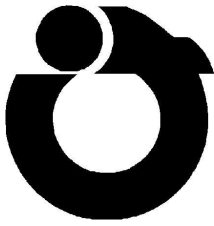


VÝZKUMNÁ ZPRÁVA Z- 1322/02



**ÚSTAV TERMOMECHANIKY**  
**Akademie věd ČR**

Dolejškova 5  
182 00 Praha 8

**PŘÍSPĚVEK KE GRAFICKÉMU  
A PARALELNÍMU ZPRACOVÁNÍ  
MOLEKULÁRNÍ DYNAMIKY**

*P. HORA, V. PELIKÁN*

## **Příspěvek ke grafickému a paralelnímu zpracování molekulární dynamiky**

*Petr Hora, Vladimír Pelikán*

### **Obsah**

<b>Seznam obrázků .....</b>	<b>4</b>
<b>Úvod .....</b>	<b>5</b>
<b>Vizualizace výsledků MD-simulace .....</b>	<b>6</b>
Vstupní formát.....	6
Popis ovládání programu .....	8
Instalace.....	14
Poznátky z provozu .....	14
Plány do budoucna .....	14
<b>Poznátky k paralelním výpočtům na Rusalce .....</b>	<b>15</b>
<b>Závěr .....</b>	<b>18</b>
<b>Literatura.....</b>	<b>19</b>
<b>Přílohy .....</b>	<b>20</b>
Příloha 1 .....	20
Příloha 2 .....	21
Příloha 3 .....	22
Příloha 4 .....	43
Příloha 5 .....	46
Příloha 6 .....	47

## **Seznam obrázků**

Obr. 1	Uživatelské rozhraní zobrazené při startu programu.....	9
Obr. 2	Dialogové okno pro načtení datového plt-souboru. ....	10
Obr. 3	Uživatelské rozhraní zobrazené po načtení dat. ....	11
Obr. 4	Grafické okno s vizualizací MD-simulace.....	12
Obr. 5	Nabídka map barev a počtu barev.....	13
Obr. 6	Nabídka map průhledností a počtu průhledností. ....	13

## **Úvod**

Tato zpráva se skládá ze dvou částí. V první kapitole je popsán program pro vizualizaci výsledků simulace problémů řešených metodou molekulární dynamiky (MD). Druhá kapitola se věnuje prvním poznatkům při paralelních výpočtech, které byly získány na Rusalce (pracovní stanice COMPAQ se čtyřmi procesory ALPHA).

Zpráva vznikla na základě podpory grantu GA AV ČR č. 10485 „Simulace křehce-tvárného chování mikrotrhlin metodou molekulární dynamiky v 2D a 3D úlohách“.

## Vizualizace výsledků MD-simulace

Jednou z vlastností MD-simulací (zejména ve 3D) je jak potřeba ohromného množství operační paměti při vlastním výpočtu, tak potřeba velkého prostoru na disku pro uložení výsledků. Vyhodnotit takové množství dat není bez účinného vizualizačního prostředku vůbec možné. I když existuje řada komerčních i „free“ programů pro vizualizaci, rozhodli jsme se pro tvorbu vlastního nástroje. Vedli nás k tomu následující okolnosti:

- Existující programy jsou pro nás zbytečně složité, neboť většinou slouží pro vizualizaci komplexních chemických struktur.
- Výstupy našich programů mají specifický formát, který by se musel převádět.
- Cizí program (i když s důkladným manuálem a zdrojovým kódem) je cizí program, není nad to napsat si něco jednoduchého vlastními silami, pokud na to stačí.

Pro tvorbu vlastního vizualizačního nástroje jsme zvolili programové prostředí MATLAB [1], které je v Akademii i na vysokých školách značně rozšířené a které už má v sobě řadu důležitých vlastností pro vizualizaci zabudovaných, např. řešení viditelnosti, rendering, průhlednost, textury, atd.

### VSTUPNÍ FORMÁT

Vstupem pro náš vizualizační nástroj je jeden z výstupních souborů z MD-simulace, což je binární soubor s formátem popsaným v

Tab. 1 na str. 7. Podrobnější popis je uveden ve zprávě [2], kde je též uveden prográmeček pro čtení tohoto souboru pomocí Fortranu. Pro tento typ souboru používáme příponu **plt**.

Tab. 1 Popis formátu vstupního souboru.

	<b>Veličina</b>	<b>Počet</b>	<b>Formát</b>	<b>Popis</b>
<b>Hlavička</b>	LB_1	1	integer*4	počet bytů následujícího bloku (116)
	M	1	integer*4	počet uložených veličin pro každý atom (5)
	N	1	integer*4	počet atomů
	Ca	1	integer*4	index speciálního (sledovaného) atomu
	Nx	1	integer*4	počet atomů základní mřížky ve směru osy x
	Ny	1	integer*4	počet atomů základní mřížky ve směru osy y
	Nz	1	integer*4	počet atomů základní mřížky ve směru osy z
	Cu	7	integer*4	(0) - hodnota konstanty Cu_inclusion, (1-3) - adresa levého, předního, dolního vrcholového atomu Cu-inkluze, (4-6) - adresa pravého, zadního, horního vrcholového atomu Cu-inkluze
	W	6	integer*4	(1-3) - adresa levého, předního, dolního vrcholu výřezu pro grafický výstup, (4-6) - adresa pravého, zadního, horního vrcholu výřezu pro grafický výstup
	C	4	integer*4	úplná adresa Ca-atomu (x,y,z,0/1)
Scales	[2,3]	real*4	(1,1) - minimální hodnota souřadnic x, (2,1) - maximální hodnota souřadnic x, (1,2) - minimální hodnota souřadnic y, (2,2) - maximální hodnota souřadnic y, (1,3) - minimální hodnota souřadnic z, (2,3) - maximální hodnota souřadnic z	
LB_1	1	integer*4	počet bytů předcházejícího bloku (116)	
<b>Krok</b>	LB_2	1	integer*4	počet bytů následujícího bloku
	krok	1	integer*4	číslo simulačního kroku
	atom	[M,N]	real*4'	(1,n) – x-ová souřadnice n-tého atomu, (2,n) – y-ová souřadnice n-tého atomu, (3,n) – z-ová souřadnice n-tého atomu, (4,n) - kinetická energie n-tého atomu, (5,n) – nenormovaný potenciál n-tého atomu
	LB_2	1	integer*4	počet bytů předcházejícího bloku

## POPIS OVLÁDÁNÍ PROGRAMU

Pro běh vizualizačního nástroje je zapotřebí těchto pěti souborů:

```
md_get_header.m  
md_get_time_cut.m  
md_viewer_gui.m  
md_viewer_gui.fig  
md_viewer_gui.cfg
```

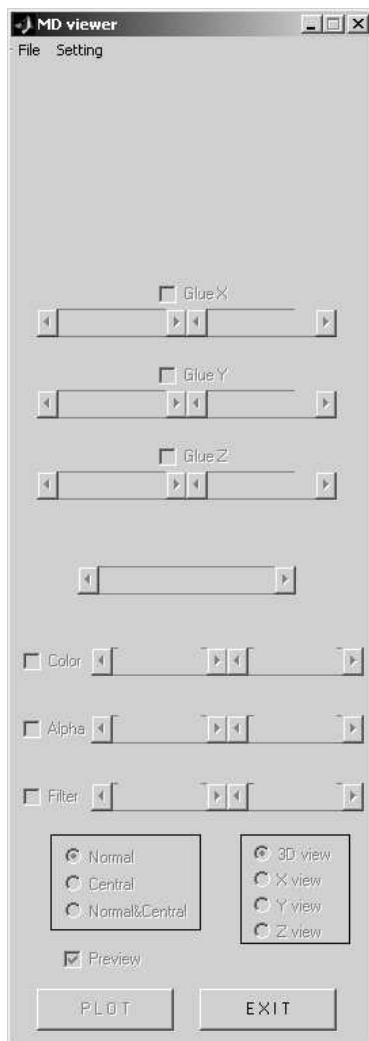
Funkce a vzájemné vazby mezi těmito soubory jsou následující:

md_get_header.m	Načte hlavičku z plt-souboru. Uživatel tento soubor nemusí spouštět. Výpis viz Příloha 1 na str. 20.
md_get_time_cut.m	Načte data pro daný simulační krok z plt-souboru. Uživatel tento soubor nemusí spouštět. Výpis viz Příloha 2 na str. 21.
md_viewer_gui.m	Spustí vlastní vizualizační program. Ze běhu volá funkce md_viewer_gui.fig, md_viewer_gui.cfg , md_get_header.m a md_get_time_cut.m. Výpis viz Příloha 3 na str. 22.
md_viewer_gui.fig	Datový soubor MATLABu. Obsahuje popis grafického uživatelského prostředí. Uživatel tento soubor nespouští. Program md_viewer_gui.m si ho načte při startu sám.
md_viewer_gui.cfg	Datový soubor MATLABu. Obsahuje pozice oken při posledním spuštění md_viewer_gui.

Pro vlastní spuštění vizualizačního prostředí stačí zadat na příkazový řádek MATLABu:

### **md\_viewer\_gui**

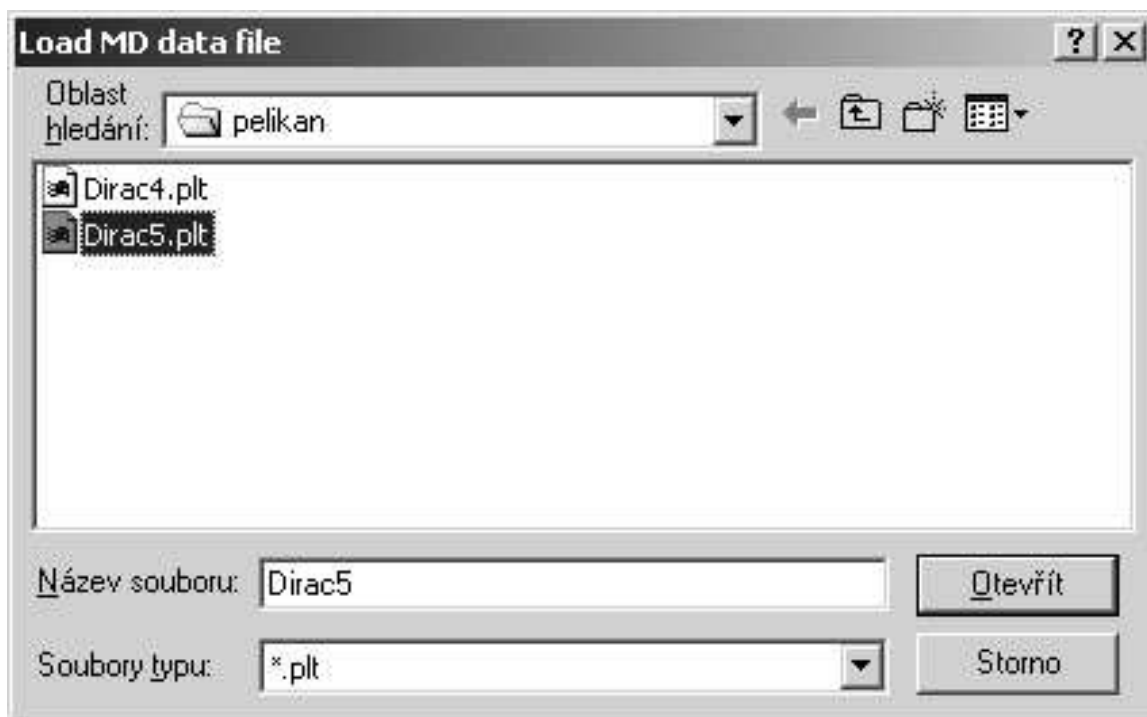
Reakcí na tento příkaz bude vykreslení uživatelského rozhraní (viz Obr. 1), na kterém jsou téměř všechny ovládací prvky zakázány.



