Variations on Richardson's method and acceleration

Claude Brezinski

Dedicated to Jean Meinguet on the occasion of his 65th birthday

Abstract

The aim of this paper is to present an acceleration procedure based on projection and preconditioning for iterative methods for solving systems of linear equations. A cycling strategy leads to a new iterative method. These procedures are closely related to Richardson's method and acceleration. Numerical examples illustrate the purpose.

For solving the system of linear equations $Ax = b$, we consider any convergent method which produces the sequence of iterates (x_n). Quite often the convergence is too slow and it has to be accelerated. There exist many processes for that purpose whose references can be easily found. They are either quite general extrapolation methods as described, for example, in [5] or particular ones as those studied in [2]. The aim of this paper is to present a new acceleration procedure based on projection and preconditioning. Cycling with this procedure will lead to a new iterative method.

1 The procedure

Let us consider the sequence (y_n) given by

$$y_n = x_n - \lambda_n z_n \tag{1}$$

where z_n is an (almost) arbitrary vector called the *search direction* or the *direction of descent* and λ_n a parameter called the *stepsize*. Setting $r_n = b - Ax_n$ and $\rho_n = b - Ay_n$, we have

$$\rho_n = r_n + \lambda_n A z_n. \tag{2}$$

1991 *Mathematics Subject Classification* : 65F10, 65F25, 65B05, 65D15.
Key words and phrases : linear systems, acceleration, extrapolation, preconditioning.

We shall now choose for λ_n the value which minimizes $\|\rho_n\| = (\rho_n, \rho_n)^{1/2}$, that is [20]

$$\lambda_n = -\frac{(Az_n, r_n)}{(Az_n, Az_n)}. \tag{3}$$

The minimum value of (ρ_n, ρ_n) is then given by

$$(\rho_n, \rho_n) = (r_n, r_n) - \frac{(Az_n, r_n)^2}{(Az_n, Az_n)} = (r_n, r_n) \sin^2 \theta_n \tag{4}$$

where θ_n is the angle between the vectors r_n and Az_n. Thus, obviously, $\|\rho_n\| \leq \|r_n\|$. Replacing λ_n by its value, we have

$$\rho_n = (I - \mathrm{P}_n) r_n$$

with

$$\mathrm{P}_n = \frac{Az_n (Az_n)^T}{(Az_n, Az_n)}.$$

It is easy to see that P_n represents an orthogonal projection ($\mathrm{P}_n^2 = \mathrm{P}_n$ and $\mathrm{P}_n^T = \mathrm{P}_n$) and so $I - \mathrm{P}_n$ also. Indeed we have $(Az_n, \rho_n) = 0$ and it follows

$$(\rho_n, \rho_n) = (r_n, r_n) - (r_n, \mathrm{P}_n r_n).$$

If $z_n = A^T u_n$, where u_n is an arbitrary vector, the usual projection methods are recovered (see [15, p. 163ff] for example). Such methods are discussed in details in [3]; see also [4].

It must be noticed that formula (2) allows to obtain ρ_n at no extra cost.

2 Choice of the search direction

Let us now see how to choose the vector z_n. From (4) we see that $\rho_n = 0$ if and only if the vectors r_n and Az_n are colinear, that is if $z_n = \alpha A^{-1} r_n$. But ρ_n does not change if z_n is replaced by αz_n and, thus, we can take $\alpha = 1$. Obviously, this choice of z_n cannot be made in practice and, thus, we shall assume that an approximation (in a sense to be defined below) C_n of A^{-1} is known and we shall take

$$z_n = C_n r_n.$$

Thus, the procedure (1)–(2) becomes

$$\begin{aligned} y_n &= x_n + \frac{(AC_n r_n, r_n)}{(AC_n r_n, AC_n r_n)} C_n r_n, \\ \rho_n &= \left[I - \frac{(AC_n r_n, r_n)}{(AC_n r_n, AC_n r_n)} AC_n \right] r_n. \end{aligned} \tag{5}$$

So, this procedure appears as a combination of projection and right preconditioning. It is a generalization of Richardson's acceleration which is recovered for the choice $C_n = I$. Let us study its properties.

We first define the vectors q_n by

$$q_n = r_n - Az_n.$$

Since the value of λ_n minimizes $\|\rho_n\|$, we have

$$\|\rho_n\| \le \|r_n - Az_n\| = \|q_n\|.$$

Let $R_n = I - AC_n$. Then, $q_n = R_n r_n$, and $\|q_n\| \le \|R_n\| \cdot \|r_n\|$. We finally have

$$\frac{\|\rho_n\|}{\|r_n\|} \le \|R_n\|$$

and thus we proved the

Theorem 1
If $\exists K < 1$ *such that* $\forall n, \|R_n\| \le K,$ *then* $\forall n, \|\rho_n\| \le K\|r_n\|$.
If $\lim_{n\to\infty} R_n = 0,$ *then* $\lim_{n\to\infty} \|\rho_n\|/\|r_n\| = 0$.

