Fortran : HP [online]. Dostupný z WWW: <http://h21007.www2.hp.com/dspp/tech/tech_TechSoftwareDetailPage_IDX/1,1703,7088,00.html>.
- [11] Elmer : CSC [online]. CSC, 31.5.2007. Dostupný z WWW: <<http://www.csc.fi/elmer/>>.
- [12] FlexPDE finite element model builder for Partial Differential Equations [online]. 6/24/07. Dostupný z WWW: <<http://www.pdesolutions.com/index.html>>.
- [13] Getfem++ : HomePage [online]. Dostupný z WWW: <http://home.gna.org/getfem/getfem_intro.html>.

- [14] GiD : The personal pre and postprocessor [online]. CIMNE, 2006. Dostupný z WWW: <<http://gid.cimne.upc.es/>>.
- [15] Min GV : Home [online]. www.mingw.org, 2004. Dostupný z WWW: <<http://www.mingw.org/>>.
- [16] Matematicko fyzikální fakulta – Úprava bakalářské práce
- [17] Python Programming Language : Official Website [online]. [1990] , Fri, 18 May 2007 13:00 +0100 Dostupný z WWW: <<http://www.python.org/>>.
- [18] Wikipedie : Otevřená encyklopedie [online]. 19:54, 30.7.2007 Dostupný z WWW: <<http://cs.wikipedia.org/>>.
- [19] Wikipedia : The Free Encyklopedia [online]. 11.8.2007. Dostupný z WWW: <<http://en.wikipedia.org/>>.
- [20] ÚM, FSI VUT v Brně – Základní pojmy a typy PDR

7. Příloha A

Seznam obrázků

Obr.4.1.:	FlexPDE 5.0.15 - Spuštění programu	22
Obr.4.2.:	FlexPDE 5.0.15 - Tvorba nového skriptu	22
Obr.4.3.:	FlexPDE 5.0.15 - Uložení nového skriptu	23
Obr.4.4.:	FlexPDE 5.0.15 - Editace skriptu	23
Obr.4.5.:	FlexPDE 5.0.15 - Panel nástrojů	24
Obr.4.6.:	FlexPDE 5.0.15 – První část skriptu (1.1)	25
Obr.4.7.:	FlexPDE 5.0.15 – Druhá část skriptu (1.1)	26
Obr.4.8.:	FlexPDE 5.0.15 – Třetí část skriptu (1.1)	26
Obr.4.9.:	FlexPDE 5.0.15 – Výsledek (1.1)	27
Obr.4.10.:	FlexPDE 5.0.15 – První část skriptu (1.2)	29
Obr.4.11.:	FlexPDE 5.0.15 – Druhá část skriptu (1.2)	30
Obr.4.12.:	FlexPDE 5.0.15 – Výsledek (1.2)	31

8. Příloha B

Seznam tabulek

Tab.3.1:	Getfem++ - systémové požadavky	13
Tab.3.2:	Python – systémové požadavky	14
Tab.3.3:	GiD – systémové požadavky	15
Tab.3.4:	GmSH – systémové požadavky.....	15
Tab.3.5:	emc2 – systémové požadavky	16
Tab.3.6:	Elmer 5.3.0 – systémové požadavky	16
Tab.3.7:	Elmer 5.3.0 – seznam překladáčů.....	17
Tab.3.8:	FreeFem++ 2.15-1 – systémové požadavky.....	18
Tab.3.9:	FlexPDE 5.0.15 – systémové požadavky	20

9. Příloha C

Obsah přiloženého CD

Elmer/ (adresář)

/Fortran/ (adresář)

/installation/ (adresář) – obsahuje archívy pro instalaci programu

/Compiling information_CSC.pdf (soubor) – návod na instalaci programu

/ElmerFrontUserGuide.pdf (soubor) – uživatelská příručka

/ElmerTutorials.pdf (soubor) – vzorové úlohy

/Manual.pdf (soubor) - manuál

FlexPDE5/ (adresář)

/Application Manual/ (adresář)

/scripts/ (adresář) – obsahuje 8 ukázkových souborů

/applications_em_r1.pdf (soubor) – manuál o elektrostatice a magnetostatice v 2D a 3D

/Documentation/ (adresář)

/getting_started.pdf (soubor) – základní manuál

/reference.pdf (soubor) – referenční příručka s příkazy

/user_guide.pdf (soubor) – uživatelská příručka

/FEM Introduction/ (adresář)

/Examples (adresář) – příklady na FEM

/FEM.pdf (soubor) – příklady řešené pomocí FEM

/FlexPDE Student License.eml (soubor) – návod a klíč ke studentské Licenci

/fpde5015wintel.exe (soubor) – instalace pro OS Windows

/flexpde5key.txt (soubor) – klíč ke studentské licenci

FreeFem/ (adresář)

/FreeFem++-2.15-1.exe (soubor) – instalace pro OS Windows

/freefem++doc.pdf (soubor) - manuál

Getfem/ (adresář)

/emc2/ (adresář) – instalace emc2

/gid/ (adresář)

/GiD8.0.9-win-x32.exe (soubor) – instalace pro OS Win x32

/GiD8-UserManual.pdf (soubor) - manuál

/gmsh/ (adresář) – program GmSh (není třeba jeho instalace)

