

Zpráva s popisem softwarového návrhu a specifikací rozhraní prototypového modulu pro odhad chyby a zjemnění sítě

MEER (Modul for Error Estimation and Refinement) je knihovna sloužící pro odhad chyby řešení metodou konečných prvků (MKP) a následnou adaptivní změnu sítě konečných prvků. Knihovna je napsána v jazyce C++ a vydána pod otevřenou licencí GNU GPL. Zdrojové kódy, volně dostupné na domovské stránce <http://mech.fsv.cvut.cz/meer> včetně uživatelského manuálu, využívají pouze standardní knihovny jazyka C++ a mohou být přeloženy na platformách Linux, Windows i Mac OS. Knihovna funguje jako modul, který může být připojen k libovolnému softwarovému nástroji postavenému na analýze metodou konečných prvků (dále jen MKP řešič).

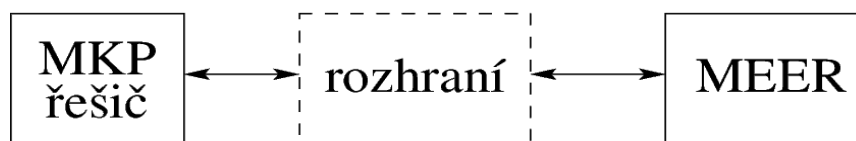
1 Použitý odhad chyby řešení a vylepšení sítě

V inženýrské praxi je v současné době MKP analýza bezpochyby nejpoužívanějším nástrojem pro numerické řešení fyzikálního problému. Metoda je založena na diskretizaci původní spojitě domény soustavou konečných prvků. To vede na pouze přibližné řešení, jehož přesnost závisí na použitém materiálovém modelu, typu konečného prvku, velikosti jednotlivých prvků atd. Rešerše strategií pro odhad chyby řešení a jeho následné „vylepšení“, tak aby bylo dosaženo požadované přesnosti, byla zpracována řešitelským kolektivem v roce 2012. Jako nejefektivnější a přitom dobře algoritmizovatelné se ukázaly projektivní odhady chyby patřící mezi aposteriorní odhady. Konkrétně byly pro implementaci v modulu MEER vybrány projektivní odhady ZZ a SPR a pro vylepšení sítě h-adaptivita. Tyto odhady jsou založeny na vyhodnocení rozdílu mezi toky (v případě stavební mechaniky napětími) získanými MKP a vylepšenými toky získanými vhodnou projekcí (vyhlazením, průměrováním) z tzv. superkonvergenčních bodů (obvykle Gaussových integračních bodů). Vylepšená napětí lze stanovit buď globální projekcí, jež minimalizuje jejich odchylku od vypočtených hodnot ve smyslu metody nejmenších čtverců (ZZ metoda), nebo projekcí lokální, která vyžaduje řešení malých soustav, jedné pro každý uzel konečněprvkové sítě, vzniklých regresí (opět ve smyslu metody nejmenších čtverců) skrze hodnoty v integračních bodech prvků sdílejících tento uzel. Z důvodů jednodušší komunikace mezi modulem a MKP řešičem byla během implementace upřednostněna metoda SPR.

Na základě velikosti odhadnuté chyby a jejího rozložení po řešené doméně můžeme zahustit nebo zředit síť konečných prvků (h-adaptivita) tak, aby bylo dosaženo rovnoměrného rozložení chyby o předepsané hodnotě.

2 Popis softwarového návrhu

Návaznost programů a tok dat mezi nimi je zobrazen na obrázku 1. Jako MKP řešič je zatím předpokládán jakýkoliv softwarový balík pro statickou analýzu stavebních konstrukcí založený na metodě konečných prvků, který jako výsledek výpočtu vrací posuny v uzlech a napětí v integračních bodech sítě. V budoucnu by měly být podporovány výpočty libovolných konstrukcí, namáhaných nejen mechanicky, ale i například teplotou.



Pro komunikaci mezi řešičem a modulem MEER je potřeba implementovat rozhraní, které se bude starat o potřebnou výměnu dat. Nejefektivnější způsob propojení je implementace rozhraní přímo do kódu MKP řešiče a přilinkování knihovny MEER. Pro vývojovou pilotní aplikaci bylo rozhraní a knihovna ověřeny v prostředí MIDAS (vyvinuto v rámci projektu MPO FRI-T1/568 – Integrované nástroje pro generativní design a zvýšení konkurenceschopnosti české architektury) sloužícím pro automatizované zpracování vstupních a výstupních dat MKP výpočtů. Jako řešič pro testování byl zvolen konečněprvkový balík OOFEM ve spojení s generátorem sítě T3d.