Obr. 1 Uživatelské rozhraní zobrazené při startu programu.

Další krok spočívá v načtení plt-souboru, tj. jeden z výstupních souborů z MD-simulace, přes nabídku **File - Open**. Zobrazí se dialogové okno pro výběr plt-souboru, viz Obr. 2.



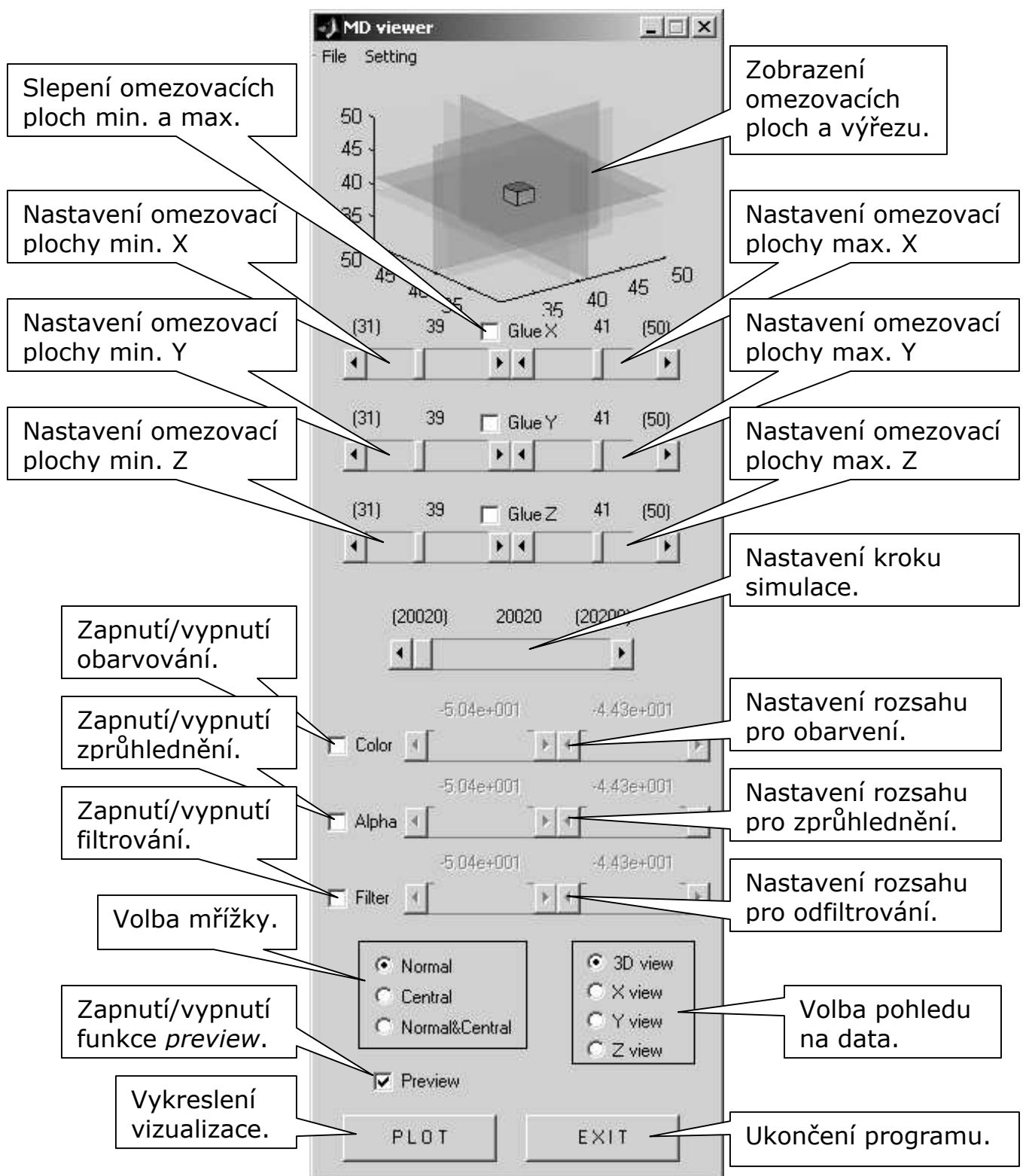


Obr. 2 Dialogové okno pro načtení datového plt-souboru.

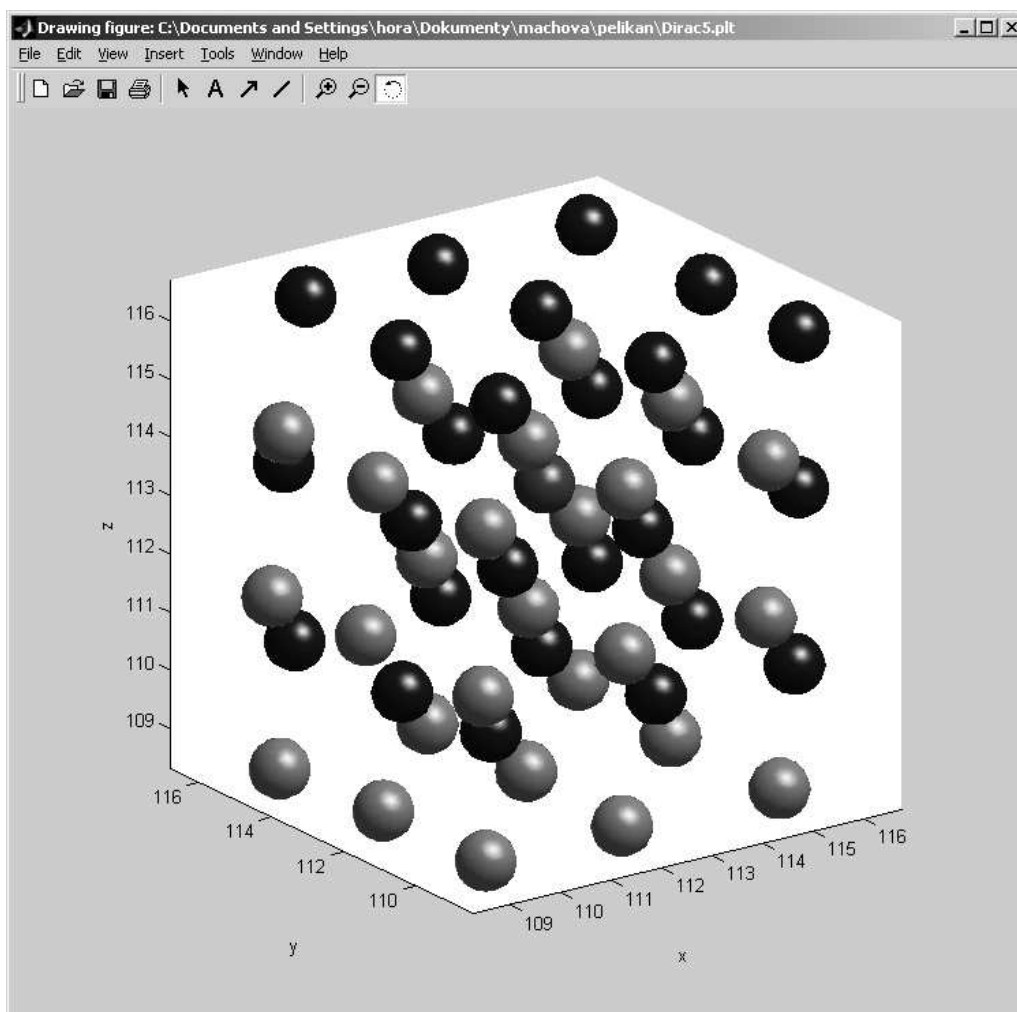
Po úspěšném načtení plt-souboru se v okně uživatelského rozhraní povolí všechny ovládací prvky (viz Obr. 3) a vykreslí se grafické okno s vizualizací (viz Obr. 4).

Nyní již můžeme přes okno uživatelského rozhraní plně zpracovávat vizualizaci MD-simulace. Jedná se zejména o:

- stanovení výřezu,
- zadání uvažované mřížky (základní, centrální nebo obě),
- zadání kroku simulace,
- stanovení pohledu (obecný 3D, ve směru osy X, Y nebo Z),
- zapnutí nebo vypnutí funkce *preview*,
- obarvení atomů podle energie,
- nastavení průhlednosti atomů podle energie,
- odfiltrování atomů podle energie, atd.



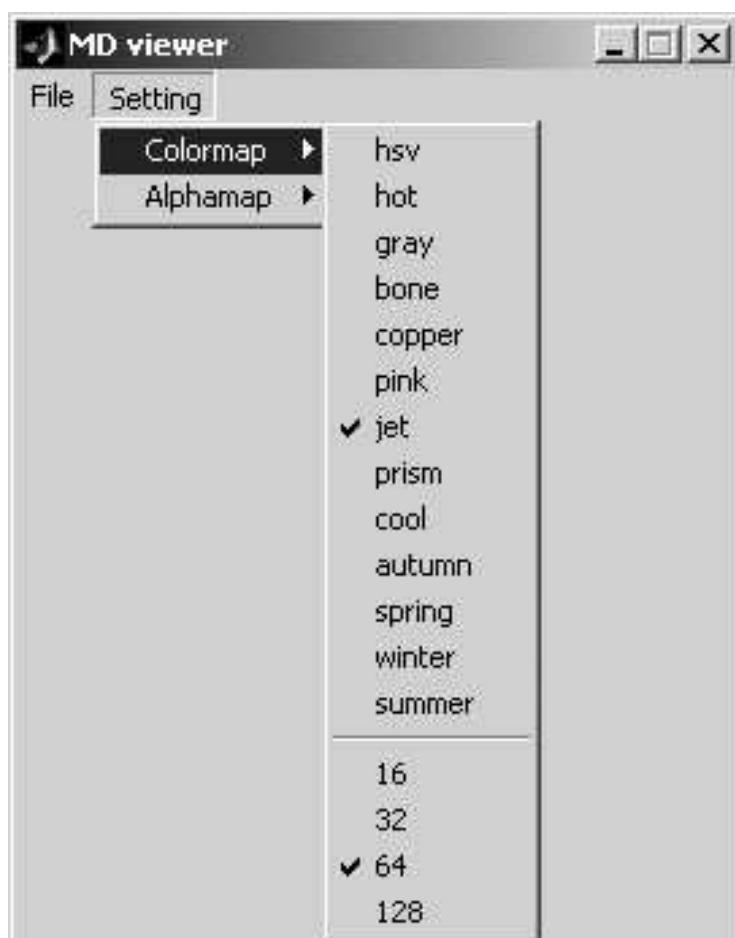
Obr. 3 Uživatelské rozhraní zobrazené po načtení dat.



