

Poprvé s ParaView

ParaView je volně dostupný software pro vizualizaci dat ve dvou a hlavně třech prostorových dimenzích, včetně animování v čase. Zobrazuje data na rovnoměrných, rektilineárních a křivočarých **strukturovaných sítích** a na polygonálních (2D) a obecných (3D) **nestrukturovaných sítích**. Stojí na vizualizační knihovně **VTK** (Visualization Toolkit), vstupní data tedy mohou být čtena v jejich formátech a také v jiných formátech, pro které jsou k dispozici **readery**. Ovládá se řetězením **filtrů** v interaktivním grafickém prostředí nebo pythonovských skriptech. Dostupné je pro Windows i Linux, kde může s podporou knihovny MPI vizualizovat rozměrná data paralelizovaně.

Sítě

Síť (mesh, grid) je tvořena **vrcholy** (points, vertices) a **buňkami** (cells, elements). Pravidelné síť mohou být popsány implicitně (rozměrem sítě a počtem vrcholů nebo buněk), všechny síť lze popsat explicitně (kartézskými souřadnicemi vrcholů a **topologií**, tj. přiřazením vrcholů k buňkám). Data (skaláry, vektory a obecná pole) mohou být definována jak ve vrcholech, tak v buňkách.

Rovnoměrná síť (uniform rectilinear grid)

- kartézská souřadnicová síť s vrcholy rozmístěnými ekvidistantně (**implicitní topologie**)
- všechny buňky sítě téhož typu, podle počtu dimenzí: 0D vrchol/vertex, 1D čára, 2D obdélník/pixel, 3D kvádr/voxel
- **implicitní souřadnice vrcholů**: předepsán jen počet vrcholů, souřadnice počátku a délka kroku v každé dimenzi, pak `coordinates(1:3,i,j,k) = [x0+i*dx,y0+j*dy,z0+k*dz]`
- „fortranské“ ukládání polí: prvky pole sousedící v paměti mají sousedící první indexy
tj. 1D „flat“ index: `ijk = i + xsize*(j + ysize*k)`
- AMR síť (adaptive mesh refinement): několik (překrývajících se) rovnoměrných sítí s různým krokováním

Rektilineární síť (rectilinear grid)

- kartézská souřadnicová síť s předepsanými (obecně neekvidistantními) polohami vrcholů na souřadnicových osách
- **semi-implicitní souřadnice vrcholů**: předepsán počet vrcholů a souřadnice vrcholů na každé z os (tři 1D pole), pak `coordinates(1:3,i,j,k) = [x(i),y(j),z(k)]`

Křivočará síť (curvilinear structured grid)

- obecná souřadnicová síť s předepsanými polohami vrcholů na křivočarých souřadnicových osách
- **explicitní souřadnice vrcholů**: předepsán počet vrcholů a souřadnice každého vrcholu (tři 3D nebo jedno 4D pole), pak `coordinates(1:3,i,j,k) = [x(i,j,k),y(i,j,k),z(i,j,k)]` nebo `coordinates(1:3,i,j,k) = xyz(1:3,i,j,k)`
- buňky sítě téhož typu: 0D vertex, 1D čára, 2D čtyřúhelník/quad, 3D šestistěn/hexahedron

Nestrukturovaná síť (unstructured grid)

- obecná síť tvořená buňkami různých typů: 0D vertex a polyvertex, 1D čára a lomená čára/polyline, 2D trojúhelník, pixel, quad a polygon, 3D voxel, čtyřstěn, šestistěn, klín, pyramida a dále např. kvadratické buňky
- **explicitní souřadnice vrcholů**, vrcholy explicitně přiřazeny buňkám (**explicitní topologie**)

Polygonální síť (polygonal grid, polydata)

- nestrukturovaná síť s omezeným výběrem typů buněk: 0D vrcholy (polyvertex), 1D čáry (polyline) a 2D polygony
- síť vhodná pro reprezentaci lomených čar a ploch (pro jejich zobrazování užívaná interně)

Formáty vstupních souborů

Data na strukturovaných sítích lze připravit vlastnoručně v nativních formátech knihovny VTK starší generace (**Legacy VTK**) nebo ve formátech, pro které jsou v ParaView připraveny konverzní readery (pro rovnoměrné síť Image, JPEG a PNG reader, pro křivočaré síť CSV a PLOT3D reader). Data na nestrukturovaných sítích se popisují složitěji a uživatel uvítá, když nalezne ke svému generátoru sítě kompatibilní reader.

