FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Lineární řešiče v OpenFOAM

Semestrální práce

KMA/PVM

Jan Půlpán

22. dubna 2021

1 Teoretický úvod

OpenFOAM je sada výpočetních nástrojů pro numerické simulace CFD (computational fluid dynamics) problémů, tedy problémů zabývajících se prouděním tekutin, vedením tepla a podobných procesů. OpenFOAM využívá k hledání řešení metody konečných objemů (FVM). My se budeme zabývat jen jednou částí celého řeťezce řešení problému a to sice popisem lineárních řešičů.

FVM je numerická metoda na řešení parciálních diferenciálních rovnic na dané 3D geometrii převedené na síť nepřekrývajích se elementů (konečných objemů). Tu pak pomocí diskretizace převedeme na soustavu algebraických rovnich, kterou řešíme pomocí lineárních řešiců a výsledné řešení obsahuje hodnotu sledovaných proměnných pro každý z těchto elementů.

Budeme se zabývat jen posledním článkem řetězce, tedy lineárními řešiči. Na testovací úloze si ukážeme vhodnost jednotlivých řešičů a budeme sledovat vliv parametrů na konvergenci řešení. Úkolem lineárních řešičů (dále budeme používat jen „řešičů“, pokud nebude hrozit záměna kontextu) je najít řešení soustavy algebraických rovnic tvaru

$$\boldsymbol{A}x = b, \tag{1.1}$$

kde $\boldsymbol{A}$ je regulární matice popisující diskretizovaný model, x vektor závislé proměnné a b vektor pravé strany.

Pokud matice $\boldsymbol{A}$ umožňuje snadnou inverzi, je nejjednodušší získat přesné řešení soustavy přímým řešičem vztahem

$$x = \boldsymbol{A}^{-1}b.$$

Většinou ale nejde inverzi snadno nalézt a v tom případě se používají přibližné řešiče iterační. První variantou jsou stacionární iterační metody, kdy matici $\boldsymbol{A}$ můžeme rozdělit (například pomocí LU rozkladu) na $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$. matice $\boldsymbol{M}$ musí být snadno invertovatelná a řešení pak hledáme poomocí iterací

$$x^{(k+1)} = \boldsymbol{M}^{-1}(\boldsymbol{N}x^{(k)} + b) \tag{1.2}$$

Pevný bod operátoru (1.2) je pak přesným řešením soustavy (1.1).

Druhou používanou metodou iteračních řešičů jsou metody Krylovových podprostorů. Hledáme tak přibližné řešení v podprostorech s rostoucí dimenzí generovaných čtvercovou maticí $\boldsymbol{A}$ a vektorem $\boldsymbol{b}$. Krylovův podprostor $r$tého řádu je definován jako

$$\mathcal{K}_r(\boldsymbol{A},\boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{r-1}\boldsymbol{b}\}.$$

TROCHU LÍP POPSAT, KOUKNOUT DO NA

Iterační metody obecně měří chybu aproximace řešení pomocí reziduí, tedy dosazením aproximovaného řešení do původních rovnic a vypočtením rozdílu oproti pravé straně $\boldsymbol{b}$. Reziduum $\boldsymbol{r}$ definujeme jako

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}x. \tag{1.3}$$

Často se při numerickém řešení lineárních algebrických rovnic používá **předpodmínění**, které zajistí k rychlejší šíření informace na výpočetní mřížce. Při levém předpodmínění (existuje i pravé a centrální) Předpokládáme soustavu ve tvaru

$$\boldsymbol{M}^{-1}\boldsymbol{A}x = \boldsymbol{M}^{-1}\boldsymbol{b}$$

kde $\boldsymbol{M}$ je předpodmiňovač. Ten zajistí, že konvergence předpodmíněného systému je rychlejší, než bez něj. $\boldsymbol{M}$ je často uvažována jako snadno invertovatelná aproximace matice $\boldsymbol{A}$. Nejjednodušší volbou je tak $\boldsymbol{M} = \boldsymbol{I}$, což ovšem vede k nepodmíněnému problému. Ideální je naopak $\boldsymbol{M} = \boldsymbol{A}$, to ovšem znamená, že inverze předpodmiňovače je stejně náročná jako původní matice. JAK VYPADÁ ITERAČNÍ METODA - ZASE

SMOOTHER

Although the preconditioners discussed before can considerably reduce the number of iterations, they do not normally reduce the mesh dependency of the numbers of iterations. OpenFOAM supplies the following smoothers to be used with the solvers in the smoothers/ directory:

These smoothers are discussed in Section 9. For now, it suffices to view them as basic iterative methods (e.g. Gauss-Seidel) which effectively smooth out the error associated with the current approximate solution

2 | Lineární řešiče v OpenFOAM

OpenFOAM obsahuje kromě základního přímého řešiče `diagonalSolver` tři základní typy řešičů:

1. řešiče metodou konjugovaných gradientů
2. smooth řešiče
3. multigrid řešiče

Jednotlivé řešiče jsou popsány v tabulce 2.1.

Tabulka 2.1: řešiče v rámci OpenFOAM

Řešič	**Označení**	**Matice $\boldsymbol{A}$**
Preconditioned conjugate gradient	PCG	sym
Preconditioned bi-conjugate gradient	PBiCG	asym
Stabilized Preconditioned (bi-)conjugate gradient	PBiCGStab	sym/asym
Solver using a smoother	smoothSolver	sym/asym
Generalised geometric-algebraic multi-grid	GAMG	sym/asym
Diagonal solver for explicit systems	diagonal	

To který řešič je možné použít je závislé na matici $\boldsymbol{A}$ a tom, jestli je symetrická nebo nesymetrická. Symetrie $\boldsymbol{A}$ samozřejmě závisí na rovnicích, které chceme řešit. V případě, že zvolíme nesprávný řešič vzhledem k symetričnosti matice $\boldsymbol{A}$ OpenFOAM sám volbu opraví.