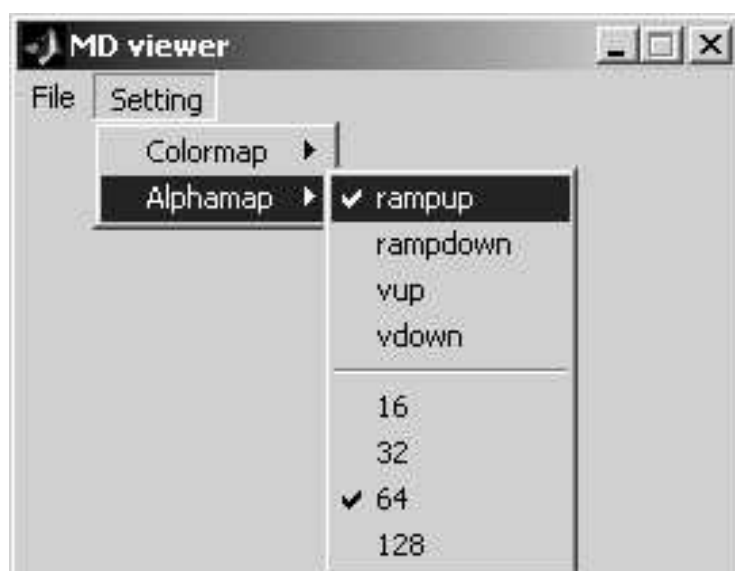
Obr. 4 Grafické okno s vizualizací MD-simulace.

Pokud uživatel v grafickém okně s vizualizací klikne myší na libovolný atom, objeví se poloha vybraného atomu v horní liště okna s uživatelským rozhraním a atom se vykreslí žlutě.

V hlavní nabídce lze ještě volbou **Setting – Colormap** nastavit mapu barev (colormap) a počet barev v této mapě, viz Obr. 5. Volbou **Setting – Alphamap** se nastaví mapa průhledností (alphamap) a počet průhledností v této mapě, viz Obr. 6.



Obr. 5 Nabídka map barev a počtu barev.



Obr. 6 Nabídka map průhledností a počtu průhledností.

## **INSTALACE**

Pro instalaci stačí vytvořit vhodný adresář a do něho zkopírovat tyto čtyři soubory:

```
md_get_header.m
md_get_time_cut.m
md_viewer_gui.m
md_viewer_gui.fig
```

Soubor `md_viewer_gui.cfg` se vytvoří po prvním spuštění.

## **POZNATKY Z PROVOZU**

Program byl psán a odladován v MATLABu 6.1 a 6.5 na notebooku IBM ThinkPad R30 s PIII 1 GHz, 512 MB RAM pod systémem Windows 2000.

Na této sestavě lze v „reálném“ čase, tj. odezva do 5 sekund, vykreslovat cca 1000 atomů. Při větším množství atomů je lepší funkci *preview* vypnout. Na sestavě s PIV 2.4 GHz, 512 MB RAM šlo v „reálném“ čase vykreslovat cca 3000 atomů.

## **PLÁNY DO BUDOUCNA**

Program by bylo vhodné doplnit o následující vlastnosti:

1. Volba veličiny pro obarvování, zprůhledňování a filtraci. S současnou verzí programu se barvy, průhlednost a filtrace odvozují pouze od kinetické energie. Nejvhodnější by bylo začlenění do hlavní nabídky.
2. Vlastní filtrační funkce napsané v MATLABu. Současná verze používá nejjednodušší filtraci, tj. zobrazí se pouze ty atomy, jejichž kinetická energie leží v daném pásmu. Složitější filtrační podmínky by bylo možné aplikovat přes uživatelské funkce, které by se vybíraly v hlavní nabídce.

## Poznatky k paralelním výpočtům na Rusalce

V průběhu prvního roku řešení grantu probíhalo úvodní testování paralelních systémů *OpenMP* a *MPI*, dostupných na stroji *Rusalka*.

V první fázi těchto přípravných prací byl vytvořen relativně jednoduchý program **OMP\_Pi** (viz Příloha 4 na str. 43), který vyčísloval *Ludolfovo číslo* na předem daný počet desetinných míst a, jak název napovídá, byl určen k ověření systému *OpenMP* [3]. Výhodou takového testovacího programu je především jeho přehlednost, poměrně značná výpočetní náročnost a velmi snadná a jednoduchá kontrola dosaženého výsledku. Během těchto testování byly zjištěny následující, velmi málo povzbudivé skutečnosti.

Paralelní kód, generovaný překladačem **F90/95** firmy **Digital** [4], má relativně velkou spotřebu režijního času, což způsobuje, že (při úspěšném zahlcení všech čtyř procesorů stroje *Rusalka*) celkový výkon programu stoupne sotva dvakrát. Dojde-li však k tomu, že alespoň jeden z procesorů stroje *Rusalka* je zatížen zpracováním jiné úlohy, program se započne zpracovávat téměř sekvenčně a jeho výkonnost tudíž může poklesnout až pod výkon původního „neparalelního“ programu.

Co je však mnohem horší, přes veškerou snahu nebylo možno žádným rozumným způsobem dosáhnout správného výsledku. Po mnoha marných pokusech zvítězila hypotéza, že chyba není v programu, ale v překladači. Od tohoto okamžiku byla veškerá snaha soustředěna na vypreparování tohoto problému do transparentní podoby. Tato snaha se setkala s úspěchem a výsledkem byl vznik zcela primitivního programu **OMP\_Matrix\_failure** (viz Příloha 5 na str. 46), který k nulové čtvercové matici 8 x 8 přičítá konstantu 1.

Pokud je program **OMP\_Matrix\_failure** přeložen bez prepínače **-omp**, pracuje bez jakýchkoliv problémů. V opačném případě však je výsledkem

zcela náhodné rozložení nul a jedniček a četnost výskytu těchto selhání závisí na zvoleném typu matice. Nejvyšší chybovost vykazuje matice typu *Integer\*1* a se vzrůstající délkou typu tato chybovost klesá k nule. K těmto selháním dochází zcela systematicky vždy, kdy je *Rusalka* volná a program je skutečně zpracováván plně paralelně. Dojde-li však k tomu, že alespoň jeden z procesorů je zatížen zpracováním jiné úlohy, chyba se zpravidla neprojeví. Rovněž se nikdy nepodařilo vyvolat toto selhání po prohození indexů  $(i, j) \rightarrow (j, i)$ .

Na základě těchto negativních zkušeností a s přihlédnutím ke známému faktu, že fortranské překladače firmy **Digital** patří k nejlepším na světě, bylo upuštěno od dalších pokusů se systémem *OpenMP* na všech platformách a veškerá pozornost byla věnována zkoumání systému *MPI*.

Systém *MPI* [5] má oproti systému *OpenMP* tu obrovskou výhodu, že paralelismu je zde dosahováno nikoliv nějakým *geniálním paralelním kódem*, do kterého nikdo nevidí, ale „prachsprostou“ duplikací sekvenčních programů. Programátor má veškeré dění plně pod kontrolou a tudíž za ně nese plnou zodpovědnost. Chod jednotlivých větví (vláken – *thread*) probíhá zcela nezávisle a transparentně a není tudíž potřeba nějakého zvláštního a složitého testování. Drobnou nevýhodou je, že tato forma paralelismu vyžaduje pokud možno komfortní ale především velice rychlou *komunikaci* mezi jednotlivými programy (*Send, Receive, Wait*), která je zajišťována procedurami z podpůrných knihoven. Rychlost těchto přenosů byla podrobena rozsáhlým zkouškám a u stroje *Rusalka* dosahuje velmi uspokojivých cca **100MB/sec**.

Na základě těchto zkušeností byl k závěrečnému dlouhodobému testu opět použit již osvědčený program pro výpočet *Ludolfova čísla*. Za tím účelem byla vytvořena a spuštěna mutace **MPI\_Pi** (viz Příloha 6 na str. 47) pro čtyři procesory na jeden milion desetinných míst. Mezitím však ale pravděpodobně došlo ke snížení počtu dostupných procesorů, neboť vše nasvědčuje tomu, že *Rusalka* má nyní (prosinec '02) procesory pouze tři, které jsou však navíc téměř nepřetržitě obsazeny, takže všechny čtyři

programy se neustále „přetahovaly“ o jeden až dva procesory a režie daná poměrně značnou *komunikací* neúměrně narůstala. Avšak i v takto ztížených podmínkách program po cca **58** hodinách dospěl ke správnému výsledku, což představuje koeficient zrychlení **1.3**.

Těsně před dokončením této zprávy se *Rusalka* nečekaně uvolnila, takže bylo možno závěrečný dlouhodobý test provést znovu v podmínkách téměř regulérních (bohužel stále pouze na **3** procesorech) s výsledkem:

Tab. 2 Bilance spotřeby strojového času

<b>Spotřeba strojového času</b>	<b>Čas [hod]</b>
1.větev	38:48
2.větev	35:18
3.větev	16:48
4.větev	13:32
Celkem	104:26
Celková doba běhu programu	48:07
Koeficient zrychlení	1.6

Relativně nízký koeficient zrychlení je dán tím, že *komunikace* mezi jednotlivými programy zaujímala v tomto případě cca **26%** spotřebovaného strojového času. Na tomto místě je nutno podotknout, že míra této *komunikace* je (na rozdíl od *OpenMP*) plně pod kontrolou programátora a závisí pouze na způsobu paralelizace daného problému. V tomto konkrétním případě by bylo zcela nepochybně možno navrhnout a zrealizovat paralelizaci této úlohy pro **3** procesory, která by pracovala mnohem efektivněji.

Z výše uvedeného vyplývá, že následující postup řešení bude směřován na paralelizaci MD-simulace pod systémem *MPI* a to jednak na *Rusalkce* a jednak na linuxovém clusteru MetaCentra na ZČU v Plzni (v současnosti bohužel mimo provoz).



## **Závěr**

Zpráva zevrubně popisuje program pro vizualizaci výsledků simulace problémů řešených metodou molekulární dynamiky a věnuje se taktéž prvním poznatkům při paralelních výpočtech, které byly získány na pracovní stanici Rusalka.

Program pro vizualizaci MD-simulací byl vytvořen v prostředí MATLAB a lze jej tudíž spustit pod MATLABem na všech jeho platformách. Na notebooku IBM ThinkPad R30 s PIII 1 GHz, 512 MB RAM pod systémem Windows 2000 šlo v „reálném“ čase, tj. odezva do 5 sekund, vykreslovat cca 1000 atomů. Na sestavě s PIV 2.4 GHz, 512 MB RAM šlo v „reálném“ čase vykreslovat cca 3000 atomů. V příštím období plánujeme doplnit program o volbu veličiny pro obarvování, zprůhledňování a filtraci a začlenit do programu uživatelské filtrační funkce.

Z poznatků získaných při paralelních výpočtech na Rusalce pro nás vyplývá, že následující postup řešení bude směřován na paralelizaci MD-simulace pod systémem *MPI* a to jednak na *Rusalce* a jednak na linuxovém clusteru MetaCentra na ZČU v Plzni.

Zpráva vznikla na základě podpory grantu GA AV ČR č. 10485 „Simulace křehce-tvárného chování mikrotrhlin metodou molekulární dynamiky v 2D a 3D úlohách“.