Protože vybraný projektivní odhad chyby (SPR i ZZ) patří mezi aposteriorní metody, je potřeba před vlastním voláním knihovny MEER provést výpočet /analýzu zadané konstrukce. Knihovna pak provede odhad chyby řešení a navrhne změnu sítě. Rozepsání jednotlivých kroků:

1. zadání konstrukce (MKP řešič),
2. generování počáteční sítě, výpočet (MKP řešič),
3. předání výsledků modulu (rozhraní),
4. odhad chyby řešení a návrh změny sítě (modul MEER),
5. předání požadavků na hustotu sítě řešiči (rozhraní),
6. generování modifikované sítě, výpočet (MKP řešič).

Body 3-6 se mohou několikrát zopakovat, dokud není dosaženo uspokojivého výsledku, tj. chyba řešení je dostatečně blízko nebo nižší chyby předepsané.

3 Dokumentace knihovny MEER a specifikace datového rozhraní

Metoda SPR je založená na průměrování hodnot napětí z elementů na uzly. To však vyžaduje spojitě pole napětí na průměrované oblasti. Řešená konstrukce se proto musí rozdělit na tzv. *regiony*, což jsou takové části konstrukce, uvnitř kterých zůstávají průřezové a materiálové charakteristiky spojitě a počty složek napětí v integračních bodech jsou stejné.

Pro odhad chyby je dále třeba definovat dvě sady integračních bodů na elementu, které budeme označovat jako *IPs1* a *IPs2*. První sada by měla obsahovat tolik integračních bodů, kolik je třeba pro numerickou integraci interpolační funkce pro průběh napětí po elementu. Obvykle je to taková sada, která se používá pro integraci matice tuhosti elementu. Druhá sada by měla obsahovat tolik integračních bodů, kolik je třeba pro numerickou integraci interpolační funkce umocněné na druhou. Obvykle je to taková sada, která se používá pro integraci matice hmotnosti elementu.

Ke všem seznamům komponent (uzlů, elementů, integračních bodů atd.) se přistupuje stejným způsobem. Prvky seznamu mají přiřazeno identifikační číslo (ID). V číslování nejsou mezery a začíná se od nuly.

Celá výkonná část knihovny MEER, tj. odhad chyby a předpis pro adaptivní změnu sítě, se nachází v jedné třídě *MEERadaptivity*. Ta je doplněna souborem obecných funkcí, které poskytují základní matematickou funkcionalitu, např. vyčíslení vektorového součinu, řešič soustavy lineárních rovnic atd. Metody (členské funkce) i atributy (členské proměnné) třídy *MEERadaptivity* se dělí do tří funkčních bloků, které se shodují s rozdělením podle typu přístupových práv. Jsou to metody a atributy veřejné, chráněné a soukromé uvedené klíčovými slovy *public*, *protected* a *private*.

Datové rozhraní mezi MKP řešičem a modulem MEER je prakticky tvořeno metodami chráněnými (vstup dat) a veřejnými (výstup dat).

3.1 Chráněné metody a atributy

Chráněné metody a atributy slouží pro import výsledků z MKP analýzy do knihovny MEER. V třídě *MEERadaptivity* jsou proto deklarovány jako čistě virtuální (pure virtual) a definovány jsou až v třídě zděděné. Je zřejmé, že metody pro import výsledků jsou implementačně závislé. Proto zděděná třída nemůže být součástí knihovny MEER, ale musí být vytvořena pro každý MKP řešič zvlášť a je jeho součástí. Tvoří tak fakticky „rozhraní“ popisované v druhé kapitole.

Přehled chráněných metod s návodem na implementaci v zděděné třídě:

void initialize (void)

Funkce nemá parametry a nic nevrací. Má za úkol naplnit chráněné atributy v kterých se uchovávají základní, ve třídě *MEERadaptivity* často používané, charakteristiky řešené úlohy. Atributy typu ukazatel na pole musí být alokovány uvnitř této funkce. Dealokovány jsou v destruktoru třídy *MEERadaptivity*. Seznam chráněných atributů:

long **nRegions** – počet regionů v konstrukci.

long ***region_nvals** – počet složek průměrované veličiny (napětí) v integračních bodech.

long **nn** – počet uzlů sítě konečných prvků.

long **ne** – počet konečných prvků (elementů) sítě.

long ***regids** – pole délky **ne** uchovává identifikační číslo (ID) regionu (číslováno od nuly) do kterého příslušný element patří.

long ***nelemnodes** – pole délky **ne** uchovává počet uzlů příslušného elementu.

long ****elemnodes** – pole (délky **ne**) ukazatelů na pole (délky **nelemnodes[i]**), které uchovává identifikační čísla (číslováno od nuly) uzlů příslušného (i-tého) elementu.

void give_superelems_to_node (long nodeID, long &nsuperelems, long *superelems)

Funkce vrací identifikační čísla elementů přilehlých k danému uzlu, tj. elementů, na kterých uzel leží.

Vstupní parametry:

long **nodeID** – ID uzlu.

Výtupní parametry:

long **&nsuperelems** – počet přilehlých elementů.

long ***superelems** – pole identifikačních čísel přilehlých elementů délky **nsuperelems**, pole je předem alokováno na délku 64.

void give_node_coords (long nodeID, double *coords)

Funkce vrací globální souřadnice daného uzlu.