Reziduum je během jednotlivých iterací vyhodnocováno pomocí vztahu (1.3). Každý řešič má k reziduu trochu jiný přístup, obecně ale dochází k normalizaci a škálování rezidua vztahy

$$n = \sum \left(|\boldsymbol{A}x - \boldsymbol{A}\hat{x}| + |\boldsymbol{b} - \boldsymbol{A}\hat{x}| \right),$$
$$r = \frac{1}{n} \sum |\boldsymbol{b} - \boldsymbol{A}x|.$$

Pro každý typ řešice se nastaví parametr pro velikost rezidua (`tolerance`) a také relativní toleranci (`relTol`) pro poměr aktuálního rezidua v rámci jedné iterace a počátečního rezidua před začátkem iterace. Řešič se následně zastaví pokud je splněna alespoň jedna z těchto podmínek:

1. reziduum je menší než nastavená `tolerance`,
2. relativní tolerance je menší než nastavená `relTol`,

3. je dosaženo nastaveného maximálního počtu iterací `maxIter`.

Metoda sdružených gradientů (conjugent gradients) - PCG. PBiCG, PBiCGStab

Metoda sdružených gradientů (CG) je iterační algoritmus řešení soustav lineárních rovnic, který využívá sdružených směrů a vyžaduje symetrickou matici (postivně definitní) matici $\boldsymbol{A}$. Proto vznikla i metoda "bi-conjugate gradients"(dvojitých sdružených směrů), která pracuje s asymetrickými maticemi. OpenFoam obsahuje řešiče PCG a PBiCG s implementací těchto řešičů. Obě varianty jsou předpodmíňěné. Navíc je obsažen i PBiCGStab řešič, který umí pracovat se symetricvkou i nesymetrickou maticí $\boldsymbol{A}$.

Předpodmičovač je volitelný a obsahuje následující možnosti:

Preconditioner Keyword Diagonal incomplete-Cholesky (symmetric) DIC Faster diagonal incomplete-Cholesky (DIC with caching) FDIC Diagonal incomplete-LU (asymmetric) DILU Diagonal diagonal Geometric-algebraic multi-grid GAMG No preconditioning none

NĚCO O SMOOTH ŘEŠIČÍCH

The solvers that use a smoother require the smoother to be specified. The smoother options are listed in Table 6.11. Generally GaussSeidel is the most reliable option, but for bad matrices DIC can offer better convergence. In some cases, additional post-smoothing using GaussSeidel is further beneficial, i.e. the method denoted as DICGaussSeidel

Smoother Keyword Gauss-Seidel GaussSeidel Diagonal incomplete-Cholesky (symmetric) DIC Diagonal incomplete-Cholesky with Gauss-Seidel (symmetric) DIC-GaussSeidel

NĚCO O MULTIGRID ŘEŠIČÍCH (MULTIGRID METODY)

The generalised method of geometric-algebraic multi-grid (GAMG) uses the principle of: generating a quick solution on a mesh with a small number of cells; mapping this solution onto a finer mesh; using it as an initial guess to obtain an accurate solution on the fine mesh. GAMG is faster than standard methods when the increase in speed by solving first on coarser meshes outweighs the additional costs of mesh refinement and mapping of field data. In practice, GAMG starts with the mesh specified by the user and coarsens/refines the mesh in stages. The user is only required to specify an approximate mesh size at the most coarse level in terms of the number of cells nCoarsestCells.

NĚCO O RESIDUAL CONTROL

By default, cases will run until the time settings are achieved in the case controlDict dictionary. Alternatively, the residualControl object can be added to the fvSolution dictionary to enable additional controls. This operates in two modes:

Steady state To terminate the case when the initial residual of the field equations falls below user-specified threshold values for SIMPLE-based solvers:

residualControl p 1e-2; "(Ux Uy)"1e-4; "(k|epsilon|omega)"1e-4;

NĚCO O RELAXACI

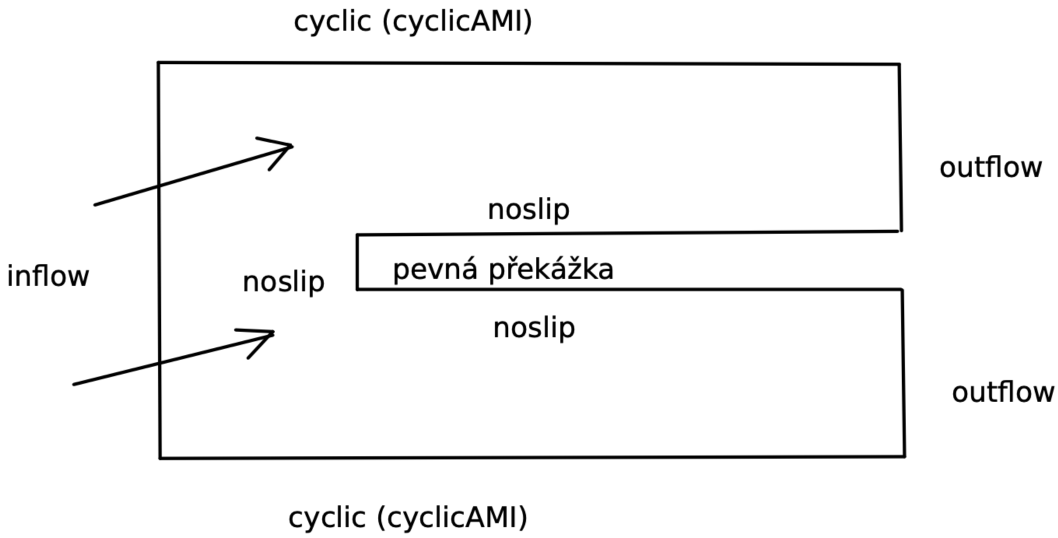
under-relaxation, a technique used for improving stability of a computation, particularly in solving steady-state problems. Under-relaxation works by limiting the amount which a variable changes from one iteration to the next, either by modifying the solution matrix and source prior to solving for a field or by modifying the field directly. An under-relaxation factor $\alpha, 0 < \alpha \leq 1$ specifies the amount of under-relaxation, ranging from none at all for $\alpha = 1$ and increasing in strength as $\alpha- > 0$ The limiting case where $\alpha = 0$ represents a solution which does not change at all with successive iterations. An optimum choice of α is one that is small enough to ensure stable computation but large enough to move the iterative process forward quickly; values of α as high as 0.9 can ensure stability in some cases and anything much below, say, 0.2 are prohibitively restrictive in slowing the iterative process

1. jaké jsou možnosti, jaké jsou parametry
   (a) diagonalSolver - pokud je matic A diagonální, pak vypočítám $A^{-1} = A/diag(A)$ a jsem hotov (https://www.openfoam.com/documentation/guides/latest/solvers.html)

3 Porovnání řešičů

Lineární řešiče porovnáme na konkrétní úloze stacionárního proudění v lopatkové mříži. Zachováme přitom jednotné parametry nastavení úlohy až na nastavení jednotlivých lineárních řešičů. Zajímat nás bude konvergence řešení a výpočetní náročnost. Zadání úlohy je na obrázku 3.1, který definuje geometrii úlohy i okrajové podmínky.