## Literatura

- [1] [[www.mathworks.com](http://www.mathworks.com)]
- [2] Pelikán, V.; Machová, A.:  
Simulace radiačního poškození v železe.  
Výzkumná zpráva Z1309/01, ÚT AV ČR, Praha 2001
- [3] [[www.openmp.org](http://www.openmp.org)]
- [4] [[www.compaq.com/fortran/dfau.html](http://www.compaq.com/fortran/dfau.html)]
- [5] [[www-unix.mcs.anl.gov/mpi](http://www-unix.mcs.anl.gov/mpi)]

## Přílohy

### PŘÍLOHA 1

```
function header = md_get_header(filename);
%
% header = md_get_header(filename);
%
% Ex.: h = md_get_header('Dirac5.plt');
%

% (c) 2002, Petr Hora

MS = 4; % marker size = 4 byty
TS = 4; % time size = 4 byty

fid = fopen(filename,'rb');

LB_1 = fread(fid,1,'integer*4');

header.M = fread(fid,1,'integer*4');
header.N = fread(fid,1,'integer*4');
header.Ca = fread(fid,1,'integer*4');
header.Nx = fread(fid,1,'integer*4');
header.Ny = fread(fid,1,'integer*4');
header.Nz = fread(fid,1,'integer*4');
header.Cu = fread(fid,7,'integer*4');
header.W = fread(fid,6,'integer*4');
header.C = fread(fid,4,'integer*4');
header.Scales =fread(fid,[2,3],'real*4');

fseek(fid, MS+LB_1+MS, 'bof');
LB_2 = fread(fid,1,'integer*4');

% rychle nacteni casu
header.time = fread(fid,inf,'integer*4',-TS+LB_2+MS+MS);

header.LB_1 = LB_1;
header.LB_2 = LB_2;

fclose(fid)
```

## PŘÍLOHA 2

```
function data = md_get_time_cut(filename,header,cut);
%
% data = md_get_time_cut(filename,header,cut);
%
% Ex.: d = md_get_time_cut('Dirac5.plt', h, 20020);

% (c) Petr Hora, 2002

MS = 4; % marker size = 4 byty
TS = 4; % time size = 4 byty

if isa(filename, 'char'),
    fid = fopen(filename,'rb');
else
    fid = filename;
end

index = find(header.time == cut);
if isempty(index), error('The CUT doesn't exist'); end

fseek(fid, MS+header.LB_1+MS+MS+TS+(index-1)*(-TS+header.LB_2+MS+MS+TS), 'bof');
data = fread(fid,[header.M,header.N],'real*4');

if isa(filename, 'char'),
    fclose(fid);
end

ix = header.W(4)-header.W(1)+1;
ix = ix + (ix-1);
iy = header.W(5)-header.W(2)+1;
iz = header.W(6)-header.W(3)+1;
X = 1e10*reshape(data(1,:),[ix,iy,iz]); % unit = Angstrom
Y = 1e10*reshape(data(2,:),[ix,iy,iz]); % unit = Angstrom
Z = 1e10*reshape(data(3,:),[ix,iy,iz]); % unit = Angstrom
E1 = reshape(log(data(4,:)),[ix,iy,iz]); % log!!!
E2 = reshape(log(data(5,:)),[ix,iy,iz]); % log!!!
%E1 = log(reshape(data(4,:),[ix,iy,iz])); % log!!! Pozor, ve verzi 6.5 nechodi!!!
%E2 = log(reshape(data(5,:),[ix,iy,iz])); % log!!! Pozor, ve verzi 6.5 nechodi!!!

data.N.X = X(1:2:end,:,:); data.C.X = X(2:2:end,:,:); clear X
data.N.Y = Y(1:2:end,:,:); data.C.Y = Y(2:2:end,:,:); clear Y
data.N.Z = Z(1:2:end,:,:); data.C.Z = Z(2:2:end,:,:); clear Z
data.N.E1 = E1(1:2:end,:,:); data.C.E1 = E1(2:2:end,:,:); clear E1
data.N.E2 = E2(1:2:end,:,:); data.C.E2 = E2(2:2:end,:,:); clear E2
```

### PŘÍLOHA 3

```
function varargout = md_viewer_gui(varargin)
% MD_VIEWER_GUI Application M-file for md_viewer_gui.fig
% FIG = MD_VIEWER_GUI launch md_viewer_gui GUI.
% MD_VIEWER_GUI('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 07-Nov-2002 15:27:58

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % fig position
    if exist('md_viewer_gui.cfg', 'file'),
        load('md_viewer_gui.cfg','POSITION_VIEWER','-MAT')
        set(fig,'Position',POSITION_VIEWER)
    else
        set(0,'Units','characters')
        ss = get(0,'ScreenSize');
        set(fig,'Units','characters')
        pos = get(fig,'Position');
        set(fig,'Position',[ss(3)-pos(3)-1, 1, pos(3), pos(4)])
    end

    % Use system color scheme for figure:
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));

    % Generate a structure of handles to pass to callbacks,
    % and store it.
    handles = guihandles(fig);
    handles.ColorN = [0 1 0];
    handles.ColorC = [0 0 1];
    handles.ColorE = [1 0 0];
    handles.ColorS = [1 1 0];
    guidata(fig, handles);

    set(handles.menu_JET, 'Checked','on')
    set(handles.menu_64C, 'Checked','on')
    ud.h_name = [handles.menu_SUMMER, handles.menu_WINTER,...
                handles.menu_SPRING, handles.menu_AUTUMN,
                handles.menu_COOL, handles.menu_PRISM,...
                handles.menu_JET, handles.menu_PINK,...
                handles.menu_COPPER, handles.menu_BONE,...
                handles.menu_GRAY, handles.menu_HOT, ...
                handles.menu_HSV];
    ud.h_number = [handles.menu_128C, handles.menu_64C, ...
                  handles.menu_32C, handles.menu_16C];
    ud.name = 'jet';
    ud.number = 64;
    set(handles.menu_COLORMAP, 'UserData', ud)

    set(handles.menu_RAMPUP, 'Checked', 'on')
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
set(handles.menu_64A, 'Checked', 'on')
ud.h_name = [handles.menu_RAMPUP, handles.menu_RAMPDOWN, ...
            handles.menu_VUP, handles.menu_VDOWN];
ud.h_number = [handles.menu_128A, handles.menu_64A, ...
              handles.menu_32A, handles.menu_16A];
ud.name = 'rampup';
ud.number = 64;
set(handles.menu_ALPHAMAP, 'UserData', ud)

if nargout > 0
    varargout{1} = fig;
end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:});
        else
            feval(varargin{:});
        end
    catch
        disp(lasterr);
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function varargout = slider_MIN_X_Callback(h, eventdata, handles, varargin)
    v = round(get(h, 'Value'));
    set(h, 'Value', v)
    set(handles.text_CUT_MIN_X, 'String', sprintf('%d', v))

    set(handles.patch_MIN_X, 'XData', [v v v v])
    vertices = get(handles.patch_CUT, 'Vertices');
    vertices([1 2 3 4], 1) = v;

    if get(handles.checkbox_GLUE_X, 'Value') == 1,
        set(handles.slider_MAX_X, 'Value', v);
        set(handles.text_CUT_MAX_X, 'String', sprintf('%d', v))
        set(handles.patch_MAX_X, 'XData', [v v v v])
        vertices([5 6 7 8], 1) = v;
    end

    vmax = get(handles.slider_MAX_X, 'Value');
    if v > vmax,
        set(handles.slider_MAX_X, 'Value', v);
        set(handles.text_CUT_MAX_X, 'String', sprintf('%d', v))
        set(handles.patch_MAX_X, 'XData', [v v v v])
        vertices([5 6 7 8], 1) = v;
    end

    set(handles.patch_CUT, 'Vertices', vertices)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_MAX_X_Callback(h, eventdata, handles, varargin)
    v = round(get(h,'Value'));
    set(h,'Value',v)
    set(handles.text_CUT_MAX_X,'String',sprintf('%d',v))

    set(handles.patch_MAX_X,'XData',[v v v])
    vertices = get(handles.patch_CUT,'Vertices');
    vertices([5 6 7 8],1)=v;

    if get(handles.checkbox_GLUE_X,'Value')==1,
        set(handles.slider_MIN_X,'Value',v);
        set(handles.text_CUT_MIN_X,'String',sprintf('%d',v))
        set(handles.patch_MIN_X,'XData',[v v v])
        vertices([1 2 3 4],1)=v;
    end

    vmin = get(handles.slider_MIN_X,'Value');
    if v < vmin,
        set(handles.slider_MIN_X,'Value',v);
        set(handles.text_CUT_MIN_X,'String',sprintf('%d',v))
        set(handles.patch_MIN_X,'XData',[v v v])
        vertices([1 2 3 4],1)=v;
    end

    set(handles.patch_CUT,'Vertices',vertices)

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = slider_MIN_Y_Callback(h, eventdata, handles, varargin)
    v = round(get(h,'Value'));
    set(h,'Value',v)
    set(handles.text_CUT_MIN_Y,'String',sprintf('%d',v))

    set(handles.patch_MIN_Y,'YData',[v v v])
    vertices = get(handles.patch_CUT,'Vertices');
    vertices([1 5 6 2],2)=v;

    if get(handles.checkbox_GLUE_Y,'Value')==1,
        set(handles.slider_MAX_Y,'Value',v);
        set(handles.text_CUT_MAX_Y,'String',sprintf('%d',v))
        set(handles.patch_MAX_Y,'YData',[v v v])
        vertices([4 8 7 3],2)=v;
    end

end
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
vmax = get(handles.slider_MAX_Y,'Value');
if v > vmax,
    set(handles.slider_MAX_Y,'Value',v);
    set(handles.text_CUT_MAX_Y,'String',sprintf('%d',v))
    set(handles.patch_MAX_Y,'YData',[v v v])
    vertices([4 8 7 3],2)=v;
end

set(handles.patch_CUT,'Vertices',vertices)

% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_MAX_Y_Callback(h, eventdata, handles, varargin)
    v = round(get(h,'Value'));
    set(h,'Value',v)
    set(handles.text_CUT_MAX_Y,'String',sprintf('%d',v))

    set(handles.patch_MAX_Y,'YData',[v v v])
    vertices = get(handles.patch_CUT,'Vertices');
    vertices([4 8 7 3],2)=v;

    if get(handles.checkbox_GLUE_Y,'Value')==1,
        set(handles.slider_MIN_Y,'Value',v);
        set(handles.text_CUT_MIN_Y,'String',sprintf('%d',v))
        set(handles.patch_MIN_Y,'YData',[v v v])
        vertices([1 5 6 2],2)=v;
    end

    vmin = get(handles.slider_MIN_Y,'Value');
    if v < vmin,
        set(handles.slider_MIN_Y,'Value',v);
        set(handles.text_CUT_MIN_Y,'String',sprintf('%d',v))
        set(handles.patch_MIN_Y,'YData',[v v v])
        vertices([1 5 6 2],2)=v;
    end

    set(handles.patch_CUT,'Vertices',vertices)

% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_MIN_Z_Callback(h, eventdata, handles, varargin)
    v = round(get(h,'Value'));
    set(h,'Value',v)
    set(handles.text_CUT_MIN_Z,'String',sprintf('%d',v))
```



## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
set(handles.patch_MIN_Z,'ZData',[v v v v])
vertices = get(handles.patch_CUT,'Vertices');
vertices([1 5 8 4],3)=v;

if get(handles.checkbox_GLUE_Z,'Value')==1,
    set(handles.slider_MAX_Z,'Value',v);
    set(handles.text_CUT_MAX_Z,'String',sprintf('%d',v))
    set(handles.patch_MAX_Z,'ZData',[v v v v])
    vertices([2 6 7 3],3)=v;
end

vmax = get(handles.slider_MAX_Z,'Value');
if v > vmax,
    set(handles.slider_MAX_Z,'Value',v);
    set(handles.text_CUT_MAX_Z,'String',sprintf('%d',v))
    set(handles.patch_MAX_Z,'ZData',[v v v v])
    vertices([2 6 7 3],3)=v;
end

set(handles.patch_CUT,'Vertices',vertices)

% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_MAX_Z_Callback(h, eventdata, handles, varargin)
    v = round(get(h,'Value'));
    set(h,'Value',v)
    set(handles.text_CUT_MAX_Z,'String',sprintf('%d',v))

    set(handles.patch_MAX_Z,'ZData',[v v v v])
    vertices = get(handles.patch_CUT,'Vertices');
    vertices([2 6 7 3],3)=v;

    if get(handles.checkbox_GLUE_Z,'Value')==1,
        set(handles.slider_MIN_Z,'Value',v);
        set(handles.text_CUT_MIN_Z,'String',sprintf('%d',v))
        set(handles.patch_MIN_Z,'ZData',[v v v v])
        vertices([1 5 8 4],3)=v;
    end

    vmin = get(handles.slider_MIN_Z,'Value');
    if v < vmin,
        set(handles.slider_MIN_Z,'Value',v);
        set(handles.text_CUT_MIN_Z,'String',sprintf('%d',v))
        set(handles.patch_MIN_Z,'ZData',[v v v v])
        vertices([1 5 8 4],3)=v;
    end

    set(handles.patch_CUT,'Vertices',vertices)

% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_TIME_Callback(h, eventdata, handles, varargin)
    v = get(h,'Value');
    [m,i]=min(abs(handles.HEADER.time-v));
    v=handles.HEADER.time(i);
    set(h,'Value',v)
    set(handles.text_TIME,'String',sprintf('%d',v))

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = checkbox_GLUE_X_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        v = get(handles.slider_MIN_X,'Value');
        set(handles.slider_MAX_X,'Value',v)
        set(handles.text_CUT_MAX_X,'String',sprintf('%d',v))
        set(handles.patch_MAX_X,'XData',v*[1 1 1 1])

        vertices = get(handles.patch_CUT,'Vertices');
        vertices([5 6 7 8],1)=v;
        set(handles.patch_CUT,'Vertices',vertices)
    end

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = checkbox_GLUE_Y_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        v = get(handles.slider_MIN_Y,'Value');
        set(handles.slider_MAX_Y,'Value',v)
        set(handles.text_CUT_MAX_Y,'String',sprintf('%d',v))
        set(handles.patch_MAX_Y,'YData',v*[1 1 1 1])

        vertices = get(handles.patch_CUT,'Vertices');
        vertices([4 2 7 3],2)=v;
        set(handles.patch_CUT,'Vertices',vertices)
    end

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% -----
function varargout = checkbox_GLUE_Z_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        v = get(handles.slider_MIN_Z,'Value');
        set(handles.slider_MAX_Z,'Value',v)
        set(handles.text_CUT_MAX_Z,'String',sprintf('%d',v))
        set(handles.patch_MAX_Z,'ZData',v*[1 1 1])

        vertices = get(handles.patch_CUT,'Vertices');
        vertices([2 6 7 3],3)=v;
        set(handles.patch_CUT,'Vertices',vertices)
    end

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = radiobutton_N_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        off = [handles.radiobutton_C handles.radiobutton_NC];
        set(off,'Value',0)

        % plot
        if get(handles.checkbox_PREVIEW,'Value')==1,
            md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
        end
    end
    else
        set(h,'Value',1)
    end

% -----
function varargout = radiobutton_C_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        off = [handles.radiobutton_N handles.radiobutton_NC];
        set(off,'Value',0)

        % plot
        if get(handles.checkbox_PREVIEW,'Value')==1,
            md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
        end
    end
    else
        set(h,'Value',1)
    end

% -----
function varargout = radiobutton_NC_Callback(h, eventdata, handles, varargin)
    if get(h,'Value')==1,
        off = [handles.radiobutton_N handles.radiobutton_C];
        set(off,'Value',0)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end
else
    set(h,'Value',1)
end

% -----
function varargout = radiobutton_3D_Callback(h, eventdata, handles, varargin)
if get(h,'Value')==1,
    off = [handles.radiobutton_X handles.radiobutton_Y handles.radiobutton_Z];
    set(off,'Value',0)

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end
else
    set(h,'Value',1)
end

% -----
function varargout = radiobutton_X_Callback(h, eventdata, handles, varargin)
if get(h,'Value')==1,
    off = [handles.radiobutton_3D handles.radiobutton_Y handles.radiobutton_Z];
    set(off,'Value',0)

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end
else
    set(h,'Value',1)
end

% -----
function varargout = radiobutton_Y_Callback(h, eventdata, handles, varargin)
if get(h,'Value')==1,
    off = [handles.radiobutton_X handles.radiobutton_3D handles.radiobutton_Z];
    set(off,'Value',0)

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end
else
    set(h,'Value',1)
end
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% -----  
function varargout = radiobutton_Z_Callback(h, eventdata, handles, varargin)  
    if get(h,'Value')==1,  
        off = [handles.radiobutton_X handles.radiobutton_Y handles.radiobutton_3D];  
        set(off,'Value',0)  
  
        % plot  
        if get(handles.checkbox_PREVIEW,'Value')==1,  
            md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))  
        end  
    else  
        set(h,'Value',1)  
    end  
end  
  
% -----  
function varargout = checkbox_PREVIEW_Callback(h, eventdata, handles, varargin)  
    % plot  
    if get(h,'Value')==1,  
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))  
    end  
end  
  
% -----  
function varargout = pushbutton_PLOT_Callback(h, eventdata, handles, varargin)  
    % fprintf(1,'PLOT, called from: %s\n',get(h,'Tag'))  
  
    set([handles.figure_GUI handles.figure_DRAW], 'Pointer', 'watch')  
  
    % nove otevreny soubor, zmena casu, pushbutton_PLOT  
    if strcmp(get(h,'Tag'),'menu_OPEN')|...  
        strcmp(get(h,'Tag'),'slider_TIME')|...  
        strcmp(get(h,'Tag'),'pushbutton_PLOT'),  
  
        d = md_get_time_cut(handles.LOAD_NAME,...  
                            handles.HEADER,...  
                            get(handles.slider_TIME,'Value'));  
  
        handles.DATA = d;  
  
        if strcmp(get(h,'Tag'),'menu_OPEN'),  
            handles.axes_DRAW = axes;  
            set(handles.axes_DRAW, 'DefaultSurfaceVertexNormals', handles.NORMALS,...  
                'DefaultSurfaceEdgeColor','none',...  
                'DefaultSurfaceFaceLighting','phong',...  
                'DefaultSurfaceButtonDownFcn','md_viewer_gui(''get_coordinates_Callback'',gcbo)')  
        else  
            axes(handles.axes_DRAW)  
            cla  
        end  
  
        i1=get(handles.slider_MIN_X,'Value')-handles.HEADER.W(1)+1;  
        i2=get(handles.slider_MAX_X,'Value')-handles.HEADER.W(1)+1;
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
j1=get(handles.slider_MIN_Y,'Value')-handles.HEADER.W(2)+1;
j2=get(handles.slider_MAX_Y,'Value')-handles.HEADER.W(2)+1;
k1=get(handles.slider_MIN_Z,'Value')-handles.HEADER.W(3)+1;
k2=get(handles.slider_MAX_Z,'Value')-handles.HEADER.W(3)+1;

for i=1:i2,
    for j=j1:j2,
        for k=k1:k2
            if get(handles.checkbox_FILTER, 'Value')==0 |...
                (get(handles.slider_MIN_F, 'Value')<=d.N.E1(i,j,k) &...
                get(handles.slider_MAX_F, 'Value')>=d.N.E1(i,j,k)),
                if get(handles.radiobutton_N, 'Value') |...
                    get(handles.radiobutton_NC, 'Value'),
                    surface('XData',handles.BALL_X+d.N.X(i,j,k),...
                        'YData',handles.BALL_Y+d.N.Y(i,j,k),...
                        'ZData',handles.BALL_Z+d.N.Z(i,j,k),...
                        'CData', handles.ZEROS+d.N.E1(i,j,k),...
                        'UserData', [i+handles.HEADER.W(1)-1,...
                            j+handles.HEADER.W(2)-1,...
                            k+handles.HEADER.W(3)-1, 78],...
                        'Tag','N');
                end
                if get(handles.radiobutton_C, 'Value') |...
                    get(handles.radiobutton_NC, 'Value'),
                    surface('XData',handles.BALL_X+d.C.X(i,j,k),...
                        'YData',handles.BALL_Y+d.C.Y(i,j,k),...
                        'ZData',handles.BALL_Z+d.C.Z(i,j,k),...
                        'CData', handles.ZEROS+d.C.E1(i,j,k),...
                        'UserData', [i+handles.HEADER.W(1)-1,...
                            j+handles.HEADER.W(2)-1,...
                            k+handles.HEADER.W(3)-1, 67],...
                        'Tag','C');
                end
            end
        end
    end
end
axis tight
axis equal
xlabel('x'), ylabel('y'), zlabel('z')

% podle radiobuttonu nastaví view
if get(handles.radiobutton_3D, 'Value'), view(3); end
if get(handles.radiobutton_X, 'Value'), view(-90,0); end
if get(handles.radiobutton_Y, 'Value'), view(0,0); end
if get(handles.radiobutton_Z, 'Value'), view(0,90); end

c = caxis;
set(handles.text_MIN_C,'String',sprintf('%7.2e',c(1)))
set(handles.text_MAX_C,'String',sprintf('%7.2e',c(2)))
set(handles.slider_MIN_C,'Min',c(1),'Max',c(2),'Value',c(1))
set(handles.slider_MAX_C,'Min',c(1),'Max',c(2),'Value',c(2))

set(handles.text_MIN_A,'String',sprintf('%7.2e',c(1)))
set(handles.text_MAX_A,'String',sprintf('%7.2e',c(2)))
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
set(handles.slider_MIN_A,'Min',c(1),'Max',c(2),'Value',c(1))
set(handles.slider_MAX_A,'Min',c(1),'Max',c(2),'Value',c(2))

set(handles.text_MIN_F,'String',sprintf('%7.2e',c(1)))
set(handles.text_MAX_F,'String',sprintf('%7.2e',c(2)))
set(handles.slider_MIN_F,'Min',c(1),'Max',c(2),'Value',c(1))
set(handles.slider_MAX_F,'Min',c(1),'Max',c(2),'Value',c(2))

end

% pri jine zmene, nic nenacita,
% pouzou vykresluje z matice, kterou uz ma nactenou

% MIN_X, MAX_X, MIN_Y, MAX_Y, MIN_Z, MAX_Z, LUE_X, GLUE_Y, GLUE_Z
if strcmp(get(h,'Tag'),'slider_MIN_X') |...
    strcmp(get(h,'Tag'),'slider_MAX_X') |...
    strcmp(get(h,'Tag'),'slider_MIN_Y') |...
    strcmp(get(h,'Tag'),'slider_MAX_Y') |...
    strcmp(get(h,'Tag'),'slider_MIN_Z') |...
    strcmp(get(h,'Tag'),'slider_MAX_Z') |...
    strcmp(get(h,'Tag'),'slider_MIN_F') |...
    strcmp(get(h,'Tag'),'slider_MAX_F') |...
    strcmp(get(h,'Tag'),'checkbox_GLUE_X') |...
    strcmp(get(h,'Tag'),'checkbox_GLUE_Y') |...
    strcmp(get(h,'Tag'),'checkbox_GLUE_Z') |...
    strcmp(get(h,'Tag'),'radiobutton_N') |...
    strcmp(get(h,'Tag'),'radiobutton_C') |...
    strcmp(get(h,'Tag'),'radiobutton_NC') |...
    strcmp(get(h,'Tag'),'checkbox_FILTER') |...
    strcmp(get(h,'Tag'),'checkbox_PREVIEW'),

