Linear and Non-Linear Preconditioning

Martin J. Gander
martin.gander@unige.ch

University of Geneva

July 2015

Joint work with Victorita Dolean, Walid Kheriji, Felix Kwok and Roland Masson

Jacobi (1845): First Idea of Preconditioning

Als ein Beispiel möge hier die Anwendung der Methode auf die in der Theoria motus p. 219 gegebenen Gleichungen dienen. Die ursprünglichen Gleichungen sind

$$27p + 6q + {*}r - 88 = 0$$
$$6p + 15q + r - 70 = 0$$
$$*p + q + 54r - 107 = 0.$$

Schafft man den Coefficienten 6 bei q in der ersten Gleichung fort, so wird $\alpha = 22^\circ 30'$

$$p = 0{,}92390\,y + 0{,}38268\,y'$$
$$q = 0{,}38268\,y - 0{,}92390\,y'$$

und die neuen Gleichungen werden

$$29{,}4853\,y + {*}\,y' + 0{,}38268\,r - 108{,}0901 = 0$$
$$*\,y + 12{,}5147\,y' - 0{,}92390\,r + 30{,}9967 = 0$$
$$0{,}38268\,y - 0{,}92390\,y' + 54{,}r - 107 = 0$$

After preconditioning, it takes only three Jacobi iterations to obtain three accurate digits!

Classical Stationary Iterative Methods

For a linear system

$$A\mathbf{u} = \mathbf{f},$$

one needs a splitting of $A = M - N$ and then iterates

$$M\mathbf{u}^{n+1} = N\mathbf{u}^n + \mathbf{f}$$

Examples

- Jacobi: $M = \text{diag}(A)$
- Gauss-Seidel: $M = \text{tril}(A)$
- Schwarz domain decomposition: M block diagonal
- Multigrid: M represents a V-cycle or W-cycle

The iterative method

$$\mathbf{u}^{n+1} = M^{-1}N\mathbf{u}^n + M^{-1}\mathbf{f} = (I - M^{-1}A)\mathbf{u}^n + M^{-1}\mathbf{f}$$

1. converges fast if $\rho(I - M^{-1}A)$ is small.
2. and is cheap, if systems with M can easily be solved

Invention of the Conjugate Gradient Method

Stiefel and Rosser 1951: Presentations at a Symposium at the National Bureau of Standards (UCLA)

Hestenes 1951: Iterative methods for solving linear equations

Stiefel 1952: Über einige Methoden der Relaxationsrechnung

Hestenes and Stiefel 1952: Methods of Conjugate Gradients for Solving Linear Systems

"An iterative algorithm is given for solving a system $Ax = k$ of n linear equations in n unknowns. The solution is given in n steps."

Lanczos 1952: Solution of systems of linear equations by minimized iterations (see also 1950)

The Conjugate Gradient Method

To solve approximately $A\mathbf{u} = \mathbf{f}$, A spd, CG finds at step n using the Krylov space

$$\mathcal{K}_n(A, \mathbf{r}^0) := \{\mathbf{r}^0, A\mathbf{r}^0, \ldots, A^{n-1}\mathbf{r}^0\}, \quad \mathbf{r}^0 := \mathbf{f} - A\mathbf{u}^0$$

an approximate solution $\mathbf{u}^n \in \mathbf{u}^0 + \mathcal{K}_n(A, \mathbf{r}^0)$ which satisfies

$$||\mathbf{u} - \mathbf{u}^n||_A \longrightarrow \min .$$

Theorem

With $\kappa(A)$ the condition number of A,

$$||\mathbf{u} - \mathbf{u}^n||_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^n ||\mathbf{u} - \mathbf{u}^0||_A.$$

The conjugate gradient method converges very fast, if the condition number $\kappa(A)$ is not large.

Preconditioning the Linear System

Find a matrix M such that the preconditioned system

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$$

is easier to solve with a Krylov method. Two goals:

1. For CG: make $\kappa(M^{-1}A)$ much smaller than $\kappa(A)$
   More generally: cluster spectrum of $M^{-1}A$ close to one
2. It should be inexpensive to apply M^{-1}

For stationary iterative methods, we needed M such that

1. the spectral radius $\rho(I - M^{-1}A)$ is small
2. it should be inexpensive to apply M^{-1}

Note that

$$\rho(I - M^{-1}A) \text{ small} \iff \text{spectrum of } M^{-1}A \text{ close to one}$$

Idea: design a good M for a stationary iterative method, and then use it as a preconditioner for a Krylov method.