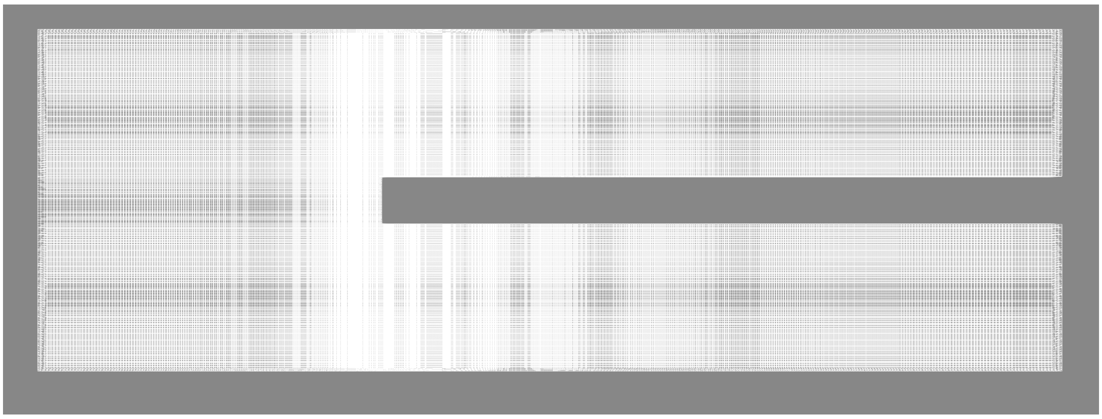
Používáme: simpleFoam - Steady-state solver for incompressible, turbulent flows



Obrázek 3.1: Popis řešeného modelu

Nejprve musíme definovat síť na které budeme naši úlohu řešit. .... Nadefinovaná síť je na obrázku 3.2, konkrétní definice v přiloženém souboru `blockMeshDict`.

PŘIDAT OBRÁZKY S DETAILAMA OKOLO TOHO VYKOUSNUTÍ, UKÁZAT, ŽE TO JE NEHOMOGENNÍ, SPOČÍTAT KOLIK JE TAM VLASTNĚ TĚCH POLÍČEK



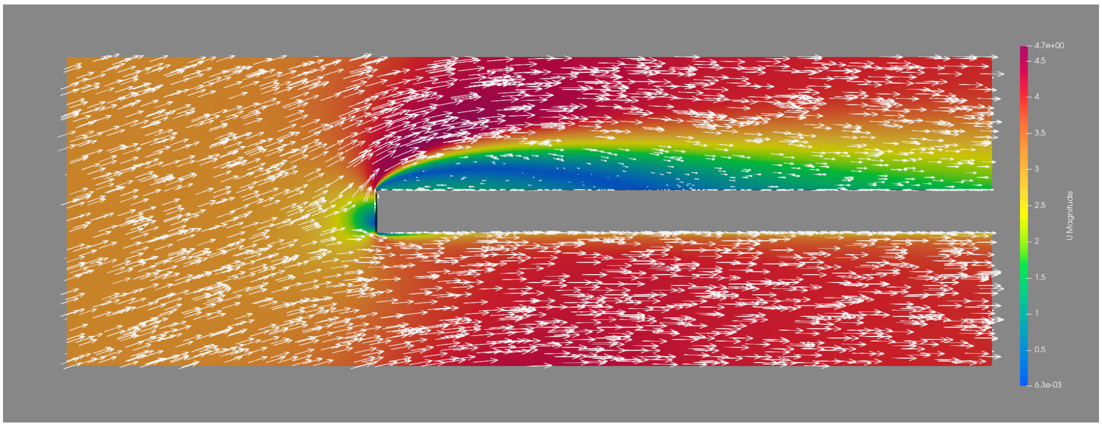
Obrázek 3.2: Síť řešeného modelu

Okrajové podmínky pro jednotlivé veličiny, pro nás primárně pro rychlost U a tlak p jsou definovány v souborech ve složce `0`.

Ostatní parametry řešiče úlohy, netýkající se přímo lineárních řešičů jsou pro všechny simulace totožné a jsou v souborech `transportProperties`, `turbulenceProperties`, `fvSchemes` a `controlDict`. Zde uvedeme jen vybrané parametry popisující naší úlohu.

• application	simpleFoam
• simulationType	RAS
• RASmodel	kEpsilon
• turbulence	on
• transportModel	Newtonian
• nu	1e-05

Na ukázku je zde obrázek 3.3 výsledného klidového stavu pořízený v ParaView. Výsledky úlohy řešené pomocí různých lineárních řešičů vypadají identicky.



Obrázek 3.3: Klidový stav s vektory rychlosti U v ParaView

Na lineárních řešičích budeme zkoumat vliv následujících parametrů na konvergenci simulace:

- typ použitého řešiče
- typ použitého předpodmiňovače
- typ použitého smootheru
- pod-relaxace

Nastavení řešičů se v OpenFOAM dělá v souboru system/fvSolution. Ukázkový soubor je v příloze A.

Nejprve nás zajímá typ použitého řešiče, případně přidruženého předpodmiňovače nebo smootheru. V tabulkách 3.1, 3.2 a 3.3 jsou uvedená nastavení a výsledné počty iterací řešení úlohy a čas běhu simulace testovací úlohy.

Konvergenci metody měříme pomocí počtu využitých iterací metody. Maximální počet iterací byl nastaven na 500. Tolerance lineárního řešiče je shodně nastavena na 10^{-6}, relativní tolerance na 0. Velikost reziduí v rámci SIMPLE algoritmu pomocí `residualControl` parametru pro tlak p nastaveno na 10^{-3} a pro rychlost U na 10^{-4}.