Legacy VTK formáty (Legacy VTK Reader pro *.vtk)

Soubory s příponou **.vtk** obsahují třířádkovou hlavičku (`# vtk DataFile`), sekci sítě (**DATASET**) a sekce dat definovaných na vrcholech nebo v buňkách (**POINT_DATA** nebo **CELL_DATA**). Sekce sítě obsahuje údaj o počtu vrcholů v každém rozměru (**DIMENSIONS** nx ny nz) a další údaje pro jednoznačný popis jejich polohy. Sekce dat mohou obsahovat data skalární (**SCALARS**), vektorová (**VECTORS**) nebo obecná (**FIELD**). Datový typ souřadnic a samotných dat se uvádí v názvosloví jazyka C (`int`, `float`, `double` aj.). Počet rozměrů se snižuje položením některého z nx, ny a nz rovno 1. Vícezměrná pole musí být uložena s prvním indexem měnícím se nejrychleji, tj. ve fortranském řazení, `print *,data(i,j,k)` neboli `do k=1,nz; do j=1,ny; do i=1,nx; print *,data(i,j,k); enddo; enddo; enddo`. Hlavičky jsou formátované, souřadnice a data mohou být formátované (**ASCII**) nebo binární (**BINARY**) v pořadí bytů BigEndian (standard PC je LittleEndian, pro Legacy VTK je tedy nutná konverze).

Sekce pro rovnoměrnou síť (DATASET STRUCTURED_POINTS) obsahuje kromě řádku s počtem vrcholů (**DIMENSIONS**) souřadnice počátku (**ORIGIN**) a délky kroků (**SPACING**) v každé dimenzi.

Vzor hlavičky, sekce sítě a sekce dat (**template1*.vtk**, též **example1*.vtk** a **TestUnif.f90** pro srdcovou izoplochu):

```
# vtk DataFile Version 3.0          hlavička souboru Legacy VTK
comment (max. 1023 chars)
ASCII                               nebo BINARY
DATASET STRUCTURED_POINTS          sekce pro rovnoměrnou síť
DIMENSIONS nx ny nz
ORIGIN x0 y0 z0
SPACING dx dy dz
POINT_DATA npoints                 sekce dat definovaných na vrcholech, npoints = nx*ny*nz
SCALARS MyScalar float 1           jméno libovolné, typ: int, float, double aj., počet skalárních hodnot 1 až 4
LOOKUP_TABLE default               tabulka barev pro SCALARS, default: přiřadí se až při vizualizaci
npoints values                     řazení: první index nejrychleji, ASCII: libovolné řádkování, BINARY:
VECTORS MyVector float             BigEndian
3*npoints values                   počet SCALARS a VECTORS polí libovolný
CELL_DATA ncells
SCALARS MyCellScalar float 1       sekce dat definovaných v buňkách, ncells = (nx-1)*(ny-1)*(nz-1)
LOOKUP_TABLE default
ncells values
```

Sekce pro rektilineární síť (DATASET RECTILINEAR_GRID) obsahuje tři 1D pole souřadnic vrcholů (**X_COORDINATES** ad.) na každé z kartézských os, tj. $nx+ny+nz$ souřadnic.