This theorem shows that, in order to accelerate the convergence of the initial iterative method, one has to be able to construct a sequence of variable preconditioners C_n so that $R_n = I - AC_n$ tends to zero when n goes to infinity. Obviously, this can never be achieved by a constant preconditioner $C_n = C_0, \forall n$. This is the case, in particular, if iterations for obtaining a good preconditioner are made before starting (1)–(2). However, if K is sufficiently small, the residual vectors will be greatly reduced. The procedure (5) will be called PR2 *acceleration* where the letters PR first stand for projection and then for preconditioning. In fact, this PR2 acceleration is identical to the application of the hybrid procedure of rank 2 to the vectors r_n and q_n [6].

If we use a restarting (also called cycling) strategy with our procedure (1)–(2), we obtain the following iterative method

$$\begin{aligned} x_{n+1} &= x_n + \frac{(AC_n r_n, r_n)}{(AC_n r_n, AC_n r_n)}\, C_n r_n, \\ r_{n+1} &= \left[I - \frac{(AC_n r_n, r_n)}{(AC_n r_n, AC_n r_n)} AC_n \right] r_n. \end{aligned} \tag{6}$$

This method will be called the PR2 *iterative method.* It is a generalization of Richardson's method which is recovered if $\forall n, C_n = I$. The PR2 ietrative method is quite close to the EN method introduced in [12] and its variants discussed in [21]. Let us also mention that iterative methods of the form (1) but with λ_n not necessarily chosen by (3) have been widely studied. In our case, we immediately obtain, from the preceding theorem

Theorem 2
If $\exists K < 1$ *such that* $\forall n, \|R_n\| \le K,$ *then* $\|r_n\| = \mathcal{O}(K^n)$.
If $\lim_{n\to\infty} R_n = 0,$ *then* $\|r_{n+1}\| = o(\|r_n\|)$.

This theorem shows that a constant preconditioner is enough to ensure the convergence of the PR2 method. In that case, the convergence is linear and the speed depends on the value of $K = \|R_0\|$. The choice $z_n = A^T r_n$ corresponds, in fact, to a projection method (see [2]) and to $C_n = A^T$. Thus, it is a good choice if the matrix A is almost orthogonal. If R_n tends to zero then a superlinear convergence is obtained.

Other results on the PR2 acceleration and on the PR2 iterative method can be found in [4].

3 Choice of the preconditioner

So, we are now faced to the problem of finding an efficient and cheap method for computing the sequence (C_n) of preconditioners. By efficient, we mean that the convergence of C_n to A^{-1} must be as fast as possible, and, by cheap, we mean that it must require as few arithmetical operations and storage as possible. General considerations about the qualities of approximate inverses can be found in [9]. However, the literature does not seem to be very rich on this topics. Some procedures, such as rank–one modifications with a finite termination and iterative methods, can be found in the book of Durand [11, vol. 2, pp. 144ff] but they have not attracted much attention and they still remain to be studied from the theoretical point of view. One can also think of using the Broyden's updates [7], or those of Huang [19], or those used in the ABS projection algorithms [1]. As proved in [16, 17], although Broyden's methods converge in a finite number of iterations for linear systems, the updates do not always converge to A^{-1}. For reviews of update methods, see [23] and [22]; on preconditioning techniques, see [8, 13, 14]. We shall now explore another possibility.

Let us consider the two following sequences of variable preconditioners

$$C_{n+1} = C_n U_n + D_n \tag{7}$$

and

$$C'_{n+1} = U'_n C'_n + D_n \tag{8}$$

where (D_n) is an arbitrary sequence of matrices, $U_n = I - AD_n$ and $U'_n = I - D_n A$. We have

$$\begin{aligned} R_{n+1} &= R_n U_n \\ R'_{n+1} &= U'_n R'_n \end{aligned}$$

with $R_n = I - AC_n$ and $R'_n = I - C'_n A$. Thus $\|R_{n+1}\| \le \|U_n\| \cdot \|R_n\|$ and $\|R'_{n+1}\| \le \|U'_n\| \cdot \|R'_n\|$ and it immediately follows

Theorem 3

Let (C_n) be constructed by (7).