Tabulka 3.1: Nastavení lineárních řešičů pro jednotlivé simulace

Tlak p: **GAMG**		Rychlost U: **GAMG**			
Předpod.	Smoother	Předpod.	Smoother	# iter	Čas
—	DIC	—	DILU	475	157.94
—	GaussSeidel	—	GaussSeidel	475	176.99

Tabulka 3.2: Nastavení lineárních řešičů pro jednotlivé simulace

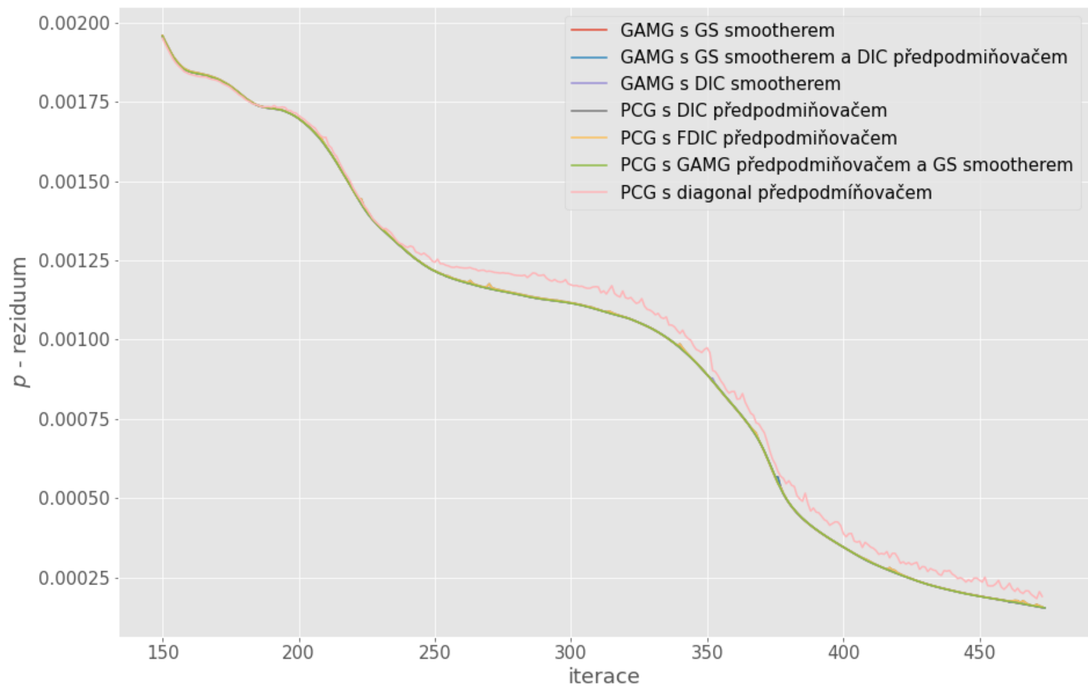
Tlak p: **PCG**		Rychlost U: **PBiCG**			
Předpod.	Smoother	Předpod.	Smoother	# iter	Čas
DIC	—	DILU	—	474	540.48
FDIC	—	DILU	—	474	531.14
GAMG GS smooth.	—	GAMG GS smooth.	—	475	184.85
diagonal	—	diagonal	—	474	679.26

Tabulka 3.3: Nastavení lineárních řešičů pro jednotlivé simulace

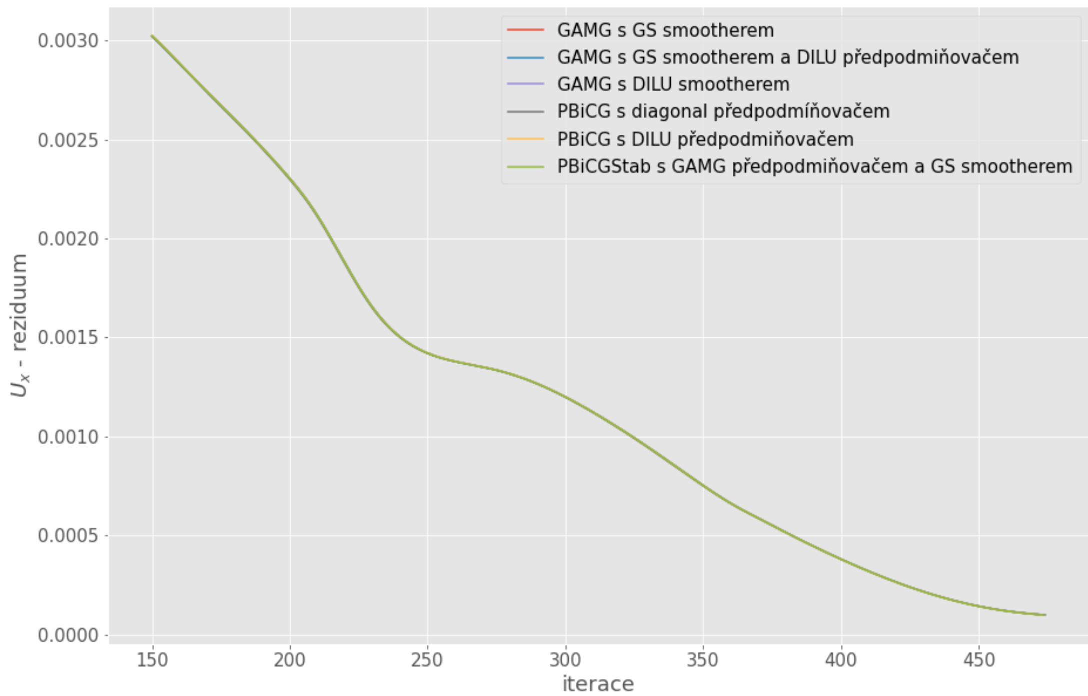
Tlak p: **smoothSolver**		Rychlost U: **smoothSolver**			
Předpod.	Smoother	Předpod.	Smoother	# iter	Čas
—	DIC	—	GaussSeidel	500+	1136.21
—	DIC	—	DILU	500+	1039.73
—	GaussSeidel	—	GaussSeidel	500+	1011.42
FDIC	GaussSeidel	—	GaussSeidel	500+	1008.47
—	symGauss-Seidel	—	symGauss-Seidel	500+	1370.58

GAMG A PCG/PBiCG JSOU V POŘÁDKU. SMOOTH ALE ANI JEDEN NEDOBĚHL. PRŮBĚH JEDNOTLIVÝCH SIMULACÍ POMOCÍ VÝVOJE REZIDUA JE NA OBRÁZCÍCH XYZ (PRO P) A XZZ (PRO XOVOU SOUŘADCICI U).