An example of non-linear preconditioning

Additive Schwarz Preconditioned Inexact Newton (ASPIN): Cai, Keyes and Young DD13 (2001), Cai and Keyes SISC (2002)

> *"The nonlinear system is transformed into a new nonlinear system, which has the same solution as the original system. For certain applications the nonlinearities of the new function are more balanced and, as a result, the inexact Newton method converges more rapidly."*

Instead of solving $F(\mathbf{u}) = \mathbf{0}$, solve instead $G(F(\mathbf{u})) = \mathbf{0}$ with

- If $G(\mathbf{v}) = \mathbf{0}$ then $\mathbf{v} = \mathbf{0}$
- $G \approx F^{-1}$ in some sense
- $G(F(\mathbf{v}))$ is easy to compute
- Applying Newton, $(G(F(\mathbf{v})))'\mathbf{w}$ should also be easy to compute

An example of non-linear preconditioning

Additive Schwarz Preconditioned Inexact Newton (ASPIN): Cai, Keyes and Young DD13 (2001), Cai and Keyes SISC (2002)

> *"The nonlinear system is transformed into a new nonlinear system, which has the same solution as the original system. For certain applications the nonlinearities of the new function are more balanced and, as a result, the inexact Newton method converges more rapidly."*

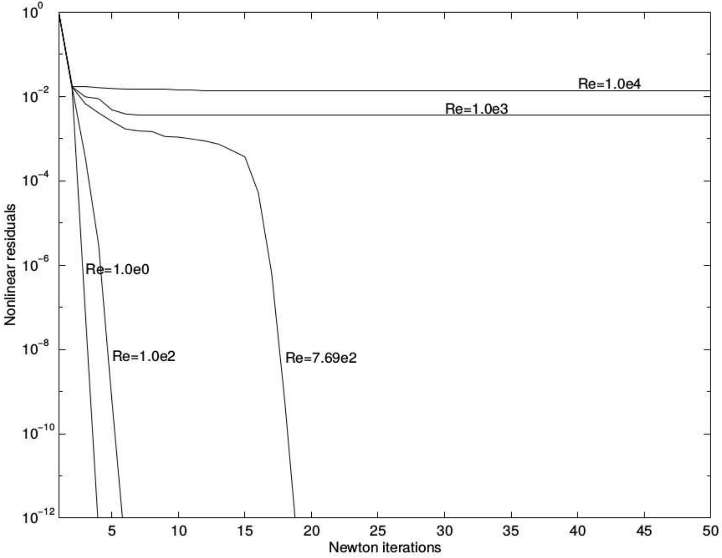
Instead of solving $F(\mathbf{u}) = \mathbf{0}$, solve instead $G(F(\mathbf{u})) = \mathbf{0}$ with

- If $G(\mathbf{v}) = \mathbf{0}$ then $\mathbf{v} = \mathbf{0}$
- $G \approx F^{-1}$ in some sense
- $G(F(\mathbf{v}))$ is easy to compute
- Applying Newton, $(G(F(\mathbf{v})))'\mathbf{w}$ should also be easy to compute

ASPIN: $\mathcal{F}_2(u) = R_0^T C_0^A(u) + \sum_{i=1}^{I} R_i^T G_i(u) - u = 0$,
$F_0(C_0^A(u) + u_0^*) = -R_0 F(u)$, $F_0(u_0^*) = 0$

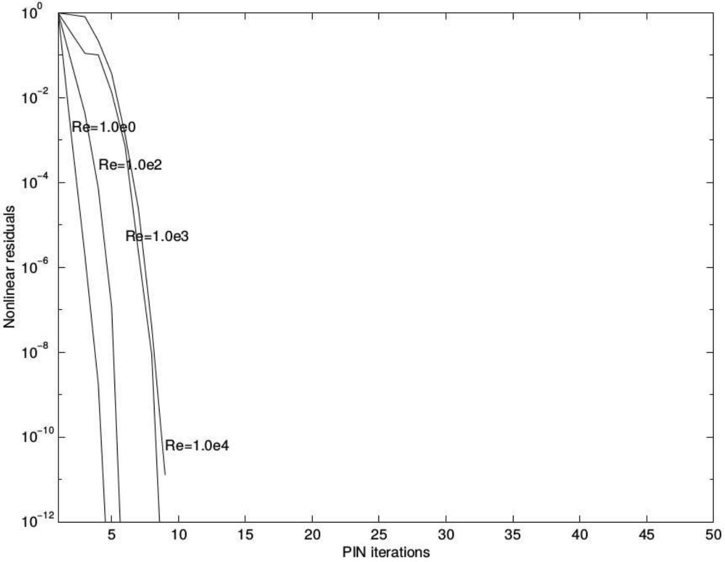
Example: Newton without preconditioning

Driven cavity flow problem (Cai, Keyes 2002)



