Iterative solvers for linear equations

Maximilien Levesque*

August 21, 2014

*École Normale Supérieure, Département de Chimie, Paris, France

Given a square system of n linear equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1}$$

Some (very slow) direct techniques exist that involve algebra. We will solve this equation by an iterative (also called relaxation) method. We will use a sequence of $\boldsymbol{x}^{(k)}$ that converges toward the fixed point $\boldsymbol{x}$, solution of our problem.

In conclusion, we want to write, for a given $\boldsymbol{x}^{(0)}$, a sequence $\boldsymbol{x}^{(k+1)} = F\left(\boldsymbol{x}^{(k)}\right)$ with $k \in \mathbb{N}$.

1 Jacobi's method

We decompose $A = D - L - U$ where D is an inversible matrix.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow \boldsymbol{D}\boldsymbol{x} = (\boldsymbol{L} + \boldsymbol{U})\,\boldsymbol{x} + \boldsymbol{b} \tag{2}$$

$$\begin{aligned} \Leftrightarrow \boldsymbol{x} &= \boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\,\boldsymbol{x} + \boldsymbol{D}^{-1}\boldsymbol{b} &(3)\\ &= F(\boldsymbol{x}) &(4) \end{aligned}$$

where F is a linear function of x.

It is shown that we can solve this equation by iteratively solving, where k indicates the iteration number

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\,\boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\boldsymbol{b}. \tag{5}$$

Since $\mathrm{diag}\left((\boldsymbol{L} + \boldsymbol{U})\,\boldsymbol{x}^{(k)}\right) = \boldsymbol{0}$, each $x_i^{(k+1)}$ is independent of $x_i^{(k)}$. It is instead a weighted average of other $x_{j \neq i}$.

Since we don't know the solution *a priori*, we define the residual at each step k as $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$.

References

[1] http://fr.wikipedia.org/wiki/M%C3%A9thode_de_Jacobi

[2] http://mathworld.wolfram.com/JacobiMethod.html

[3] http://en.wikipedia.org/wiki/Jacobi_method

[4] http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/ebooks/html/csa/node38.html

Algorithm 1 Jacobi's algorithm

Output

- $\boldsymbol{x}$, an approximate solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{A}$ and $\boldsymbol{b}$ are known.

Inputs

- $\boldsymbol{A}$
- $\boldsymbol{b}$
- $\boldsymbol{x}^{(0)}$ including boundary conditions. No restriction on $\boldsymbol{x}^{(0)}$.
- ϵ, the tolerance or convergence criteria. A suitable tolerance might be $\left\|\boldsymbol{r}^{(k)}\right\| < \epsilon$ or $\left\|\boldsymbol{r}^{(k)} - \boldsymbol{r}^{(k-1)}\right\| < \epsilon$.

Restrictions

- $\boldsymbol{A}$ must be strictly diagonaly dominant, *i.e.*, $|A_{i,i}| > \sum_{j=1,n;j\neq i} |A_{i,j}| \Rightarrow A_{i,i} \neq 0$.
- $\epsilon > 0$

Algo

1. $\boldsymbol{D} \longleftarrow \mathrm{diag}\left(\boldsymbol{A}\right)$
2. $\boldsymbol{R} \longleftarrow \boldsymbol{D} - \boldsymbol{A}$
3. $\boldsymbol{D}^{-1} \longleftarrow 1/\boldsymbol{D}$
4. do while $\xi \geq \epsilon$
   (a) $\boldsymbol{x}_{\mathrm{old}} \longleftarrow \boldsymbol{x}$
   (b) $\boldsymbol{x} \longleftarrow \boldsymbol{D}^{-1}\boldsymbol{R}\boldsymbol{x}_{\mathrm{old}} + \boldsymbol{D}^{-1}\boldsymbol{b}$
   (c) $\boldsymbol{r} \longleftarrow \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$
   (d) $\xi \longleftarrow \|\boldsymbol{r}\|$