NEJLEPŠÍCH VÝSLEDKŮ DOSAHUJE GAMG ŘEŠIČ. - VÍME PROČ? - OBECNĚ ZNÁMÝ FAKT, SPECIÁLNĚ PRO P. GAMG NENÍ ALE TAK ŠKÁLOVATELNÝ JAKO PCG/PBICG A TAK JE MOŽNÉ, ŽE PROP JINOU A SLOŽITEJŠÍ ÚLOHU BY MOHLO BÝT PCG VHODNĚJŠÍ.



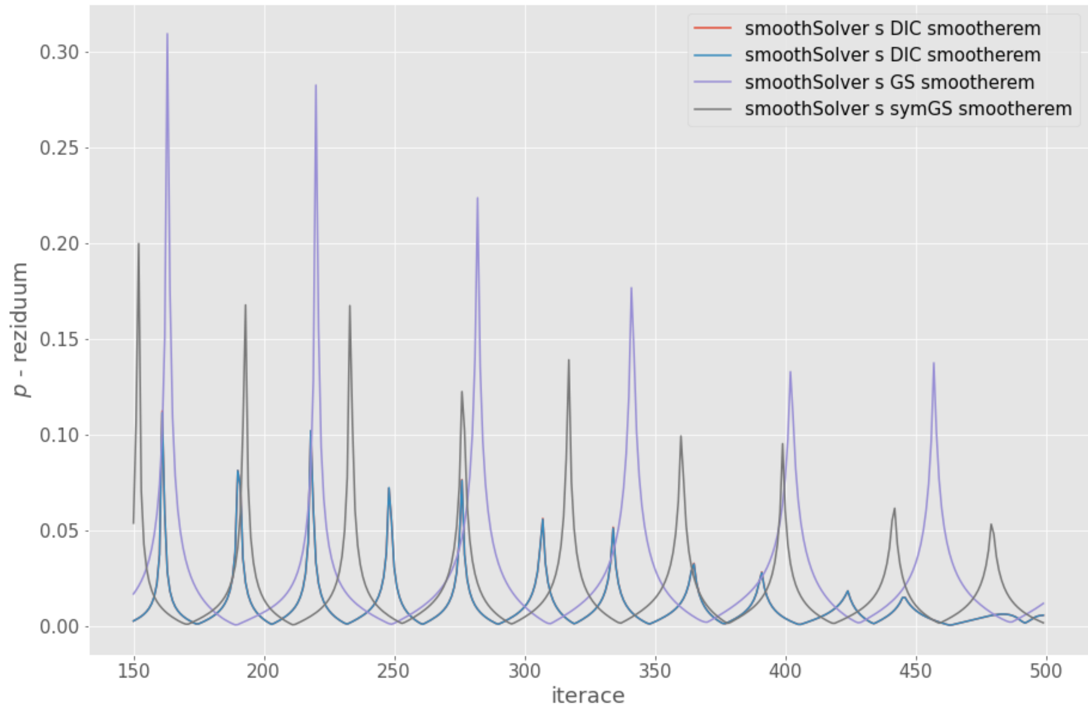
Obrázek 3.4: Konvergence lineárních řešičů v různé konfiguraci pro tlak p



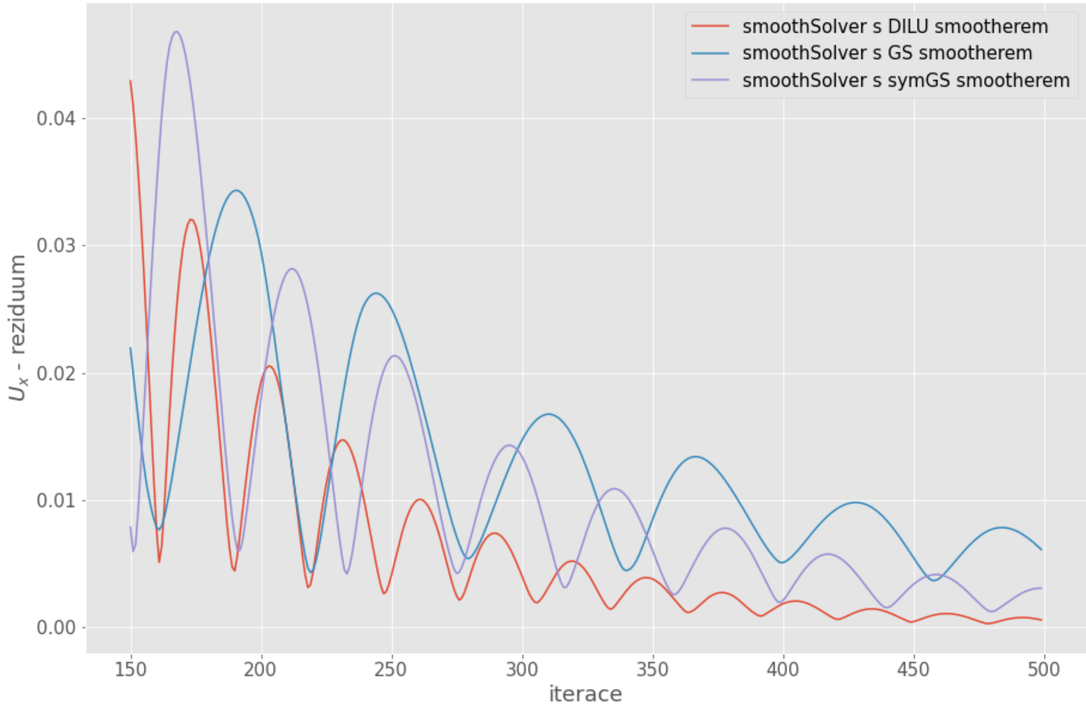
Obrázek 3.5: Konvergence lineárních řešičů v různé konfiguraci pro rychlost U_x

NA OBRÁZCÍCH ASD A BNM JE VIDĚT PROBLÉM SE SMOOTH ŘEŠIČI. N EJEN ŽE JSOU EXTRÉMNĚ POMALÉ, ALE JEŠTĚ K TOMU NEJSOU

SCHPNY V NASTAVENÉM POČTU MAXITER DOKONVERGOVAT. POPSAT OSCILACE A ŽE JDOU PROTI SOBĚ. KDYŽ NAJDU DŮVOD NEVHODNOSTI SMOOTH ŘEŠIČE, TAK DOPSAT.