1. *If $\|U_n\| = \mathcal{O}(1)$, then $\|R_{n+1}\| = \mathcal{O}(\|R_n\|)$.*

2. *If $\|U_n\| = o(1)$, then $\|R_{n+1}\| = o(\|R_n\|)$.*

3. *If $\|U_n\| = \mathcal{O}(\|R_n\|)$, then $\|R_{n+1}\| = \mathcal{O}(\|R_n\|^2)$.*

4. If $\|U_n\| = o(\|R_n\|)$*, then* $\|R_{n+1}\| = o(\|R_n\|^2)$*.*

Similar results hold for the sequence (C'_n) *constructed by (8).*

Let us remark that, in the case 1, $\|R_n\| \le K^n \|R_0\|$ and thus (C_n) tends to A^{-1} if $K < 1$.

We shall now consider several choices for the sequence (D_n)

3.1 Constant preconditioner

If $\forall n, D_n = 0$, then $\forall n, U_n = U'_n = I$ and it follows that $\forall n, C_n = C_0$ and $C'_n = C'_0$. So, we are in the case 1 of theorem 3 and $\forall n, R_n = R_0$ and $R'_n = R'_0$. If the matrix A is strictly diagonally dominant and if C_0 is the inverse of the diagonal part of A, then $\|R_0\| < 1$ for the l_1 and the l_∞ norms.

3.2 Linear iterative preconditioner

If $\forall n, D_n = D_0$, then $\forall n, U_n = U_0$ and $U'_n = U'_0$. Thus, we are again in the case 1 of theorem 3 and it follows that $\|R_n\| \le \|U_0\|^n \|R_0\|$. So, if $\|U_0\| < 1$, we have $\|R_{n+1}\| \le \|U_0\| \cdot \|R_n\|$ and $\|R_n\| = o(1)$ and similar results for (C'_n). Both sequences of preconditioners converge linearly to A^{-1}. This is, in particular, the case if $D_0 = C_0$ with $\|R_0\| < 1$.

Let us consider another procedure which is a generalization of a method due to Durand [11, vol. 2, p. 150], or a modification of the procedure given in [10]. Starting from the splitting $A = M - N$ and replacing A by its expression in $AA^{-1} = I$ leads to

$$A^{-1} = \left(M^{-1}N\right) A^{-1} + M^{-1}$$

and to the iterative procedure

$$C_{n+1} = \left(M^{-1}N\right) C_n + M^{-1} \tag{9}$$

with C_0 arbitrary. Since we have

$$C_{n+1} - A^{-1} = M^{-1}N \left(C_n - A^{-1}\right)$$

we immediately obtain the

Theorem 4
The sequence (C_n) *constructed by (9) with* C_0 *arbitrary, converges to* A^{-1} *if and only if* $\rho\,(M^{-1}N) < 1$*. In that case*

$$\|C_n - A^{-1}\| = \mathcal{O}\left(\rho^n(M^{-1}N)\right).$$

So, if the sequence (x_n) is obtained by $x_{n+1} = M^{-1}Nx_n + c$ with $c = M^{-1}b$, then the sequence (C_n) can be obtained by the same first order stationary iterative process and it is easy to see that $x_n = C_n b$. It follows that the convergence behavior of the sequences (x_n) and (C_n) is the same. Moreover, we have $R_{n+1} = NM^{-1}R_n$.

Another procedure consists of replacing A by its expression in $A^{-1}A = I$. Thus it follows

$$A^{-1} = A^{-1}\left(NM^{-1}\right) + M^{-1}$$

which leads to the iterative procedure

$$C_{n+1} = C_n\left(NM^{-1}\right) + M^{-1} \tag{10}$$

with C_0 arbitrary. Since we have

$$C_{n+1} - A^{-1} = \left(C_n - A^{-1}\right)NM^{-1}$$

we immediately obtain the

Theorem 5
The sequence (C_n) constructed by (10) with C_0 arbitrary, converges to A^{-1} if and only if $\rho\left(NM^{-1}\right) < 1$. In that case

$$\|C_n - A^{-1}\| = \mathcal{O}\left(\rho^n(NM^{-1})\right).$$

It is easy to see that $R_{n+1} = R_n\left(NM^{-1}\right)$. So, if the sequence (x_n) is obtained by $x_{n+1} = M^{-1}Nx_n + c$ with $c = M^{-1}b$, then $r_{n+1} = \left(NM^{-1}\right)r_n$ which shows that the behavior of the sequences (r_n) and (R_n) is the same. Moreover, we have $R_{n+1} = R_nM^{-1}N$ where now $R_n = I - C_nA$.