Vzor hlavičky a sekce sítě, sekce dat jako v ukázce výše (**template2*.vtk**, též **example2*.vtk**):

```
# vtk DataFile Version 3.0
comment (max. 1023 chars)
ASCII
DATASET RECTILINEAR_GRID          sekce pro rektilineární síť
DIMENSIONS nx ny nz
X_COORDINATES nx float           x souřadnice vrcholů na kartézské ose s y=z=0
nx values
Y_COORDINATES ny float           y souřadnice vrcholů na kartézské ose s x=z=0
ny values
Z_COORDINATES nz float           z souřadnice vrcholů na kartézské ose s x=y=0
nz values
```

Sekce pro křivočarou síť (DATASET STRUCTURED_GRID) obsahuje trojice kartézských souřadnic vrcholů na křivočarých souřadnicových čarách (**POINTS**), tj. $3*nx*ny*nz$ souřadnic.

Vzor hlavičky a sekce sítě (**template3*.vtk**, též **example3*.vtk** a **TestCurv.f90** pro vajíčko, banán, šneka a kapříky):

```
# vtk DataFile Version 3.0
comment (max. 1023 chars)
ASCII
DATASET STRUCTURED_GRID          sekce pro křivočarou síť
DIMENSIONS nx ny nz
POINTS npoints float            npoints = nx*ny*nz
3*npoints values                 trojice x, y, z souřadnic každého vrcholu, první index se mění nejrychleji
```

Ukázka. Příprava sférické sítě s kulovou plochou (podle **TestSphere.f90**)

```
print '(A)', 'DATASET STRUCTURED_GRID'
print '(A,3(1X,I0))', 'DIMENSIONS',nph+1,nth+1,nr+1
print '(A,I0,A)', 'POINTS ',(nph+1)*(nth+1)*(nr+1), ' float'
do ir=0,nr;      r=rmin+(rmax-rmin)*ir/max(nr,1)
do ith=0,nth;   th=thmin+(thmax-thmin)*ith/max(nth,1)
do iph=0,nph;   ph=phmin+(phmax-phmin)*iph/max(nph,1)
x=r*sin(th)*cos(ph); y=r*sin(th)*sin(ph); z=r*cos(th)
print *,x,y,z
enddo; enddo; enddo
```

Sekce pro nestruturovanou síť (DATASET UNSTRUCTURED_GRID) obsahuje trojice kartézských souřadnic vrcholů (**POINTS**), seznam buněk obsahující seznamy vrcholů přiřazených jednotlivým buňkám a ještě jeden seznam buněk obsahující číslo jejich typu (1 vertex, 2 množina vrcholů/polyvertex, 3 čára, 4 lomená čára, 5 trojúhelník, 6 pás trojúhelníků, 7 2D polygon, 8 obdélník/pixel, 9 čtyřúhelník/quad, 10 čtyřstěn, 11 kvádr/voxel, 12 šestistěn, 13 klín, 14 pyramida ad.; typy popsány v části VTK File Formats příručky The VTK User's Guide).

Vzor hlavičky, sekce sítě a sekce dat (**template4*.vtk**, též **example4*.vtk**):

```
# vtk DataFile Version 3.0
comment (max. 1023 chars)
ASCII
DATASET UNSTRUCTURED_GRID           sekce pro nestruturovanou síť
POINTS npoints float                npoints: počet vrcholů sítě
... (3*npoints floats)
CELLS ncells ncelllistsize          ncells: počet buněk sítě
no_of_vertices indices_of_vertices  pro každou buňku: počet vrcholů a jejich indexy
... (ncelllistsize integers)        ncelllistsize=ncells+sum(no_of_vertices)
CELL_TYPES ncells                   pro každou buňku: typ buňky
... (ncells integers)
POINT_DATA npoints                  nebo CELL_DATA ncells
SCALARS MyScalar float              nebo VECTORS jako výše, nebo NORMALS, TENSORS, FIELD aj.
LOOKUP_TABLE default
... (npoints floats)
```

Sekce pro polygonální síť (DATASET POLYDATA) obsahuje trojice kartézských souřadnic vrcholů (**POINTS**) a nepovinné seznamy vrcholů (**VERTICES**), čar (**LINES**), polygonů (**POLYGONS**) a pásů trojúhelníků (**TRIANGLE_STRIP**) obsahující seznamy přiřazených vrcholů.