Obrázek 3.6: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro tlak p



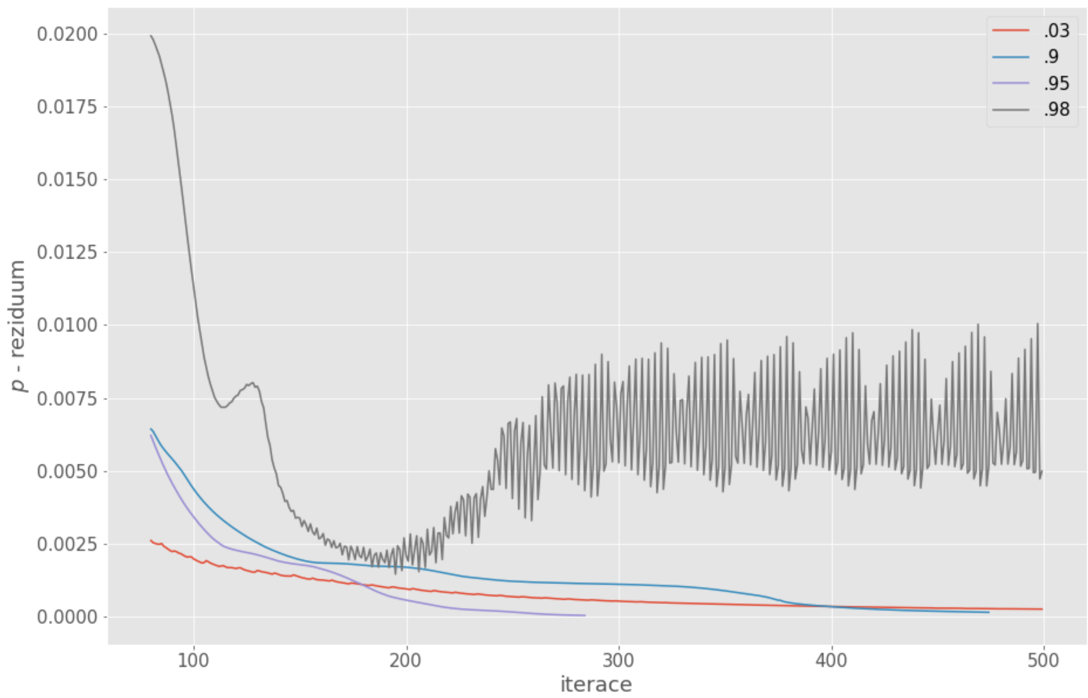
Obrázek 3.7: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x

DÁLE NÁS BUDE ZAJÍMAT POD-RELAXACE (VYSVĚTLENÍ JE JIŽ NAHOŘE) PRIMÁRNĚ U GAMG. V TABULCE HJK JSOU VÝSLEDKY. UŽ TADY OKOMENTOVAT.

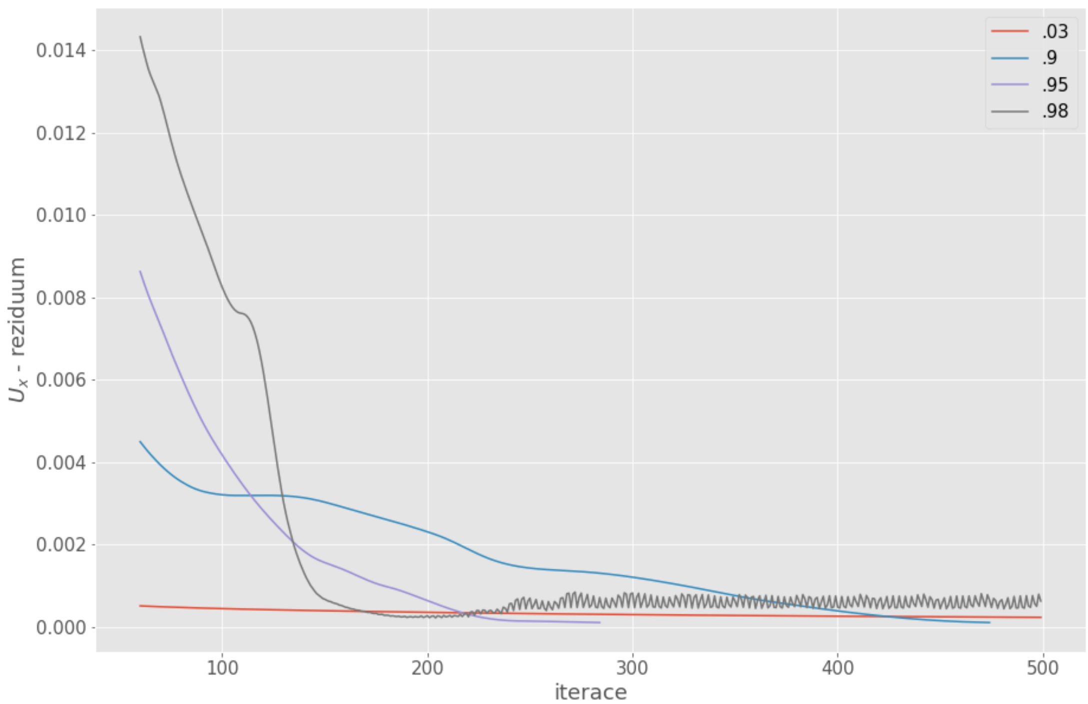
Tabulka 3.4: Nastavení lineárních řešičů pro jednotlivé simulace

Tlak p: **GAMG**		Rychlost U: **GAMG**				
Předpod.	Smoother	Předpod.	Smoother	Relaxace	# iter	Čas
—	GaussSeidel	—	GaussSeidel	0.9	475	176.99
—	GaussSeidel	—	GaussSeidel	0.3	500+	151.49
—	GaussSeidel	—	GaussSeidel	0.95	285	114.15
—	GaussSeidel	—	GaussSeidel	0.98	500+	257.28
DIC	GaussSeidel	DILU	GaussSeidel	0.98	475	160.59