3.3 Quadratic iterative preconditioner

For obtaining a sequence (R_n) converging faster to zero, we shall make use of the method of iterative refinement. Assuming that $\|R_0\| < 1$, we construct the sequence (C_n) by

$$C_{n+1} = C_n(I + R_n) \qquad R_{n+1} = I - AC_{n+1}. \tag{11}$$

It is easy to see that this method corresponds to the choice $D_n = C_n$ and that $R_{n+1} = R_n^2$. Thus

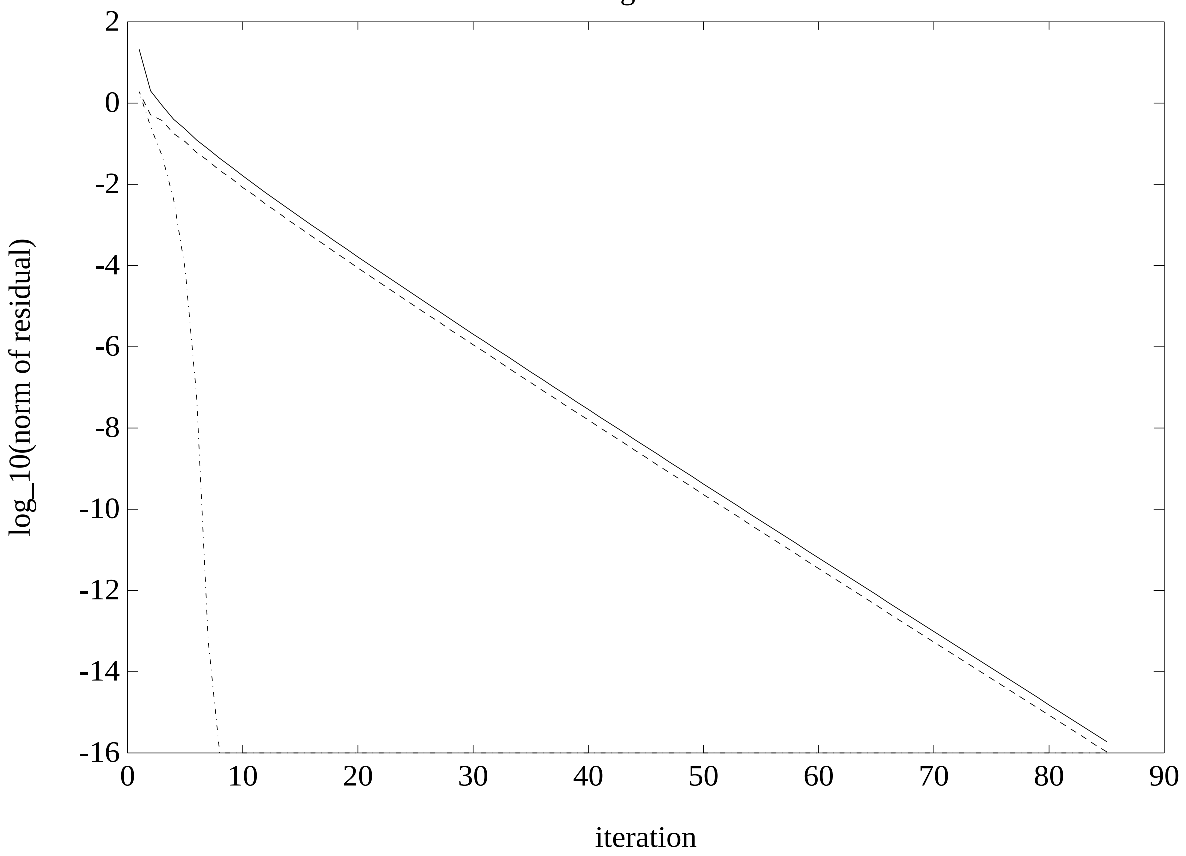
$$\|C_n - A^{-1}\| \le \frac{\|R_0\|^{2^n}}{1 - \|R_0\|}\|C_0\|$$

and thus the convergence of (C_n) to A^{-1} is quadratic. It follows that $\|\rho_n\| = \mathcal{O}\left(\|R_0\|^{2^n}\|r_n\|\right)$ which shows that the convergence of the PR2 acceleration is extremely fast and that (ρ_n) can converge even if (r_n) does not tend to zero. A similar result holds for the PR2 iterative method.

