[ Prev ] [ Index ] [ Next ]

# Tests de sensiblité du solver PETSC

Created mercredi 15 mars 2017

### Solution départ:

```
# self.ksp.setType('cg')
self.ksp.setType('gmres')
# self.ksp.setType('bicg')
self.ksp.setInitialGuessNonzero(False)
# incomplete Cholesky for preconditionning
# self.ksp.getPC().setType('icc')
# Block Jacobi for preconditionning
# self.ksp.getPC().setType('bjacobi')
# Multi Grid for preconditionning
# self.ksp.getPC().setType('mg')
# no preconditionning
#self.ksp.getPC().setType('none')
# set tolerances
self.ksp.setTolerances(rtol=1e-7)
self.ksp.setTolerances(max_it=1000)
```

### diags rms sur psi_in - psi_inv :

99 itérations
Surface : diff ordre de grandeur = 2.279060e-05
Bottom : diff ordre de grandeur = 3.944816e-05
Lateral: diff ordre de grandeur = 1.585962e-09
Interior: diff ordre de grandeur = 2.584103e-05

### Tests préconditionneur

Multiplier L et RHS par une puissance de dix en surface, lateral, bottom ne change rien si on utilise un préconditionneur. Par contre si on supprime le préconditionneur :

```
self.ksp.getPC().setType('none')
```

alors on a un déséquilibre entre les poids des conditions aux limites et de l'interieur et même si une solution converge, elle est fausse.
Dans le terme RHS: (nx=ny=100, nz=50)

poids de la surface = nx*ny*17 = 1.e5
poids au fond = nx*ny*11 = 1.e5
poids sur les côtés = 2*(nx+ny)*nz*35000 = 7.e8
poids interieur = nx*ny*nz*7.e-4 = 350

On voit que c'est sur les côtés que le poids est le plus fort et c'est la que la

solution est la meilleure.
rms(psi_in - psi_inv)=

Surface: diff ordre de grandeur = 6.821837e-01
Bottom: diff ordre de grandeur = 8.379446e-01
Lateral: diff ordre de grandeur = 1.656146e-12
Interior: diff ordre de grandeur = 6.231530e-01

En conclusion, il faut utiliser un préconditionneur. Le préconditionneur par défaut est Block Jacobi, si on utilise d'autres préconditionneurs, le nombre d'itérations pour converger change légèrement mais la solution diffère très peu.

Préconditionneur MultiGrid
142 itérations
Surface : diff ordre de grandeur = 2.471076e-05
Bottom : diff ordre de grandeur = 4.181846e-05
Lateral: diff ordre de grandeur = 1.879628e-09
Interior: diff ordre de grandeur = 2.762053e-05

## Tests tolérance relative

On étudie le système A*X=B
Default residual norm :

rnorm = B - A*X

Si

rnorm0 = sqrt(sum(B**2))

Le système converge si

rnorm < max(rtol*rnorm0,abstol)