NA OBRÁZCÍCH X A Y JE OPĚT PRŮBĚH REZIDUA. Z NICH JE VIDĚT NESTABILITA .98 RELAXACE JAK V P TAK V U



Obrázek 3.8: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x



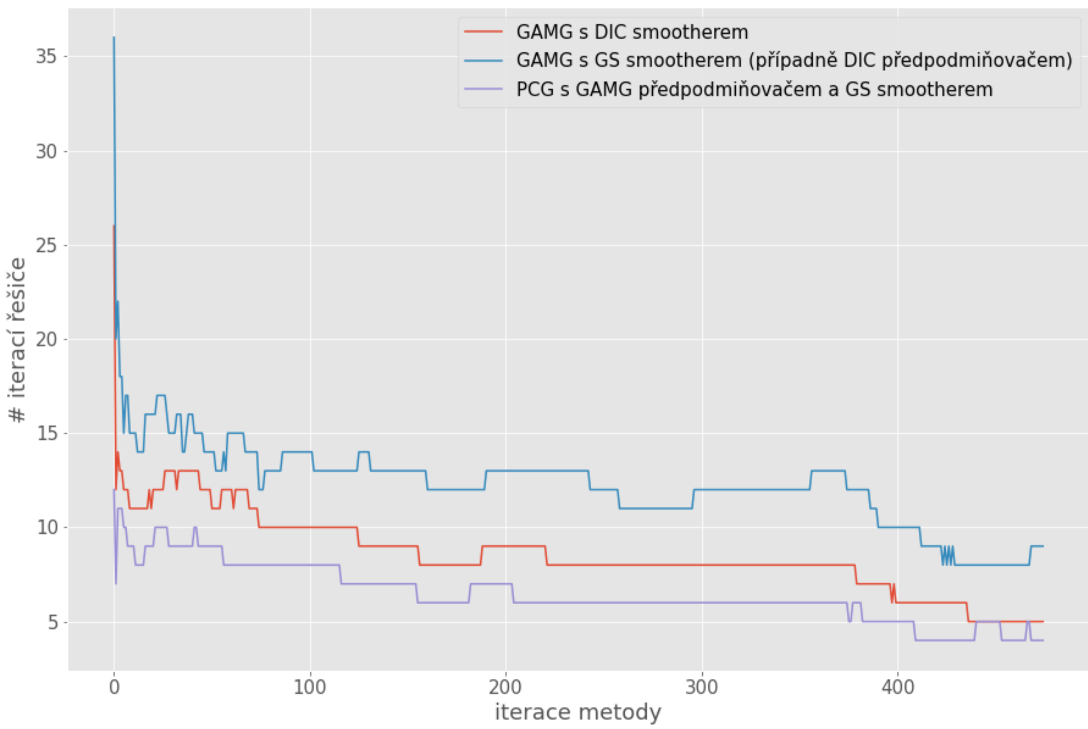
Obrázek 3.9: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x

POKUS POUŽÍT UNDERRELAX U SMOOTH NEVYŠEL, VÝSLEDKY LEPŠÍ NEJSOU

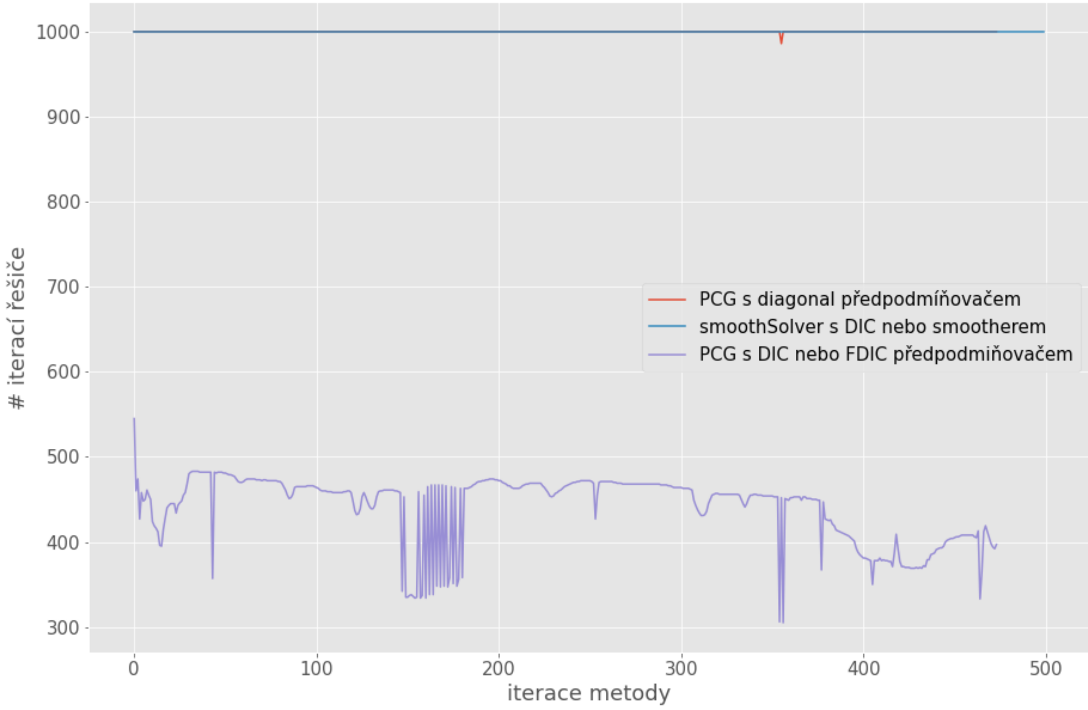
Tabulka 3.5: Nastavení lineárních řešičů pro jednotlivé simulace

Tlak p: **smoothSolver**		Rychlost U: **smoothSolver**				
Předpod.	Smoother	Předpod.	Smoother	Relaxace	# iter	Čas
—	GaussSeidel	—	GaussSeidel	0.9	500+	1011.42
—	GaussSeidel	—	GaussSeidel	0.3	500+	951.71
—	GaussSeidel	—	GaussSeidel	0.95	500+	1086.7
—	GaussSeidel	—	GaussSeidel	0.98	500+	1222.24
FDIC	GaussSeidel	—	GaussSeidel	0.9	500+	1008.47