4 Numerical examples

Let us illustrate the preceding procedures. We consider the system of dimension $p = 50$ whose matrix is given by $a_{ii} = 3, a_{i+1,i} = 1, a_{i,i+1} = -1$ and $a_{1p} = 2$. The vector b is then computed so that $\forall i, x_i = 1$. We take for C_0 the inverse of the diagonal of A. The sequence (r_n) is obtained by the method of Jacobi with $x_0 = 0$. In Figure 1, the highest curve refers to the method of Jacobi, the curve very close

Figure 1



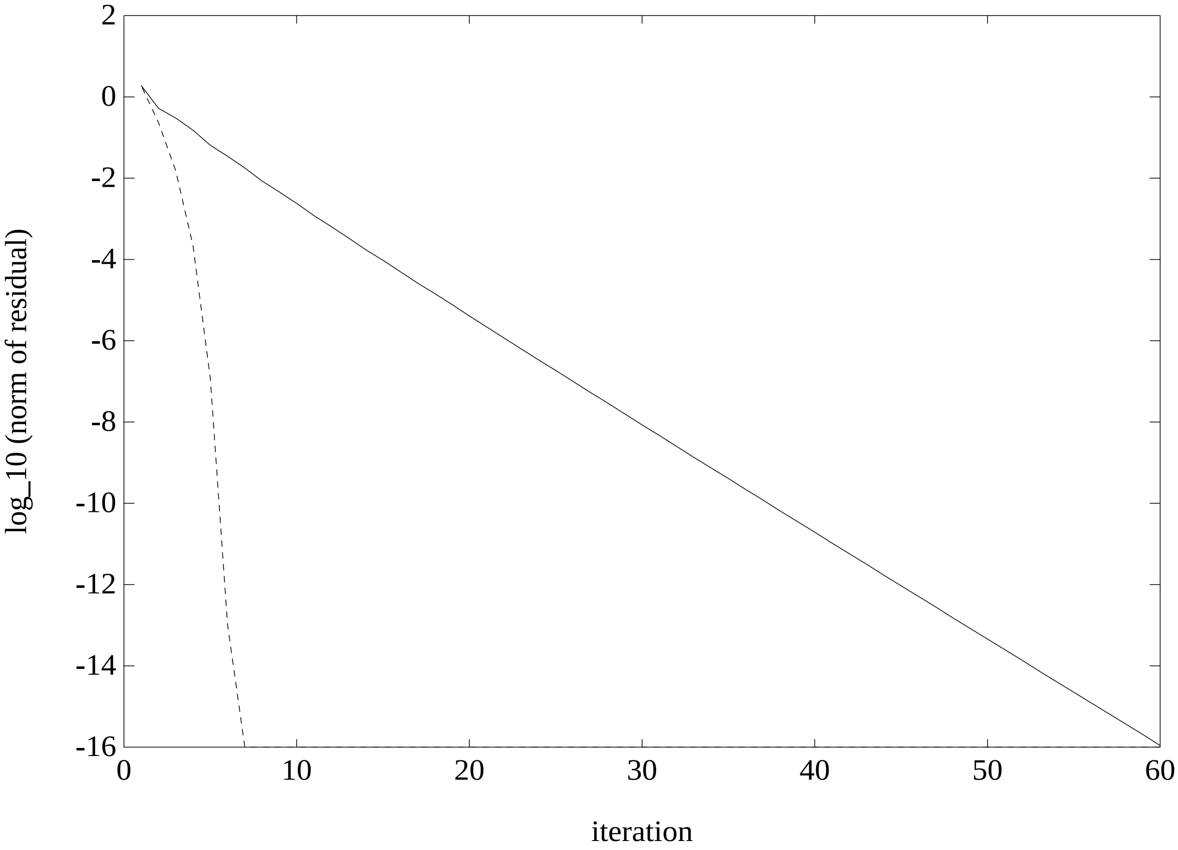
to it is obtained by the PR2 acceleration with $\forall n, C_n = C_0$, while the lowest one corresponds to the PR2 acceleration with the sequence (C_n) given by the formulae (11).

In Figure 2, the results given by the PR2 iterative method for the same system are displayed. Let us mention that, in both cases, the results obtained for the dimension $p = 300$ are almost the same.

For the same example and when (C_n) is constructed by (10), we obtain the results displayed in Figure 3 for the PR2 acceleration of the method of Jacobi, and the results of Figure 4 for the PR2 iterative method.