    % vykreslit jina data
    axes(handles.axes_DRAW)
    cla

    i1=get(handles.slider_MIN_X,'Value')-handles.HEADER.W(1)+1;
    i2=get(handles.slider_MAX_X,'Value')-handles.HEADER.W(1)+1;
    j1=get(handles.slider_MIN_Y,'Value')-handles.HEADER.W(2)+1;
    j2=get(handles.slider_MAX_Y,'Value')-handles.HEADER.W(2)+1;
    k1=get(handles.slider_MIN_Z,'Value')-handles.HEADER.W(3)+1;
    k2=get(handles.slider_MAX_Z,'Value')-handles.HEADER.W(3)+1;

    for i=i1:i2,
        for j=j1:j2,
            for k=k1:k2
                if get(handles.checkbox_FILTER,'Value')==0 |...
                    (get(handles.slider_MIN_F,'Value')<=handles.DATA.N.E1(i,j,k) &...
                     get(handles.slider_MAX_F,'Value')>=handles.DATA.N.E1(i,j,k)),
                    if get(handles.radiobutton_N,'Value') |...
                        get(handles.radiobutton_NC,'Value'),
                        surface('XData',handles.BALL_X+handles.DATA.N.X(i,j,k),...
                                'YData',handles.BALL_Y+handles.DATA.N.Y(i,j,k),...
                                'ZData',handles.BALL_Z+handles.DATA.N.Z(i,j,k),...
                                'CData', handles.ZEROS+handles.DATA.N.E1(i,j,k),...
                                'UserData', [i+handles.HEADER.W(1)-1,...
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```

                                j+handles.HEADER.W(2)-1,...
                                k+handles.HEADER.W(3)-1, 78],...
                                'Tag','N');
                                end
                                if get(handles.radiobutton_C, 'Value') |...
                                    get(handles.radiobutton_NC, 'Value'),
                                    surface('XData',handles.BALL_X+handles.DATA.C.X(i,j,k),...
                                        'YData',handles.BALL_Y+handles.DATA.C.Y(i,j,k),...
                                        'ZData',handles.BALL_Z+handles.DATA.C.Z(i,j,k),...
                                        'CData', handles.ZEROS+handles.DATA.C.E1(i,j,k),...
                                        'UserData', [i+handles.HEADER.W(1)-1,...
                                                    j+handles.HEADER.W(2)-1,...
                                                    k+handles.HEADER.W(3)-1, 67],...
                                        'Tag','C');
                                end
                                end
                                end
                                end
                                axis tight
end

% 3D
if strcmp(get(h,'Tag'),'radiobutton_3D'),
    axes(handles.axes_DRAW)
    view(3)
end

% X
if strcmp(get(h,'Tag'),'radiobutton_X'),
    axes(handles.axes_DRAW)
    view(-90,0)
end

% Y
if strcmp(get(h,'Tag'),'radiobutton_Y'),
    axes(handles.axes_DRAW)
    view(0,0)
end

% Z
if strcmp(get(h,'Tag'),'radiobutton_Z'),
    axes(handles.axes_DRAW)
    view(0,90)
end