Par défaut:
rtol=1.e-5, abstol=1.e-50, max iterations = 1.e5

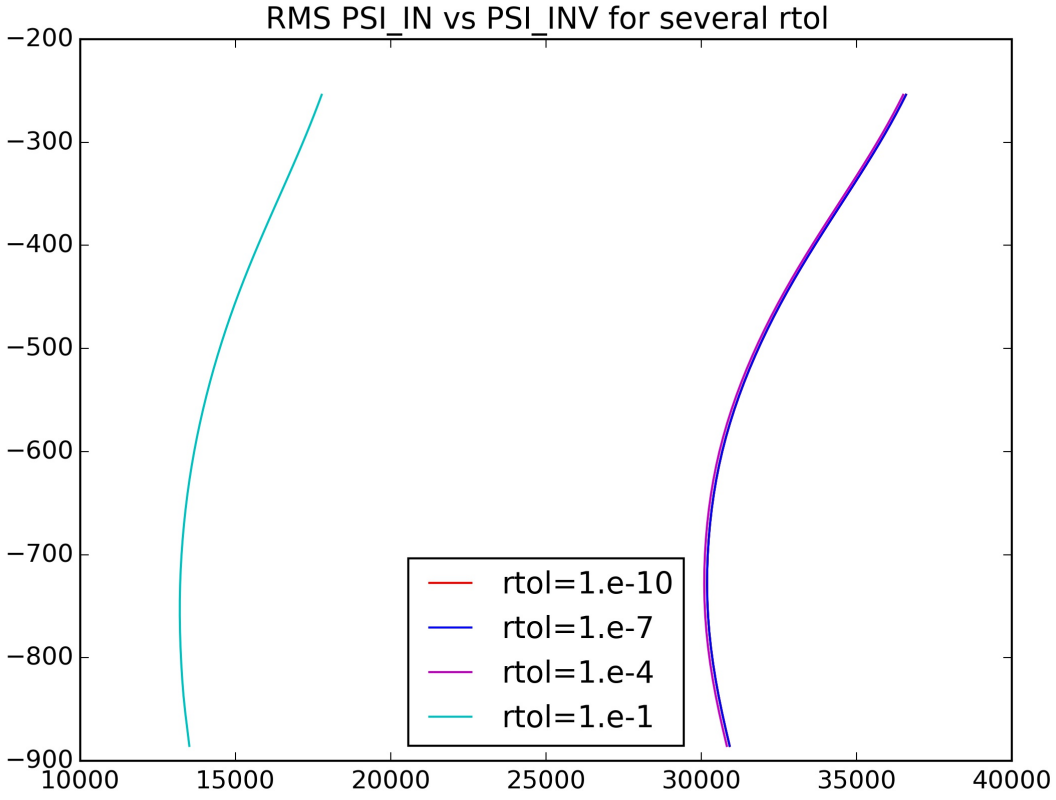
self.ksp.setTolerances(rtol=1e-10)
151 itérations
Surface: diff ordre de grandeur = 2.052609e-05
Bottom: diff ordre de grandeur = 3.618916e-05
Lateral: diff ordre de grandeur = 8.455947e-14
Interior: diff ordre de grandeur = 2.355784e-05

self.ksp.setTolerances(rtol=1e-7)
99 itérations
Surface: diff ordre de grandeur = 2.279060e-05
Bottom: diff ordre de grandeur = 3.944816e-05
Lateral: diff ordre de grandeur = 1.585962e-09
Interior: diff ordre de grandeur = 2.584103e-05

self.ksp.setTolerances(rtol=1e-4)
44 itérations
Surface: diff ordre de grandeur = 2.525800e-03
Bottom: diff ordre de grandeur = 3.556744e-03
Lateral: diff ordre de grandeur = 6.519068e-07
Interior: diff ordre de grandeur = 2.521782e-03

```
        self.ksp.setTolerances(rtol=1e-1)
```

3 itérations
Surface: diff ordre de grandeur = 5.399220e-01
Bottom: diff ordre de grandeur = 7.159000e-01
Lateral: diff ordre de grandeur = 2.749672e-03
Interior: diff ordre de grandeur = 5.240726e-01




On peut donc jouer sur ce paramètre rtol pour diminuer le nombre d'itérations et donc le coût de l'inversion mais attention au résultat.

## Méthode du solveur

La méthode par défaut est GMRES
self.ksp.setType('bicg')
80 iterations
Surface: diff ordre de grandeur = 2.052661e-05
Bottom: diff ordre de grandeur = 3.618942e-05
Lateral: diff ordre de grandeur = 1.164183e-15
Interior: diff ordre de grandeur = 2.355740e-05

## Norme pour la convergence

Si on utilise un psi initial comme Initial guess, on peut changer le calcul de rnorm0:

rnorm0 = sqrt( sum( (B - A*(InitGuess))**2 ) )

Il faut alors lancer le run avec l'option:

-ksp_converged_use_initial_residual_norm