Usually the procedures described above (and, in particular, the quadratic iterative preconditioner) are not easily feasible for large values of p (the dimension of the system) because of the almost immediate fill–in of the matrices C_n (even if A is sparse) and the number of matrix–by–matrix multiplications. So, our numerical examples are only given to illustrate the preceding theorems. However, if $D_n = a_n b_n^T$ where a_n and b_n are arbitrary vectors, then the fill–in of the matrices C_n can be more easily controlled. In particular, the sequence (C_n) can be constructed by $C_{n+1} = C_n + a_n b_n^T$ where the vectors a_n and b_n are chosen to control the sparsity of C_{n+1} and so that C_{n+1}^{-1} be a good approximation of A. Usually, in update methods, such as those mentioned above, approximations of A are constructed by rank–one modifications and then inverted by the Sherman–Morrison formula [18]. In that case, it is much more difficult to control the sparsity of C_{n+1}. Here, the reverse strategy is adopted since the approximations C_n of A^{-1} are directly computed.

Figure 2



However, in general, it is impossible simultaneously to control the sparsity and to have convergence of (C_n) to A^{-1}.

5 Another choice for the search direction

In the formulae (1), (3), (5) and (6), the products AC_nr_n and C_nr_n are needed. For obvious considerations, the separate computation of the matrices AC_n is not recommended. As far as possible, the storage of C_n has also to be avoided. On the other hand, the recursive computation of the vectors C_nr_n is not so easy since both C_n and r_n depend on n. This is why another choice of the vectors z_n, avoiding these drawbacks, will now be proposed.

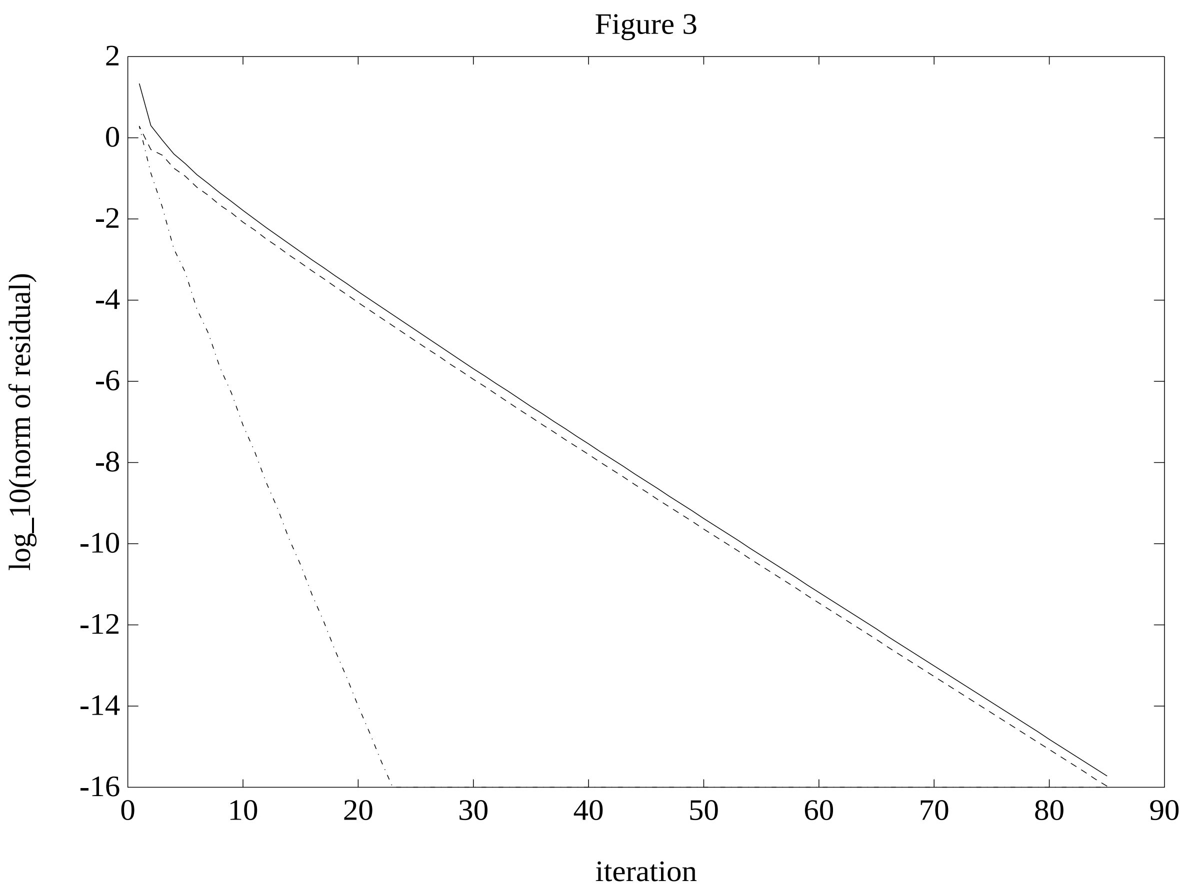
In Section 2, we saw that the best choice for z_n is a vector colinear to $A^{-1}r_n$. But $A^{-1}r_n = A^{-1}b - x_n$. Thus, we shall take

$$z_n = C_nb - x_n$$

where C_n is again an approximation of A^{-1}. So, now, we have to compute recursively matrix–by–vector products where the vector no longer depends on the iteration n.