% COLOR
if strcmp(get(h,'Tag'),'checkbox_COLOR'),
    %
end

% ALPHA
if strcmp(get(h,'Tag'),'checkbox_ALPHA'),
```



## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
%
end

% ColorLim
if strcmp(get(h,'Tag'),'slider_MIN_C') |...
    strcmp(get(h,'Tag'),'slider_MAX_C'),

    caxis(handles.axes_DRAW, [get(handles.slider_MIN_C, 'Value'),...
                             get(handles.slider_MAX_C, 'Value')])
end

% AlphaLim
if strcmp(get(h,'Tag'),'slider_MIN_A') |...
    strcmp(get(h,'Tag'),'slider_MAX_A'),

    axes(handles.axes_DRAW)
    alim([get(handles.slider_MIN_A, 'Value'),...
         get(handles.slider_MAX_A, 'Value')])
end

% proved vzdy
axes(handles.axes_DRAW)
if isempty(findobj(gca,'Type','light')),
    camlight
end

if get(handles.checkbox_COLOR, 'Value')==1,
    set(findobj(gca,'Type','surface'),...
        'FaceColor', 'flat')
else
    set(findobj(gca,'Type','surface','Tag','N'),...
        'FaceColor', handles.ColorN)
    set(findobj(gca,'Type','surface','Tag','C'),...
        'FaceColor', handles.ColorC)
end

if get(handles.checkbox_ALPHA, 'Value')==1,
    alpha('color')
    alpha('flat')
else
    alpha('opaque')
end

% excited atom
if handles.HEADER.C(4) == 0, pom = 'N'; else pom = 'C'; end
set(findobj(gca,'Type','surface', 'UserData', [handles.HEADER.C(1),...
                                                handles.HEADER.C(2),...
                                                handles.HEADER.C(3),...
                                                abs(pom)]),...
    'FaceColor', handles.ColorE)

set([handles.figure_GUI handles.figure_DRAW], 'Pointer', 'arrow')

guidata(h,handles)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% -----  
function varargout = get_coordinates_Callback(h)  
    % Pozor, h je handle surface  
    fGUI = findobj('Type', 'figure', 'Tag', 'figure_GUI');  
    set(fGUI, 'Name', ['MD viewer - ' sprintf('%d, %d, %d, (%c)', get(h, 'UserData'))])  
  
    handles = guidata(fGUI);  
    if isfield(handles, 'sel_ball_h') & ishandle(handles.sel_ball_h),  
        set(handles.sel_ball_h, 'FaceColor', handles.sel_ball_c)  
    end  
    handles.sel_ball_h = h;  
    handles.sel_ball_c = get(h, 'FaceColor');  
    set(h, 'FaceColor', handles.ColorS)  
  
    guidata(fGUI,handles)  
  
% -----  
function varargout = menu_FILE_Callback(h, eventdata, handles, varargin)  
    % disp('FILE')  
  
% -----  
function varargout = menu_OPEN_Callback(h, eventdata, handles, varargin)  
    [fname,fpath]=uigetfile('*.plt','Load MD data file');  
    if fname == 0, return, end  
  
    % enable on  
    set(handles.axes_VIEW, 'Visible', 'on')  
    set(findobj('Enable', 'off'),'Enable','on')  
    set([handles.slider_MIN_C, handles.slider_MAX_C,...  
        handles.text_MIN_C, handles.text_MAX_C],'Enable','off')  
    set([handles.slider_MIN_A, handles.slider_MAX_A,...  
        handles.text_MIN_A, handles.text_MAX_A],'Enable','off')  
    set([handles.slider_MIN_F, handles.slider_MAX_F,...  
        handles.text_MIN_F, handles.text_MAX_F],'Enable','off')  
  
    LOAD_NAME = fullfile(fpath,fname);  
  
    HEADER = md_get_header(LOAD_NAME);  
  
    handles.HEADER=HEADER;  
    handles.LOAD_NAME=LOAD_NAME;  
  
    minx = HEADER.W(1);  
    miny = HEADER.W(2);  
    minz = HEADER.W(3);  
    maxx = HEADER.W(4);  
    maxy = HEADER.W(5);  
    maxz = HEADER.W(6);  
  
    minxc = HEADER.C(1)-1;  
    minyc = HEADER.C(2)-1;  
    minzc = HEADER.C(3)-1;
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
maxxc = HEADER.C(1)+1;
maxyc = HEADER.C(2)+1;
maxzc = HEADER.C(3)+1;

set(handles.text_MIN_X, 'string', sprintf('%d', minx))
set(handles.text_MIN_Y, 'string', sprintf('%d', miny))
set(handles.text_MIN_Z, 'string', sprintf('%d', minz))
set(handles.text_MAX_X, 'string', sprintf('%d', maxx))
set(handles.text_MAX_Y, 'string', sprintf('%d', maxy))
set(handles.text_MAX_Z, 'string', sprintf('%d', maxz))

set(handles.text_CUT_MIN_X, 'String', sprintf('%d', minxc))
set(handles.text_CUT_MIN_Y, 'String', sprintf('%d', minyc))
set(handles.text_CUT_MIN_Z, 'String', sprintf('%d', minzc))
set(handles.text_CUT_MAX_X, 'String', sprintf('%d', maxx))
set(handles.text_CUT_MAX_Y, 'String', sprintf('%d', maxyc))
set(handles.text_CUT_MAX_Z, 'String', sprintf('%d', maxzc))

set(handles.slider_MIN_X, 'Min', minx, 'Max', maxx, ...
    'Value', minxc, 'SliderStep', 1/(maxx-minx)*[1 1])
set(handles.slider_MIN_Y, 'Min', miny, 'Max', maxy, ...
    'Value', minyc, 'SliderStep', 1/(maxy-miny)*[1 1])
set(handles.slider_MIN_Z, 'Min', minz, 'Max', maxz, ...
    'Value', minzc, 'SliderStep', 1/(maxz-minz)*[1 1])
set(handles.slider_MAX_X, 'Min', minx, 'Max', maxx, ...
    'Value', maxx, 'SliderStep', 1/(maxx-minx)*[1 1])
set(handles.slider_MAX_Y, 'Min', miny, 'Max', maxy, ...
    'Value', maxyc, 'SliderStep', 1/(maxy-miny)*[1 1])
set(handles.slider_MAX_Z, 'Min', minz, 'Max', maxz, ...
    'Value', maxzc, 'SliderStep', 1/(maxz-minz)*[1 1])

set(handles.text_MIN_TIME, 'string', sprintf('%d', HEADER.time(1)))
set(handles.text_MAX_TIME, 'string', sprintf('%d', HEADER.time(end)))

set(handles.text_TIME, 'String', sprintf('%d', HEADER.time(1)))

set(handles.slider_TIME, ...
    'Min', HEADER.time(1), 'Max', HEADER.time(end), 'Value', HEADER.time(1), ...
    'SliderStep', (HEADER.time(2)-HEADER.time(1))/(HEADER.time(end)-HEADER.time(1))*[1 1])

xlabel('x')
ylabel('y')
zlabel('z')
axis([minx maxx miny maxy minz maxz])

v = [minxc minyc minzc; minxc minyc maxzc; minxc maxyc maxzc; minxc maxyc minzc; ...
    maxx minyc minzc; maxx minyc maxzc; maxx maxyc maxzc; maxx maxyc minzc];
f = [1 2 3 4; 5 6 7 8; 1 5 6 2; 4 8 7 3; 1 5 8 4; 2 6 7 3];
c = [0 1 0; 1 0 0; 0 1 0; 1 0 0; 0 1 0; 1 0 0];

handles.patch_CUT = patch('Vertices',v,'Faces',f,...
    'FaceVertexCData',c,...
    'FaceColor','flat','FaceAlpha',0.8);

handles.patch_MIN_X = patch([minxc minxc minxc minxc],...
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
        [miny maxy maxy miny],...
        [minz minz maxz maxz],...
        'g','EdgeColor','none','FaceAlpha',0.25);
handles.patch_MAX_X = patch([maxxc maxxc maxxc maxxc],...
        [miny maxy maxy miny],...
        [minz minz maxz maxz],...
        'r','EdgeColor','none','FaceAlpha',0.25);
handles.patch_MIN_Y = patch([minx maxx maxx minx],...
        [minyc minyc minyc minyc],...
        [minz minz maxz maxz],...
        'g','EdgeColor','none','FaceAlpha',0.25);
handles.patch_MAX_Y = patch([minx maxx maxx minx],...
        [maxyc maxyc maxyc maxyc],...
        [minz minz maxz maxz],...
        'r','EdgeColor','none','FaceAlpha',0.25);
handles.patch_MIN_Z = patch([minx maxx maxx minx],...
        [miny miny maxy maxy],...
        [minzc minzc minzc minzc],...
        'g','EdgeColor','none','FaceAlpha',0.25);
handles.patch_MAX_Z = patch([minx maxx maxx minx],...
        [miny miny maxy maxy],...
        [maxzc maxzc maxzc maxzc],...
        'r','EdgeColor','none','FaceAlpha',0.25);

view(3)
% axis tight

% drawing figure
if isfield(handles,'figure_DRAW') & ishandle(handles.figure_DRAW),
    figure(handles.figure_DRAW)
    clf
else
    if exist('md_viewer_gui.cfg', 'file'),
        load('md_viewer_gui.cfg','POSITION_DRAW','-MAT')
        handles.figure_DRAW = figure('Units','characters',...
            'Position',POSITION_DRAW,...
            'Name',['Drawing figure: ' LOAD_NAME],...
            'NumberTitle','off');
    else
        pos = get(handles.figure_GUI,'Position');
        handles.figure_DRAW = figure('Units','characters',...
            'Position',[1, pos(2), pos(1)-2, pos(4)],...
            'Name',['Drawing figure: ' LOAD_NAME],...
            'NumberTitle','off');
    end
end

[handles.BALL_X,handles.BALL_Y,handles.BALL_Z]=sphere(20); % 20
handles.BALL_X=handles.BALL_X*0.5;
handles.BALL_Y=handles.BALL_Y*0.5;
handles.BALL_Z=handles.BALL_Z*0.5;
handles.ZEROS=zeros(size(handles.BALL_Z));
handles.NORMALS=cat(3,handles.BALL_X, handles.BALL_Y, handles.BALL_Z);

guidata(h,handles)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% plot
md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))

% -----
function varargout = checkbox_COLOR_Callback(h, eventdata, handles, varargin)
    hs=findobj(handles.axes_DRAW,'Type','surface');
    if get(h, 'Value')==1,
        set([handles.slider_MIN_C, handles.slider_MAX_C,...
            handles.text_MIN_C,  handles.text_MAX_C], 'Enable', 'on')
    else
        set([handles.slider_MIN_C, handles.slider_MAX_C,...
            handles.text_MIN_C,  handles.text_MAX_C], 'Enable', 'off')
    end

    % plot
    if get(handles.checkbox_PREVIEW, 'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = checkbox_ALPHA_Callback(h, eventdata, handles, varargin)
    hs=findobj(handles.axes_DRAW,'Type','surface');
    if get(h, 'Value')==1,
        set([handles.slider_MIN_A, handles.slider_MAX_A,...
            handles.text_MIN_A,  handles.text_MAX_A], 'Enable', 'on')
    else
        set([handles.slider_MIN_A, handles.slider_MAX_A,...
            handles.text_MIN_A,  handles.text_MAX_A], 'Enable', 'off')
    end

    % plot
    if get(handles.checkbox_PREVIEW, 'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = checkbox_FILTER_Callback(h, eventdata, handles, varargin)
    if get(h, 'Value')==1,
        set([handles.slider_MIN_F, handles.slider_MAX_F,...
            handles.text_MIN_F,  handles.text_MAX_F], 'Enable', 'on')
    else
        set([handles.slider_MIN_F, handles.slider_MAX_F,...
            handles.text_MIN_F,  handles.text_MAX_F], 'Enable', 'off')
    end

    % plot
    if get(handles.checkbox_PREVIEW, 'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% -----  
function varargout = slider_MIN_C_Callback(h, eventdata, handles, varargin)  
    v = get(h, 'Value');  
    set(handles.text_MIN_C, 'String', sprintf('%7.2e', v))  
  
    vmax = get(handles.slider_MAX_C, 'Value');  
    if v >= vmax,  
        vmax=v;  
        set(handles.slider_MAX_C, 'Value', vmax);  
        set(handles.text_MAX_C, 'String', sprintf('%7.2e', vmax))  
    end  
  
    % plot  
    if get(handles.checkbox_PREVIEW, 'Value')==1,  
        md_viewer_gui('pushbutton_PLOT_Callback', gcbo, [], guidata(gcbo))  
    end  
  
% -----  
function varargout = slider_MAX_C_Callback(h, eventdata, handles, varargin)  
    v = get(h, 'Value');  
    set(handles.text_MAX_C, 'String', sprintf('%7.2e', v))  
  
    vmin = get(handles.slider_MIN_C, 'Value');  
    if v <= vmin,  
        vmin=v;  
        set(handles.slider_MIN_C, 'Value', vmin);  
        set(handles.text_MIN_C, 'String', sprintf('%7.2e', vmin))  
    end  
  
    % plot  
    if get(handles.checkbox_PREVIEW, 'Value')==1,  
        md_viewer_gui('pushbutton_PLOT_Callback', gcbo, [], guidata(gcbo))  
    end  
  
% -----  
function varargout = slider_MIN_A_Callback(h, eventdata, handles, varargin)  
    v = get(h, 'Value');  
    set(handles.text_MIN_A, 'String', sprintf('%7.2e', v))  
  
    vmax = get(handles.slider_MAX_A, 'Value');  
    if v >= vmax,  
        vmax=v;  
        set(handles.slider_MAX_A, 'Value', vmax);  
        set(handles.text_MAX_A, 'String', sprintf('%7.2e', vmax))  
    end  
  
    % plot  
    if get(handles.checkbox_PREVIEW, 'Value')==1,  
        md_viewer_gui('pushbutton_PLOT_Callback', gcbo, [], guidata(gcbo))  
    end  
  
% -----  
function varargout = slider_MAX_A_Callback(h, eventdata, handles, varargin)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
v = get(h,'Value');
set(handles.text_MAX_A,'String',sprintf('%7.2e',v))

vmin = get(handles.slider_MIN_A,'Value');
if v <= vmin,
    vmin=v;
    set(handles.slider_MIN_A,'Value',vmin);
    set(handles.text_MIN_A,'String',sprintf('%7.2e',vmin))
end

% plot
if get(handles.checkbox_PREVIEW,'Value')==1,
    md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
end

% -----
function varargout = slider_MIN_F_Callback(h, eventdata, handles, varargin)
    v = get(h,'Value');
    set(handles.text_MIN_F,'String',sprintf('%7.2e',v))

    vmax = get(handles.slider_MAX_F,'Value');
    if v >= vmax,
        vmax=v;
        set(handles.slider_MAX_F,'Value',vmax);
        set(handles.text_MAX_F,'String',sprintf('%7.2e',vmax))
    end

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = slider_MAX_F_Callback(h, eventdata, handles, varargin)
    v = get(h,'Value');
    set(handles.text_MAX_F,'String',sprintf('%7.2e',v))

    vmin = get(handles.slider_MIN_F,'Value');
    if v <= vmin,
        vmin=v;
        set(handles.slider_MIN_F,'Value',vmin);
        set(handles.text_MIN_F,'String',sprintf('%7.2e',vmin))
    end

    % plot
    if get(handles.checkbox_PREVIEW,'Value')==1,
        md_viewer_gui('pushbutton_PLOT_Callback',gcbo,[],guidata(gcbo))
    end

% -----
function varargout = menu_SETTING_Callback(h, eventdata, handles, varargin)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
% -----  
function varargout = menu_COLORMAP_Callback(h, eventdata, handles, varargin)  
  
% -----  
function varargout = menu_ALPHAMAP_Callback(h, eventdata, handles, varargin)  
  
% -----  
function varargout = menu_COLORMAP_NAME_Callback(h, eventdata, handles, varargin)  
    ud = get(handles.menu_COLORMAP, 'UserData');  
    set(ud.h_name, 'Checked', 'off')  
    set(h, 'Checked', 'on')  
    ud.name = get(h, 'Label');  
    set(handles.menu_COLORMAP, 'UserData', ud);  
    axes(handles.axes_DRAW)  
    eval(['colormap(', ud.name, '(', num2str(ud.number), ')'])  
  
% -----  
function varargout = menu_COLORMAP_NUMBER_Callback(h, eventdata, handles, varargin)  
    ud = get(handles.menu_COLORMAP, 'UserData');  
    set(ud.h_number, 'Checked', 'off')  
    set(h, 'Checked', 'on')  
    ud.number = str2num(get(h, 'Label'));  
    set(handles.menu_COLORMAP, 'UserData', ud);  
    axes(handles.axes_DRAW)  
    eval(['colormap(', ud.name, '(', num2str(ud.number), ')'])  
  
% -----  
function varargout = menu_ALPHAMAP_NAME_Callback(h, eventdata, handles, varargin)  
    ud = get(handles.menu_ALPHAMAP, 'UserData');  
    set(ud.h_name, 'Checked', 'off')  
    set(h, 'Checked', 'on')  
    ud.name = get(h, 'Label');  
    set(handles.menu_ALPHAMAP, 'UserData', ud);  
    axes(handles.axes_DRAW)  
    eval(['alphamap(', '''', ud.name, '''', ',', num2str(ud.number), ')'])  
  
% -----  
function varargout = menu_ALPHAMAP_NUMBER_Callback(h, eventdata, handles, varargin)  
    ud = get(handles.menu_ALPHAMAP, 'UserData');  
    set(ud.h_number, 'Checked', 'off')  
    set(h, 'Checked', 'on')  
    ud.number = str2num(get(h, 'Label'));  
    set(handles.menu_ALPHAMAP, 'UserData', ud);  
    axes(handles.axes_DRAW)  
    eval(['alphamap(', '''', ud.name, '''', ',', num2str(ud.number), ')'])  
  
% -----  
function varargout = menu_EXIT_Callback(h, eventdata, handles, varargin)  
    delete(gcf)  
  
% -----
```



## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
function varargout = pushbutton_EXIT_Callback(h, eventdata, handles, varargin)
    delete(gcf)

% -----
function varargout = figure_GUI_CloseRequestFcn(h, eventdata, handles, varargin)
    % disp('closerequest')
    delete(gcf)

% -----
function varargout = figure_GUI_DeleteFcn(h, eventdata, handles, varargin)
    % disp('delete')

    if isfield(handles,'figure_DRAW') & ishandle(handles.figure_DRAW),
        POSITION_DRAW = get(handles.figure_DRAW, 'position');
        save('md_viewer_gui.cfg','POSITION_DRAW','-MAT')
        close(handles.figure_DRAW)
    end

    POSITION_VIEWER = get(gcf, 'position');
    save('md_viewer_gui.cfg','POSITION_VIEWER','-MAT', '-APPEND')
```

## PŘÍLOHA 4

```

Program OMP_Pi ! (C) Pelikan - Listopad 2002
! *****
Implicit None
Character*10 Navez
integer*4 pocet_mist,i,n,nr(5),d,t,s,d1,d2,r1,r2,r3,t1,t2
parameter ( pocet_mist = 32768)
integer*1 reg(0:7,pocet_mist) ! Využívá CACHE => až 3x rychlejší
real*4 secnds,save

data n/-3/nr/5*1/s/0/save/0./d1,d2,r1,r2,r3,t1,t2/1,2,3,4,5,6,7/
reg=0
reg(d1,1)=4*2 ! Pi = (4/2+4/3) - (4/2^3+4/3^3)/3 + (4/2^5+4/3^5)/5 - ...
reg(t1,1)=4*3
reg(r2,1)=30
save=secnds(save)

Write(Navez,"('Pi_'I7.7)")pocet_mist
Open(unit=1,file=Navez//'.stav',action='write')
write(1,('' Pocet mist:'' I10/'\n, nd, nr, s:''6I10,F11.0)')
1 pocet_mist,n,nr,save
D print ('' Pocet mist:'' I10/'\n, nd, nr, s:''6I10,F11.0)',
D 1 pocet_mist,n,nr,save
Close(unit=1)

DoWhile(nr(r3).lt.pocet_mist)