Vstupní parametry:

long **nodeID** – ID uzlu.

Výtupní parametry:

`double *coords` – pole se souřadnicemi (předem alokované na délku 3).

`double give_element_volume (long elementID)`

Funkce jako návratovou hodnotu vrací objem elementu s ID **`elementID`**. U plošných/prutových prvků je objem počítán jako plocha/délka přenásobená tloušťkou/plochou průřezu.

`long give_number_of_IPs1 (long elementID)`

Funkce jako návratovou hodnotu vrací počet integračních bodů sady IPs1 elementu s ID **`elementID`**.

`long give_number_of_IPs2 (long elementID)`

Funkce jako návratovou hodnotu vrací počet integračních bodů sady IPs2 elementu s ID **`elementID`**.

`void give_EDl_natcoords_weight_IPs1 (long elementID, int ipID, A_ElemDisplInterpol &edi, double *ncoords, double &w)`

Funkce zaprvé vrací typ interpolace posunů po prvku s ID **`elementID`** pomocí výčtového typu

`enum A_ElemDisplInterpol { A_EDl_Void, A_EDl_2d_3n_linear, A_EDl_2d_4n_bilinear },`

kde **`2d`** značí dimenzi elementu, **`3n`** značí počet uzlů na elementu a **`linear`** popisuje stupeň aproximace.

Dále funkce vrací přirozené souřadnice a váhu integračního bodu s ID **`ipID`** na elementu s ID **`elementID`** pro integrační sadu IPs1.

Vstupní parametry:

`long elementID` – ID elementu.

`long nodeID` – ID uzlu.

Výtupní parametry:

`A_ElemDisplInterpol &edi` – typ interpolace posunů po prvku.

`double *ncoords` – pole s přirozenými souřadnicemi integračního bodu (předem alokované na délku 3).

`double &w` – váha integračního bodu.

`void give_EDl_natcoords_weight_IPs2 (long elementID, int ipID, A_ElemDisplInterpol &edi, double *ncoords, double &w)`

Viz předchozí funkce. Hodnoty odpovídají integrační sadě IPs2.

`const double* give_IP1_values (long elementID, int ipID)`

Funkce jako návratovou hodnotu vrací konstantní ukazatel na pole typu `double`, v kterém jsou uloženy

hodnoty napětí pro integrační bod s ID **ipID** na elementu s ID **elementID** pro integrační sadu IPs1.

```
const double* give_IP2_values (long elementID, int ipID)
```

Funkce jako návratovou hodnotu vrací konstantní ukazatel na pole typu double, v kterém jsou uloženy hodnoty napětí pro integrační bod s ID **ipID** na elementu s ID **elementID** pro integrační sadu IPs2.

3.2 Veřejné metody

Veřejné metody slouží pro spuštění výpočtu a pro předávání výsledků.

Přehled metod:

```
void run (double req_error)
```

Funkce **run** je hlavní, a jediná, výkonná funkce knihovny MEER. Ve funkci je proveden odhad chyby a následně je navržena změna hustoty sítě konečných prvků tak, aby bylo dosaženo požadované chyby **req_error** (hodnota je zadána v procentech).

```
double give_error (void)
```

Funkce vrací hodnotu vypočtené chyby pro celou konstrukci.

```
const double* give_element_error (void)
```

Funkce vrací konstantní ukazatel na pole délky „počet elementů“. V poli jsou uloženy vypočtené chyby na jednotlivých elementech. Hodnoty jsou v procentech.

```
const double* give_mesh_refinement (void)
```

Funkce vrací konstantní ukazatel na pole délky „počet uzlů“. V poli jsou uloženy vypočtené hodnoty charakteristické velikosti elementů sítě konečných prvků v místě příslušných uzlů. Tyto hodnoty slouží pro generování nové sítě požadované hustoty.

3.3 Soukromé metody a atributy

Soukromé metody a atributy nejsou přístupné pro používání z vnějšku třídy ani z třídy zděděné a slouží pro vlastní výpočet. V prvním kroku je napětí získané z MKP výpočtu interpolováno pomocí metody SPR do uzlů. Rozdíl mezi tímto tzv. „vylepšeným polem napětí“ a původním vypočteným napětím je označen za chybu výpočtu. Na základě chyby je pak navržena hustota nové sítě, která by měla mít rovnoměrné rozložení chyby o požadované hodnotě. Tuto hodnotu zadává uživatel a zadává ji pomocí parametru funkce **run**. Pokud výsledná síť nemá být neúměrně hustá, neměla by být předepsaná chyba příliš malá. Měla by se orientačně pohybovat v rozmezí 5-20% dle typu úlohy. Vypočtená chyba řešení a nebo přímo předpis pro generování nové sítě je pak vrácen MKP řešiči jako výsledek výpočtu modulu MEER.