With the quadratic iterative preconditioner discussed in Subsection 3.3, the products C_nb cannot be computed recursively. So, we shall now consider the case of a linear iterative preconditioner as given in Subsection 3.2 by

$$C_{n+1} = U_nC_n + D_n.$$



Setting

$$v_n = C_n b \quad \text{and} \quad w_n = D_n b$$

we immediately obtain

$$v_{n+1} = U_n v_n + w_n$$

with $u_0 = C_0 b$ and $w_0 = D_n b$. We also have $z_n = v_n - x_n$.

If, as in (9), $\forall n, U_n = M^{-1}N$ and $D_n = M^{-1}$ then $\forall n, w_n = w_0$ and the products $U_n v_n$ are easily computed.

Let us remark that, if $(C_n b)$ tends to a limit different from x (as in the case of a constant preconditioner), (λ_n) given by (3) nevertheless tends to zero and, so, (y_n) tends to x.

Numerical examples with this choice of the search direction still have to be performed.

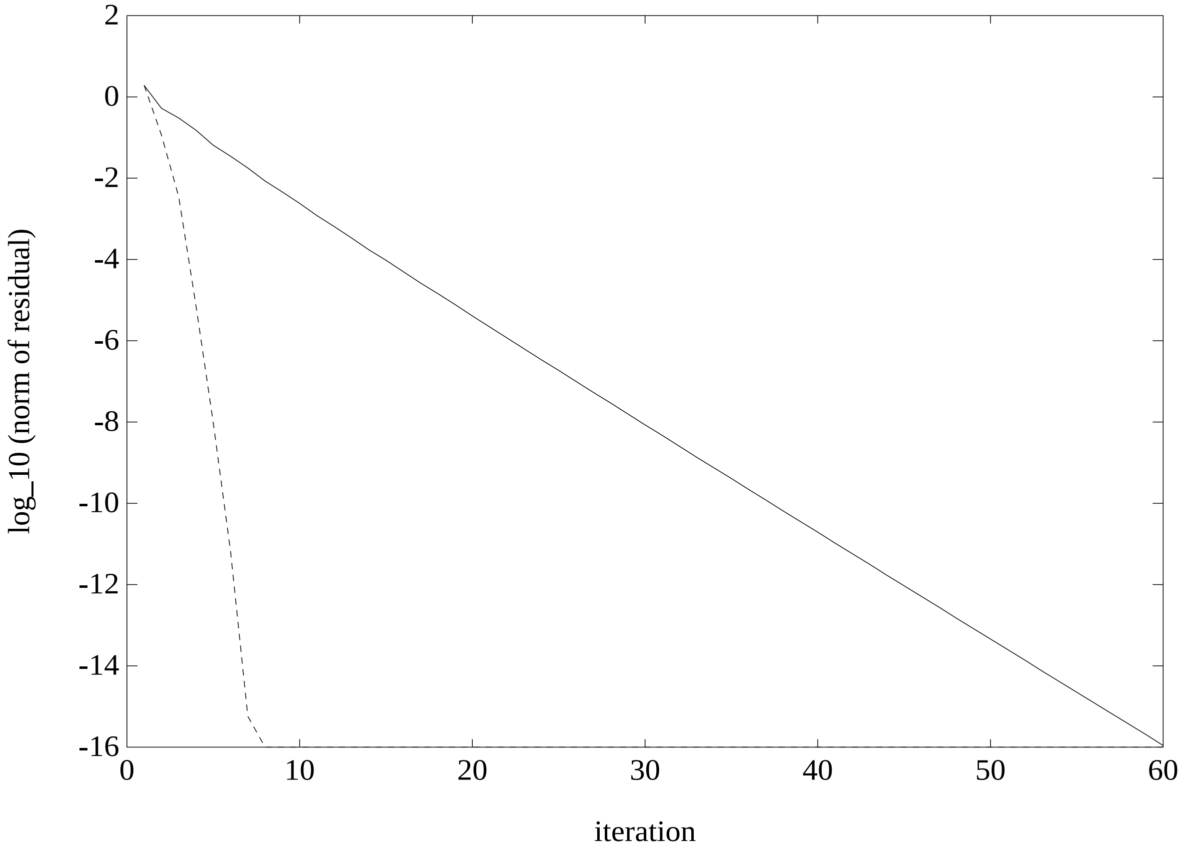
6 A generalization

Both the PR2 acceleration and the PR2 iterative method can be generalized by considering

$$y_n = x_n - Z_n \lambda_n$$

where Z_n is a $p \times k$ matrix, $\lambda_n \in \mathbb{R}^k$ and $k \geq 1$ an integer which can depend on n. The vector λ_n can again be computed so that (ρ_n, ρ_n) is minimized that is such