Example: Newton with ASPIN preconditioning

Driven cavity flow problem (Cai, Keyes 2002)



Nonlinear Preconditioners: Systematic Construction

Recall from the linear case: from a stationary iterative method for $A\mathbf{u} = \mathbf{f}$,

$$\mathbf{u}^{n+1} = (I - M^{-1}A)\mathbf{u}^n + M^{-1}\mathbf{f}$$

with fast convergence, i.e. $\rho(I - M^{-1}A)$ small, we obtain a good preconditioner M to solve

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$$

with a Krylov method.

Idea: For the non-linear problem $F(\mathbf{u}) = 0$, construct a fixed point iteration

$$\mathbf{u}^{n+1} = \mathcal{G}(\mathbf{u}^n)$$

and then solve

$$\mathcal{F}(\mathbf{u}) := \mathcal{G}(\mathbf{u}) - \mathbf{u} = 0$$

using Newton's method.

Using a Nonlinear Schwarz Method

One dimensional non-linear model problem

$$\begin{aligned} \mathcal{L}(u) := -\partial_x((1+u^2)\partial_x u) &= f, \quad \text{in } \Omega := (0,L), \\ u(0) &= 0, \\ u(L) &= 0, \end{aligned}$$

Parallel Schwarz method with two subdomains $\Omega_1 := (0,\beta)$ and $\Omega_2 := (\alpha, L)$, $\alpha < \beta$

$$\begin{aligned} \mathcal{L}(u_1^n) &= f, && \text{in } \Omega_1 := (0,\beta), \\ u_1^n(0) &= 0, \\ u_1^n(\beta) &= u_2^{n-1}(\beta), \\ \mathcal{L}(u_2^n) &= f, && \text{in } \Omega_2 := (\alpha,L), \\ u_2^n(\alpha) &= u_1^{n-1}(\alpha), \\ u_2^n(L) &= 0. \end{aligned}$$

This is a non-linear fixed point iteration.

How can we apply Newton to solve at the fixed point?

Defining the Fixed Point Equation

$$u^n(x) := \begin{cases} u_1^n(x) & \text{if } 0 \leq x < \frac{\alpha+\beta}{2}, \\ u_2^n(x) & \text{if } \frac{\alpha+\beta}{2} \leq x \leq L, \end{cases}$$

or with zero extension operators $\tilde{R}_i^T$

$$u^n = \tilde{R}_1^T u_1^n + \tilde{P}_2^T u_2^n.$$

With the solution operators for the non-linear subdomain problems

$$u_1^n = G_1(u^{n-1}), \qquad u_2^n = G_2(u^{n-1}),$$

we obtain (for I subdomains)

$$u^n = \sum_{i=1}^{I} \tilde{R}_i^T G_i(u^{n-1}) =: \mathcal{G}_1(u^{n-1}).$$

RASPEN: Solve the fixed point equation with Newton

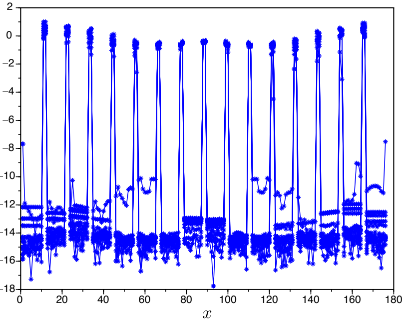
$$\tilde{\mathcal{F}}_1(u) := \mathcal{G}_1(u) - u = \sum_{i=1}^{I} \tilde{R}_i^T G_i(u) - u = 0$$

Example: Forchheimer Equation

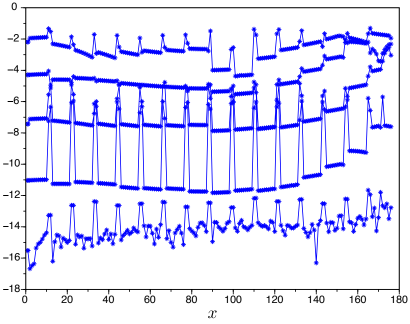
$$\begin{cases} (q(-\lambda(x)u(x)'))' &= f(x) \quad \text{in } \Omega, \\ u(0) &= u_0^D, \\ u(L) &= u_L^D. \end{cases}$$

$$\beta > 0,\ 0 < \lambda_{min} \leq \lambda(x) \leq \lambda_{max},\ q(g) = \operatorname{sgn}(g)\frac{-1+\sqrt{1+4\beta|g|}}{2\beta}$$