Vzor hlavičky a sekce sítě (**template5*.vtk**, též **example5*.vtk** a **TestLine.f90** pro lomenou čáru):

```
# vtk DataFile Version 3.0
comment (max. 1023 chars)
ASCII
DATASET POLYDATA                    sekce pro polygonální síť
POINTS npoints float                npoints: počet vrcholů sítě
... (3*npoints floats)
VERTICES nverts nvertlistsize       nverts: počet seznamů vrcholů
no_of_vertices indices_of_vertices  pro každý seznam: počet vrcholů a jejich indexy
... (nvertlistsize integers)        nvertlistsize=nverts+sum(no_of_vertices)
LINES nlines nlinelistsize          nebo POLYGONS npolys npolylistsize
no_of_vertices indices_of_vertices
... (nlinelistsize integers)        nlinelistsize=nlines+sum(no_of_vertices)
```

Reader pro formát Raw binary data (Image Reader pro ***.raw**)

Vytváří **2D/3D rovnoměrnou síť**. V souboru je uloženo **binární datové pole**, jehož datový typ, pořadí bytů (na PC LittleEndian), počet dimenzí a počet skalárních komponent se spolu se souřadnicemi počátku, krokováním a počtem vrcholů sítě vkládá interaktivně v readeru.

Reader pro formát Comma-separated values (CSV Reader pro ***.csv**)

Vytváří tabulku, kterou lze filtrem Table To Structured Grid převést na **křivočarou síť**. Ve formátovaném (ASCII) souboru jsou sloupce se **souřadnicemi a daty** oddělené obvykle čárkou. Zadání počtu vrcholů sítě a spárování souřadnic s příslušnými sloupci tabulky se provádí interaktivně ve filtru. Po aplikaci filtru nutno otevřít vizualizační panel a aktivovat zobrazení manuálně.

Reader pro formát PLOT3D (PLOT3D Reader pro ***.xyz** a ***.q**)

Vytváří **3D křivočarou síť** se skalárními a vektorovými daty. Soubor s příponou **.xyz** obsahuje 1 řádek s rozměry sítě a dále tři reálná 3D pole se souřadnicemi vrcholů: **real,dimension(nx,ny,nz)::x,y,z; print *,nx,ny,nz; print *,x,y,z**. Soubor s příponou **.q** obsahuje opět 1 řádek s rozměry sítě, 1 řádek se 4 irelevantními reálnými skaláry a dále 5 skalárních reálných datových polí, po importu nazvaných Density, (vektor) Momentum a StagnationEnergy: **real,dimension(nx,ny,nz)::d,m1,m2,m3,se; print *,nx,ny,nz; print *,x,x,x,x; print *,d,m1,m2,m3,se**. Soubory mohou být formátované nebo binární; binární data jsou integer pro nx, ny, nz, jinak real(4), řazení bytů na PC je LittleEndian.

Vizualizace (rendering)

Grafické uživatelské rozhraní ParaView je tvořeno 5 částmi: **Menu**, **Toolbars**, **Pipeline Browser**, **Object Inspector** a **View Area**. K přípravě dat slouží **readery** dostupné přes běžné Menu-File-Open nebo předpřipravená data v Menu-Sources. K datům se připojují **filtry** (Menu-Filters, ikony v Toolbars nebo Ctrl-mezerník), konfigurovatelné v Object Inspectoru (karty Properties, Display a Information), a vytvářejí tak v Pipeline Browseru **rouru**, jejíž výstup se zobrazuje ve View Area jako 3D View, 2D View, Spreadsheet View aj. Pohyb kolem objektů ve View Area se řídí pomocí myši, jejich tlačítek a kláves Shift a Ctrl; aktuální přiřazení funkcí **Rotate**, **Pan**, **Zoom** a **Roll** se nalezne v Edit-Settings-Render View-Camera.