/installation/ (adresář) – obsahuje archívy pro instalaci programu Getfem++

/MinGW-5.1.3.exe (soubor) – nainstaluje minimum GNU pro OS Win

/BasicDocumentation.pdf (soubor) – uživatelská příručka

/GetfemFEM.pdf (soubor) – FEM metoda

/python-2.5.1.msi (soubor) – instalace rozhraní Python

/python_key.txt (soubor) – informace o získání klíče a jeho použití pro
Python

/BP_tomicj1.pdf (soubor) – bakalářská práce v elektronické podobě

/1.1_FlexPDE.pde (soubor) – řešení rovnice (1.1) v programu FlexPDE 5.0.15

/1.2_FlexPDE.pde (soubor) – řešení rovnice (1.2) v programu FlexPDE 5.0.15

10. Příloha D

Vygenerovaný tiskový výstup kódu z programu FlexPDE 5.0.15 pro řešení rovnice (1.1).


```

{ 1.1_FlexPDE.PDE }

{ *****
The spreading of heat in the panel.
Author: Jelena Tomicova
***** }

TITLE
"Straightforward heat"

COORDINATES
cartesian2 {cartesian system 'X' and 'Y'}

SELECT
spectral_colors {values from red to violet}

VARIABLES
u(threshold=1000) { 'u = temperature' with maximum of 1000 [K] }

DEFINITIONS
width = 5 { 'X = width' coordinate of the panel [m] }
long = 10 { 'Y = long' coordinate of panel [m] }
K = 1 {constant number [m^2*s^-1]}
Q = 30 {thermal source [K*s^-1]}
tmax = 100 {time - plot range control [s]}
uindoor = 20 {indoor heating [K]}

INITIAL VALUE
u = uindoor

EQUATIONS
dt(u) = div[k*grad(u)] + Q { the heat equation }

BOUNDARIES {he domain definition}
REGION 1 { 'desk' only one region }

START(0,0) { specify the starting point }
VALUE(u)=uindoor
LINE TO (width,0) {walk the boundary}
NATURAL(u)=normal(k*grad(u))
LINE TO (width, long)
NATURAL(u)=normal(k*grad(u))
LINE TO (0,long)
LINE TO CLOSE {bring boundary back to starting point}

TIME 0 TO 20 BY 1 {time dependent [s]}

MONITORS
for t=0 by 1 to 20
elevation(u) from (0,0) to (width,0) as "axis X_bottom Temperature" range=(0,tmax)
elevation(u) from (width, long) to (0,long) as "axis X_top Temperature" range=(0,tmax)
elevation(u) from (0,0) to (0,long) as "axis Y_left Temperature" range=(0,tmax)
elevation(u) from (width,0) to (width, long) as "axis Y_right Temperature" range=(0,tmax)

PLOTS {write these plots to disk at completion}
for t=0 by 1 to 20
CONTOUR(u) AS "u_heat - contour" {show the solution}
SURFACE (u) AS "u_heat - surface" {show a surface plot}
VECTOR(k*grad(u)) AS "Heat Flow"

SUMMARY AS "Summary"
REPORT(Q) AS "Thermal source"
REPORT(width) AS "Width of panel"
REPORT(long) AS "Long of panel"
REPORT(k) AS "Kapa"

HISTORIES
HISTORY(u) AT (0,0) (width/2,0) (width,0)
(width,long/2) (width,long)
(width/2,long) (0,long)
(0,long/2) AS "History"

END {end of descriptor file}

```

11. Příloha E

Vygenerovaný tiskový výstup kódu z programu FlexPDE 5.0.15 pro řešení rovnice (1.2).

```
{ 1.2_FlexPDE.PDE }
```

```
{ *****  
The deflection of the panel.  
Author: Jelena Tomicova  
***** }
```

```
TITLE 'Deflection of panel'
```

```
SELECT
```

```
errlim = 0.005 {error tolerance - error in any variable  
is less than 0.1% over every cell of the  
mesh}  
cubic {Use Cubic Basis}
```

```
VARIABLES
```

```
U(0.1)  
V(0.1)  
W(0.1)
```

```
DEFINITIONS
```

```
width = 5 {width of panel [m]}  
long = 10 {long of panel [m]}  
h = 2 {thickness of panel [m]}  
L = 1.0e4 {large number}  
E = 29e6 {Youngs modul [N/m^2]}  
Q = 14.7 {load distribution [N/m^2]}  
nu = 0.3 {Poissons ratio [-]}  
D = E*h^3/(12*(1-nu^2))  
rho = 1 {density of panel [kg/m^2]}
```

```
INITIAL VALUES
```

```
W = 0  
U = 0  
V = 0
```

```
EQUATIONS
```

```
W: dt(W)=U  
V: del2(W)=V  
U: del2(V) + (rho/D)*dt(U) = q/D
```

```
BOUNDARIES
```

```
Region 1 {panel' just one region}  
start (0,0)  
natural(W) = 0  
natural(U) = 0  
natural(V) = L*U  
line to (width,0)  
to (width,long)  
to (0,long)  
to close
```

```
TIME 0 to 5 by 0.01 {time dependent [s]}
```

```
MONITORS
```

```
for cycle = 1 {refreshing for every number of time step}  
!for t=0 by 0.01 to 5 {for each number, the best step was found 0.01}  
elevation(U) from (0,long/2) to (width,long/2) as "Displacement_in the middle"
```

```
PLOTS
```

```
for cycle = 1 {refreshing for every number of time step}  
! for t=0 by 0.01 to 5 {for each number, the best step was found 0.01}  
contour (U) as "Displacement_contour"  
surface(U) as "Displacement_surface"
```

```
SUMMARY as "Summary"
```

```
report(width) as "Width of panel"  
report(long) as "Long of panel"  
report(h) as "Thickness of panel"  
report(rho) as "Density of panel"  
report(nu) as "Poissons ratio"  
report(U) as "Displacement"
```

```
END {end of descriptor file}
```