Residual as a function of iterations for 8 subdomains



Nonlinear Schwarz



RASPEN

Adding a Coarse Grid Correction

Use the Full Approximation Scheme (FAS) from multigrid: compute correction c from a non-linear coarse problem

$$\mathcal{L}^c(R_0 u^{n-1} + c) = \mathcal{L}^c(R_0 u^{n-1}) + R_0(f - \mathcal{L}(u^{n-1})),$$

Add the correction $c := C_0(u^{n-1})$ to the iterate

$$u_{new}^{n-1} = u^{n-1} + R_0^T C_0(u^{n-1}),$$

This gives naturally the two level fixed point iteration

$$u^n = \sum_{i=1}^{I} \tilde{R}_i^T G_i(u^{n-1} + R_0^T C_0(u^{n-1})) =: \mathcal{G}_2(u^{n-1}),$$

Two level RASPEN means solving with Newton

$$\tilde{\mathcal{F}}_2(u) := \mathcal{G}_2(u) - u = \sum_{i=1}^{I} \tilde{R}_i^T G_i(u + R_0^T C_0(u)) - u = 0.$$

Comparison of ASPIN and RASPEN

One Level Variants:

$$\text{RASPEN} : \tilde{\mathcal{F}}_1(u) := \sum_{i=1}^{I} \tilde{R}_i^T G_i(u) - u = 0$$

$$\text{ASPIN} : \mathcal{F}_1(u) := \sum_{i=1}^{I} R_i^T G_i(u) - u = 0$$

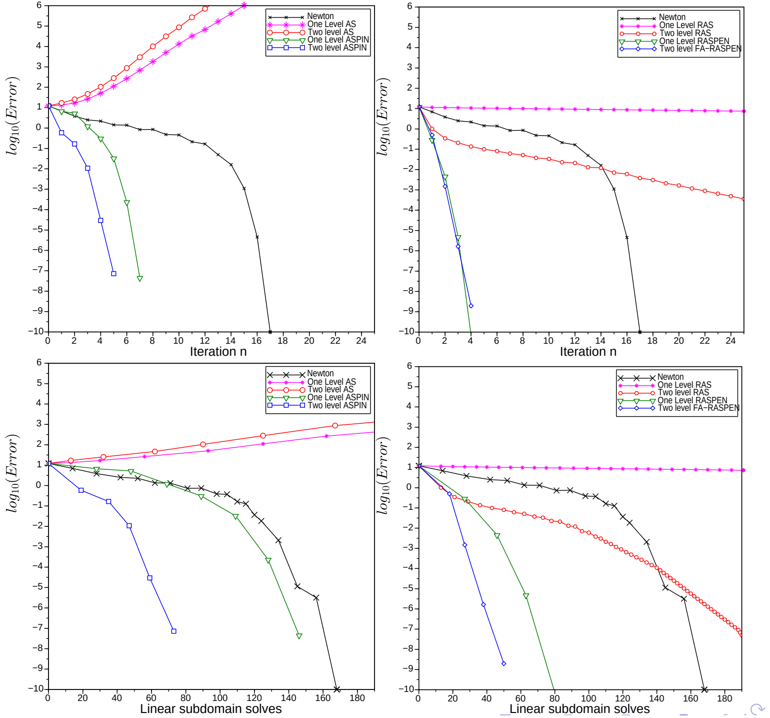
Two Level Variants:

$$\text{RASPEN} : \tilde{\mathcal{F}}_2(u) := \sum_{i=1}^{I} \tilde{R}_i^T G_i(u + R_0^T C_0(u)) - u = 0$$

$$\text{ASPIN} : \mathcal{F}_2(u) = R_0^T C_0^A(u) + \sum_{i=1}^{I} R_i^T G_i(u) - u = 0, \quad F_0(C_0^A(u) + u_0^*) = -R_0 F(u),\ F_0(u_0^*) = 0$$

Three main differences:

1. RASPEN does not sum corrections in the overlap like ASPIN, since it is a convergent fixed point method
2. RASPEN uses the full approximation scheme for the coarse correction, whereas ASPIN does an additive ad hoc construction
3. RASPEN uses exact Newton, since one obtains the exact Jacobian from the inner non-linear solves.