!$omp Parallel Sections Default(None)
!$omp& Shared(nr,reg)
!$omp& FirstPrivate(n,d1,d2,r1,r2,r3,t1,t2)
!$omp& Private(i,d,t)

d=0
t=0
do i=nr(d1),pocet_mist
d=d*10+reg(d1,i) ! reg(d2) := reg(d1)/4
reg(d2,i)=ishft(d,-2)
d=iand(d,3)
t=t*10+reg(t1,i) ! reg(t2) := reg(t1)/9
reg(t2,i)=t/9
t=t-reg(t2,i)*9
end do
dowhile(nr(d2).lt.nr(d1).or.reg(d2,nr(d2)).eq.0)
reg(d2,nr(d2))=0
reg(t2,nr(d2))=0
if(nr(d2).ge.pocet_mist)Exit
nr(d2)=nr(d2)+1
end do

!$omp Section
d=0 ! reg(r1) := 4/2^n + 4/3^n
do i=pocet_mist,nr(r1),-1
reg(r1,i)=d+reg(d1,i)+reg(t1,i)

```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
d=0
if(reg(r1,i).ge.10)then
  reg(r1,i)=reg(r1,i)-10
  d=1
end if
end do
dowhile(nr(r1).lt.pocet_mist.and.reg(r1,nr(r1)).eq.0)
  nr(r1)=nr(r1)+1
end do

!$omp Section
d=0
do i=nr(r2),pocet_mist      ! reg(r2) := reg(r2)/n
  d=d*10+reg(r2,i)
  reg(r2,i)=d/n
  d=d-reg(r2,i)*n
end do

!$omp Section
d=0
i=pocet_mist
if(iand(n,2).ne.0)then      ! pi := pi+reg(r3)
  doWhile(i.ge.nr(r3).or.d.ne.0)
    reg(0,i)=reg(0,i)+reg(r3,i)+d
    d=0
    if(reg(0,i).ge.10)then
      reg(0,i)=reg(0,i)-10
      d=1
    end if
    i=i-1
  end do
else                          ! pi := pi-reg(r3)
  doWhile(i.ge.nr(r3).or.d.ne.0)
    reg(0,i)=reg(0,i)-reg(r3,i)-d
    d=0
    if(reg(0,i).lt.0)then
      reg(0,i)=reg(0,i)+10
      d=1
    end if
    i=i-1
  end do
end if

!$omp End Parallel Sections

i=d2
d2=d1
d1=i
i=t2
t2=t1
t1=i
i=r3
r3=r2
r2=r1
r1=i
n=n+2
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
if(Mod(n,1000).eq.999.or.nr(r3).ge.pocet_mist)then
  Open(unit=1,file=Nazev//'.stav',
1      action='write',access='APPEND')
  i=s
  s=secnds(save)
  write(1,(''n, nd, nr, s:''8I10')n,nr,s,s-i
D      print  (''n, nd, nr, s:''8I10',n,nr,s,s-i
      Close(unit=1)
  end if
End Do

D print (''Tabulkove Pi:''F20.17)',2*asin(1d0)
D print ('' Soucet rady:''i2''.'17i1)',(reg(0,i),i=1,18)

Open(unit=1,file=Nazev)
write(1,(''Delka Eulerovy rady:''I//(100i1))'n,
1      (reg(0,i),i=1,pocet_mist)
End
```

## PŘÍLOHA 5

```
Program OMP_Matrix_failure ! (C) Pelikan - November 2002
! *****
Implicit None
Integer*4 i
Integer*1 Matrix(8,8)
Data Matrix/64*0/

!$omp Parallel Sections Default(None)
!$omp& Shared(Matrix)
!$omp& Private(i)
!$omp Section
  Do i=1,8
    Matrix(1,i)=Matrix(1,i)+1
    Matrix(2,i)=Matrix(2,i)+1
  End Do
!$omp Section
  Do i=1,8
    Matrix(3,i)=Matrix(3,i)+1
    Matrix(4,i)=Matrix(4,i)+1
  End Do
!$omp Section
  Do i=1,8
    Matrix(5,i)=Matrix(5,i)+1
    Matrix(6,i)=Matrix(6,i)+1
  End Do
!$omp Section
  Do i=1,8
    Matrix(7,i)=Matrix(7,i)+1
    Matrix(8,i)=Matrix(8,i)+1
  End Do
!$omp End Parallel Sections

Print "(8(' '8i1))",Matrix
End
```

## PŘÍLOHA 6

```

Program MPI_Pi ! (C) Pelikan - Prosinec 2002
! *****
Use MPI
Implicit None
Character*10 Nazev
integer*4 pocet_mist,pn,pr,i,n,d,t,s
parameter ( pocet_mist = 1000050
1, pn = ( pocet_mist+3 )/4
1, pr = pn*4 )
integer*4 nreg(0:pn,3),w,b1,b2,h1,h2,status(MPI_STATUS_SIZE)
integer*1 reg(-3:pr,3)
real*4 secnds,save
equivalence ( nreg,reg )

Data n/1/w/1/b1/2/b2/3/h2/MPI_REQUEST_NULL/save/0./s/0/

Call MPI_Init(i)
Call MPI_COMM_Size(MPI_COMM_WORLD,d,i)
Call MPI_COMM_Rank(MPI_COMM_WORLD,t,i)

If(d.lt.4)Then
  If(t.eq.0)Print '('Pocet procesoru:''I2''.'/
1 '(' Program musi byt spusten na 4 procesorech. )''',d
Else

Select Case(t)

Case(0) ! reg := reg(horni)/4 + reg(dolni)/9
! %%%
nreg=0
nreg(0,b1)=pocet_mist
reg(nreg(0,b1),b1)=-116 ! 4*2*16 + 4*3

Do
  d=0
  t=0
  do i=nreg(0,b1),1,-1
    s=reg(i,b1)
    d=iand(d,3*16)
    d=d*10+iand(s,15*16)
    t=t*10+iand(s,15)
    s=t/9
    t=t-s*9
    reg(i,b2)=iand(ishft(d,-2),15*16)+s
  end do
  nreg(0,b2)=nreg(0,b1)
  DoWhile(nreg(0,b2).gt.0.and.reg(nreg(0,b2),b2).eq.0)
    nreg(0,b2)=nreg(0,b2)-1
  End Do
  call MPI_Wait(h2,status,i)

```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
If(nreg(0,b1).le.0)Exit
call MPI_IsSend(nreg(0,b2),nreg(0,b2)+4,MPI_BYTE,
1      1,1,MPI_COMM_WORLD,h2,i)
  i=b1
  b1=b2
  b2=i
End Do

Case(1) ! reg := 4/2^n + 4/3^n
! %%%
call MPI_IRECV(nreg(0,w),pocet_mist+4,MPI_BYTE,
1      0,1,MPI_COMM_WORLD,h1,i)

Do
  call MPI_Wait(h1,status,i)
  If(nreg(0,w).le.0)Exit
  call MPI_IRECV(nreg(0,b1),pocet_mist+4,MPI_BYTE,
1      0,1,MPI_COMM_WORLD,h1,i)
  d=0
  do i=1,nreg(0,w)
    reg(i,w)=d+iand(reg(i,w),15)+iand(ishft(reg(i,w),-4),15)
    d=0
    if(reg(i,w).ge.10)then
      reg(i,w)=reg(i,w)-10
      d=1
    end if
  end do
  call MPI_Wait(h2,status,i)
  call MPI_IsSend(nreg(0,w),nreg(0,w)+4,MPI_BYTE,
1      2,2,MPI_COMM_WORLD,h2,i)
  i=b2
  b2=w
  w=b1
  b1=i
End Do

call MPI_Wait(h2,status,i)
call MPI_IsSend(nreg(0,w),nreg(0,w)+4,MPI_BYTE,
1      2,2,MPI_COMM_WORLD,h2,i)
call MPI_Wait(h2,status,i)

Case(2) ! reg := reg/n
! %%%
call MPI_IRECV(nreg(0,w),pocet_mist+4,MPI_BYTE,
1      1,2,MPI_COMM_WORLD,h1,i)

Do
  call MPI_Wait(h1,status,i)
  If(nreg(0,w).le.0)Exit
  call MPI_IRECV(nreg(0,b1),pocet_mist+4,MPI_BYTE,
1      1,2,MPI_COMM_WORLD,h1,i)
  d=0
  do i=nreg(0,w),1,-1
    d=d*10+reg(i,w)
```

## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```

        reg(i,w)=d/n
        d=d-reg(i,w)*n
    end do
    call MPI_Wait(h2,status,i)
    call MPI_IsSend(nreg(0,w),nreg(0,w)+4,MPI_BYTE,
1       3,3,MPI_COMM_WORLD,h2,i)
    n=n+2
    i=b2
    b2=w
        w=b1
        b1=i
End Do

call MPI_Wait(h2,status,i)
call MPI_IsSend(nreg(0,w),nreg(0,w)+4,MPI_BYTE,
1       3,3,MPI_COMM_WORLD,h2,i)
call MPI_Wait(h2,status,i)

Case(3) ! Pi := (4/2+4/3) - (4/2^3+4/3^3)/3 + (4/2^5+4/3^5)/5 - ...
! %%%
t=pocet_mist
nreg=0
nreg(0,b2)=t
Write(Nazev,"('Pi_'I7.7)")t
Open(unit=1,File=Nazev//'.stav',action='write')
write(1,('File: 'A'.stav'))Nazev
D print ('File: 'A'.stav'),Nazev
Close(unit=1)
call MPI_Irecv(nreg(0,b1),pocet_mist+4,MPI_BYTE,
1       2,3,MPI_COMM_WORLD,h1,i)
save=secnds(save)

Do
    call MPI_Wait(h1,status,i)
    if(Mod(n,1000).eq.1.or.nreg(0,b1).le.0)then
        Open(unit=1,file=Nazev//'.stav',
1           action='write',access='APPEND')
        d=s
        s=secnds(save)
        write(1,('n, reg(0), s:'4I10'))n,nreg(0,b1),s,s-d
D print ('n, reg(0), s:'4I10'),n,nreg(0,b1),s,s-d
        Close(unit=1)
        If(nreg(0,b1).le.0)Exit
    end if
    Do i=nreg(0,b1)+1,t
        reg(i,b1)=0
    End Do
    t=nreg(0,b2)
    call MPI_Irecv(nreg(0,b2),pocet_mist+4,MPI_BYTE,
1       2,3,MPI_COMM_WORLD,h1,i)
    d=0
    i=0
    if(iand(n,2).eq.0)then ! Pi := Pi + reg
        doWhile(i.lt.nreg(0,b1).or.d.ne.0)
            i=i+1
            reg(i,w)=reg(i,w)+reg(i,b1)+d

```



## Výzkumná zpráva Ústavu termomechaniky AV ČR č. Z1322/02

```
d=0
if(reg(i,w).ge.10)then
  reg(i,w)=reg(i,w)-10
  d=1
end if
end do
else
  ! Pi := Pi - reg
doWhile(i.lt.nreg(0,b1).or.d.ne.0)
  i=i+1
  reg(i,w)=reg(i,w)-reg(i,b1)-d
  d=0
  if(reg(i,w).lt.0)then
    reg(i,w)=reg(i,w)+10
    d=1
  end if
end do
end if
n=n+2
i=b1
b1=b2
b2=i
End Do

Open(unit=1,file=Nazev)
write(1,(''Delka Eulerovy rady:''I//(100i1)')n
1,      (reg(i,w),i=pocet_mist,1,-1)

Case Default
Continue
End Select
End If
Call MPI_Finalize(i)
End
```