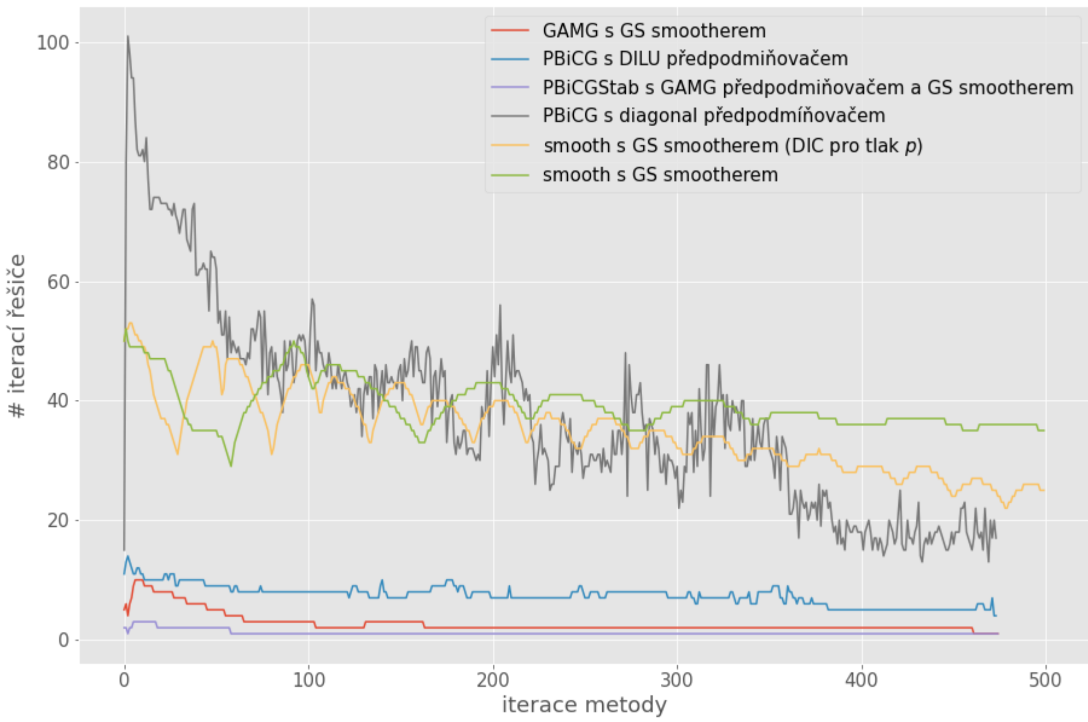
ZAJÍMAVÝ JE TAKÉ VÝVOJ POČTU ITERACÍ JEDNOTLIVÝCH ŘEŠENÍ POMOCÍ LINEÁRNÍHO ŘEŠIČE - CO TO VLASTNĚ JE, JAK JE TO NASTAVENÉ, ATD - ZASE JE Z TOHO VIDĚT ŽE GAMG FUNGUJE NEJLÍP



Obrázek 3.10: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x



Obrázek 3.11: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x



Obrázek 3.12: Konvergence lineárních řešičů smoothSolver v různé konfiguraci pro rychlost U_x

4 Závěr

Nelze dělat obecné závěry, volba řešiče vždy záleží na úloze, nastavených parametrech, tolerancích a třeba i hordwaru, který máme k dispozici. Jediný závěr, který tedy můžeme udělat se týká jen naší úlohy, pro kterou je nejvhodnější řešič XYZ.

Tabulka 4.1: Nastavení lineárních řešičů pro jednotlivé simulace

Řešič pro p		**Řešič pro** U				
GAMG		**GAMG**				
Předpod.	Smoother	Předpod.	Smoother	Relaxace	# iter	Čas
—	DIC	—	DILU	0.9	475	157.94
—	GaussSeidel	—	GaussSeidel	0.9	475	176.99
—	GaussSeidel	—	GaussSeidel	0.3	500+	151.49
—	GaussSeidel	—	GaussSeidel	0.95	285	114.15
—	GaussSeidel	—	GaussSeidel	0.98	500+	257.28
DIC	GaussSeidel	DILU	GaussSeidel	0.98	475	160.59

Tabulka 4.2: Nastavení lineárních řešičů pro jednotlivé simulace

Řešič pro p		**Řešič pro U**				
PCG		**PBiCG**				
Předpod.	Smoother	Předpod.	Smoother	Relaxace	# iter	Čas
DIC	—	DILU	—	0.9	474	540.48
FDIC	—	DILU	—	0.9	474	531.14
GAMG GS smooth.	—	GAMG GS smooth.	—	0.9	475	184.85
diagonal	—	diagonal	—	0.9	474	679.26

Tabulka 4.3: Nastavení lineárních řešičů pro jednotlivé simulace

Řešič pro p		**Řešič pro U**				
smoothSolver		**smoothSolver**				
Předpod.	Smoother	Předpod.	Smoother	Relaxace	# iter	Čas
—	DIC	—	GaussSeidel	0.9	500+	1136.21
—	DIC	—	DILU	0.9	500+	1039.73
—	GaussSeidel	—	GaussSeidel	0.9	500+	1011.42
—	GaussSeidel	—	GaussSeidel	0.3	500+	951.71
—	GaussSeidel	—	GaussSeidel	0.95	500+	1086.7
—	GaussSeidel	—	GaussSeidel	0.98	500+	1222.24
FDIC	GaussSeidel	—	GaussSeidel	0.9	500+	1008.47
—	symGauss-Seidel	—	symGauss-Seidel	0.9	500+	1370.58

A | fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
|  \\    /   O peration     | Version:  v2006
|   \\  /    A nd           | Website:  www.openfoam.com
|    \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
version     2.0;
format      ascii;
class       dictionary;
location    "system";
object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
p
{
solver          smoothSolver;
smoother        GaussSeidel;
tolerance       1e-06;
relTol          0;
}

"(U|k|epsilon|omega|f|v2)"
{
solver          smoothSolver;
smoother        GaussSeidel;
tolerance       1e-06;
relTol          0;
}
```

```
}

SIMPLE
{
nNonOrthogonalCorrectors 0;
consistent      yes;

residualControl
{
p               1e-3;
U               1e-4;
"(k|epsilon|omega|f|v2)" 1e-4;
}
}

relaxationFactors
{
equations
{
U               .95;
".*"            .95;
}
}


// ************************************************************** //
```