Newton
One Level AS
Two level AS
One Level ASPIN
Two level ASPIN
One Level RAS
Two level RAS
One Level RASPEN
Two level FA−RASPEN
$log_{10}(Error)$
Iteration n
Linear subdomain solves

A Non-Linear Diffusion Problem: One Level

$$\begin{cases} -\nabla \cdot ((1+u^2)\nabla u) = f, & \Omega = [0,1]^2, \\ u = 1, & x = 1, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{otherwise.} \end{cases}$$

Results for RASPEN (ASPIN in parentheses)

$N \times N$	n	ls_n^G	ls_n^{in}	$ls_n^{\min}$	LS_n 1 level
2×2	1	15(20)	3(3)	2(2)	57(75)
	2	17(23)	2(2)	2(2)	
	3	18(26)	1(1)	1(1)	
4×4	1	32(37)	2(2)	2(2)	110(129)
	2	35(41)	2(2)	1(1)	
	3	38(46)	1(1)	1(1)	
6×6	1	46(54)	2(2)	2(2)	164(183)
	2	51(59)	2(2)	1(1)	
	3	57(65)	1(1)	1(1)	

ls_n^G: GMRES steps for Jacobian, ls_n^{in} maximium and $ls_n^{\min}$ minimum iterations to evaluate $\mathcal{F}$

A Non-Linear Diffusion Problem: Two Level

$$\begin{cases} -\nabla\cdot((1+u^2)\nabla u) = f, & \Omega=[0,1]^2, \\ u = 1, & x=1, \\ \dfrac{\partial u}{\partial \mathbf{n}} = 0, & \text{otherwise.} \end{cases}$$

Results for Two Level RASPEN (Two Level ASPIN)

$N\times N$	n	ls_n^G	ls_n^{in}	$ls_n^{\min}$	LS_n 2 level
2×2	1	13(23)	3(3)	2(2)	51(83)
	2	15(26)	2(2)	2(2)	
	3	17(28)	1(1)	1(1)	
4×4	1	18(33)	2(2)	2(2)	71(123)
	2	22(39)	2(2)	1(1)	
	3	26(46)	1(1)	1(1)	
6×6	1	18(35)	2(2)	2(2)	73(133)
	2	23(42)	2(2)	1(1)	
	3	27(51)	1(1)	1(1)	

ls_n^G: GMRES steps for Jacobian, ls_n^{in} maximium and $ls_n^{\min}$ minimum iterations to evaluate $\mathcal{F}$

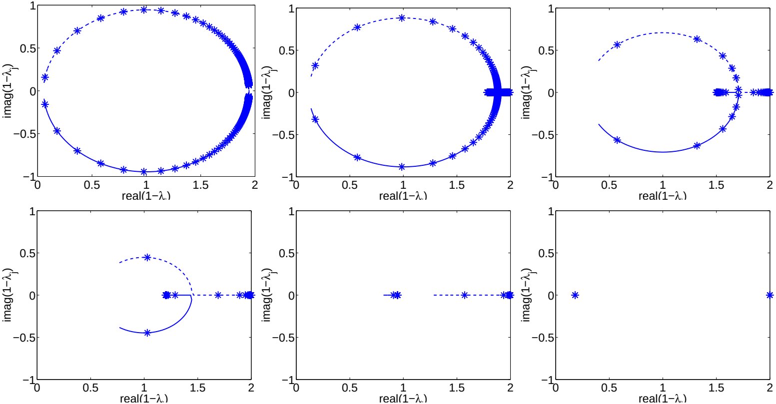
Summary

- Linear preconditioning means solve the preconditioned linear system $M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$ using a Krylov method.
- Non-linear preconditioning means solve the preconditioned non-linear system $G(F(\mathbf{u})) = 0$ using Newton's method.
- It is easy to define non-linear (and linear!) preconditioners from a fixed point iteration (e.g. Haeberlein, Halpern, Anthony, DD20, talk by Axel Klawonn)

Summary

- Linear preconditioning means solve the preconditioned linear system $M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$ using a Krylov method.
- Non-linear preconditioning means solve the preconditioned non-linear system $G(F(\mathbf{u})) = 0$ using Newton's method.
- It is easy to define non-linear (and linear!) preconditioners from a fixed point iteration (e.g. Haeberlein, Halpern, Anthony, DD20, talk by Axel Klawonn)
- But what should a linear or non-linear preconditioner really do ? ($\Longrightarrow$ talk by Andy Wathen)

Example of the HSS Preconditioner



Optimization of the Hermitian and Skew-Hermitian Splitting Iteration for Saddle-Point Problems, Benzi, G., Golub, BIT Vol. 43, 2003.