Figure 4



that

$$\frac{\partial(\rho_n, \rho_n)}{\partial(\lambda_n)_i} = 0$$

for $i = 1, \ldots, k$, where $(\lambda_n)_i$ denotes the ith component of the vector λ_n. In other words, λ_n is given by the least–squares solution of the system $AZ_n\lambda_n = -r_n$, that is

$$\lambda_n = -\left[(AZ_n)^T AZ_n\right]^{-1} (AZ_n)^T r_n.$$

Such a generalization still has to be studied.

Acknowledgments: I would like to thank Charles G. Broyden and David M. Gay for sending me pertinent informations.

This work was supported by the Human Capital and Mobility Programme of the European Community, Project ROLLS, Contract CHRX–CT93–0416 (DG 12 COMA).

References

[1] J. Abaffy, E. Spedicato, *ABS Projection Algorithms*, Ellis Horwood, Chichester, 1989.

[2] C. Brezinski, Hybrid procedures and semi–iterative methods for linear systems, submitted.

[3] C. Brezinski, Projection methods for linear systems, to appear.

[4] C. Brezinski, *Projections and Hybrid Procedures for Systems of Equations*, to appear.

[5] C. Brezinski, M. Redivo–Zaglia, *Extrapolation Methods. Theory and Practice*, North–Holland, Amsterdam, 1991.

[6] C. Brezinski, M. Redivo–Zaglia, Hyprid procedures for solving systems of linear equations, Numer. Math., 67 (1994) 1–19.

[7] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, Math. Comput., 19 (1965) 577–593.

[8] A.M. Bruaset, *A Survey of Preconditioned Iterative Methods*, Longman, Harlow, 1995.

[9] J.D.F. Cosgrove, J.C. Díaz, A. Griewank, Approximate inverse preconditionings for sparse linear systems, Int. J. Comput. Math., 44 (1992) 91–110.

[10] P.F. Dubois, A. Greenbaum, G.H. Rodrigue, Approximating the inverse of a matrix for use in iterative algorithms on vector processors, Computing, 22 (1979) 257–268.

[11] E. Durand, *Solutions Numériques des Équations Algébriques*, 2 vols., Masson, Paris, 1972.

[12] T. Eirola, O. Nevanlinna, Accelerating with rank–one updates, Linear Algebra Appl., 121 (1989) 511–520.

[13] D.J. Evans, *Preconditioning Methods: Theory and Applications*, Gordon and Breach, New York, 1983.

[14] D.J. Evans, *Preconditioned Iterative Methods*, Gordon and Breach, New York, 1994.

[15] N. Gastinel, *Analyse Numérique Linéaire*, Hermann, Paris, 1966.

[16] D.M. Gay, Some convergence properties of Broyden's methods, SIAM J. Numer. Anal., 16 (1979) 623–630.

[17] R.R. Gerber, F.T. Luk, A generalized Broyden's method for solving simultaneous linear equations, SIAM J. Numer. Anal., 18 (1981) 882–890.

[18] G.H. Golub, C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 2nd ed., 1989.

[19] H.Y. Huang, A direct method for the general solution of a system of linear equations, J. Optimization Theory Appl., 16 (1975) 429–445.

[20] M.R. Hestenes, The conjugate–gradient method for solving linear systems, in *Proceedings of the Sixth Symposium in Applied Mathematics*, J. Curtiss ed., American Mathematical Society, Providence, 1956, pp. 83–102.

[21] U. Meier Yang, K.A. Gallivan, A new family of preconditioned iterative solvers for nonsymmetric linear systems, Appl. Numer. Math., to appear.

[22] M. Minoux, *Mathematical Programming, Theory and Algorithms*, Wiley, Chichester, 1986.

[23] W.C. Rheinboldt, *Methods for Solving Systems of Nonlinear Equations*, SIAM, Philadelphia, 1974.

Laboratoire d'Analyse Numérique et d'Optimisation
UFR IEEA – M3
Université des Sciences et Technologies de Lille
59655 – Villeneuve d'Ascq cedex, France
e–mail: `brezinsk@omega.univ-lille1.fr`