Nastavení v Object Inspectoru (OI)

Časným krokem pro zobrazení přiložených příkladů bude přepnutí OI-Display-Style-**Representation** z Outline na Surface. Lokální osy objektu se zapnou tamtéž, OI-Display-Annotation-**Show cube axis**, posun, rotace a škálování objektu se provádí také tam, OI-Display-**Transformation**. Tabulky barev se přiřazují nebo editují cestou přes OI-Display-Color, kde lze po **Edit Color Map** volit z předdefinovaných tabulek pomocí Choose Preset, přeškálovat po Automatically Rescale: No a Rescale Range atd. Průhlednost objektu zvyšuje snížení OI-Display-Style-**Opacity**. Většina uvedených ovládacích prvků je vynesena i mezi Toolbars.

Filtry

Filtrů je přes stovku, pro daná vstupní data jsou dostupné vždy jen některé (Menu-Filters-Alphabetical). Ikony frekventovaných filtrů jsou vyneseny v Toolbars: **Calculator** provádí matematické úpravy dat, **Contour** připravuje izočáry a izopovrchy, **Clip** odřezává části objektu, **Slice** vytváří rovinné řezy, **Threshold** odstraňuje části objektu s daty mimo předepsaný rozsah, **Extract Subset** vybírá bloky z objektu a může podvzorkovat, **Glyph** umístí symboly do vrcholů sítě, **Stream Tracer** vytváří proudnice, **Warp by Vector** posouvá síť podél vektorového pole atd.

Výstupy

Statickým výstupem jsou bitmapové obrázky snímání přes File-**Save Screenshot** ve formátech **PNG**, **BMP**, **JPG**, **PDF** aj. Vizualizace časové řady (číslované soubory se společným základem jména, př. file00.vtk, file01.vtk ad.) se uchová jako **AVI** soubor, File-**Save Animation**. (Možný problém s vytvářením animací lze řešit nastavením Frame Rate na násobek 2, např. 24.) Animovat lze i změnu jakéhokoliv konfigurovatelného prvku jeho navázáním na čas přes Menu-View-**Animation View**. Načtená i filtrovaná data se mohou ukládat ve formátech knihovny VTK a také v CSV aj., File-**Save Data**. Uložit lze i rozpracovaný stav, File-**Save State**.

O zobrazování během interakce

V Edit-Settings-**Render View** lze nastavit velikost v MB, nad kterou se síť během interakce zobrazují ve sníženém rozlišení (Level-of-Detail Threshold). Tamtéž se upravuje prodleva před překreslením objektu po ukončení interakce (Lock Interactive Render).

Binární soubory

Pro zápis binárních souborů se změnou endianity jsou (nejen ve Fortranu) vhodné **proudy** (streams). Rychlost čtení i zápisu souborů je s různými překladači výrazně jiná, a to i v závislosti na operačním systému. Např. ve Windows je **g95** pomalý při čtení a **gfortran** při zápisu a **g95** je rychlý při zápisu a **gfortran** při čtení; na Linuxu pracuje **gfortran** se soubory velmi rychle.

Ukázka. Příprava binárního souboru s rovnoměrnou sítí (podle **TestStream.f90**)

```
OPEN (id,file=trim(filename),form='unformatted',access='stream',status='replace')
```

```
do i=1,nheader; write (id) trim(header_line(i))/new_line(' '); enddo
```

```
do i=1,npoints; write (id) swapEndian(s(i)); enddo
```

```
CLOSE (id)
```

```
FUNCTION swapEndian(x) RESULT (result)
```

```
REAL(4),INTENT(IN) :: x; CHARACTER(4) :: result,z; CHARACTER(1),DIMENSION(4) :: za; EQUIVALENCE (z,za)
```

```
z=transfer(real(x,4),z); za(1:4)=za(4:1:-1); result=z
```

```
END FUNCTION
```

Odkazy

www.paraview.org

Moreland K., **The ParaView Tutorial** (PDF), www.vtk.org/Wiki/The_ParaView_Tutorial

ParaView User's Guide (PDF v share/doc)

The VTK User's Guide, www.kitware.com (PDF k nalezení nesnadno)

Datové soubory: **template*.vtk**, **example*.vtk**, **kaprik*** ad.

Zdrojové texty (Fortran 95): **TestSphere**, **TestStream**, **TestUnif**, **TestCurv**, **TestLine**, **TopoEarth**