# Errata #1

This errata was sent to SIAM for the second printing of the book – so you may find these errors if you bought the copies from the first printing (before 2004 or so).
Many thanks to Kees Vuik, Zhong-Zhi Bai, Sabine Le Borne, and Rudnei Dias da Cunha, for bringing a few errors to my attention.

## 1 The easy ones

1. Page 140, Line 5 of section 5.3.2. $(r, r)$ should be $(Ar, r)$. Correct formula is:

$$\alpha \leftarrow (Ar, r)/(Ar, Ar)$$

   The same error occurs again in line 3 of algorithm 5.3 which should be:

   3. $\alpha \leftarrow (Ar, r)/(p, p)$

2. Page 145, Line -3. There is no square in the denominator. Correct formula:

$$\|d_{new}\|_A \leq \left(1 - \frac{1}{n\kappa(A)}\right)^{1/2} \|d\|_A,$$

3. Page 146, line 10. Same formula - same error.
4. Page 425, Line 15. $u^h_{new} = ..$ has a term missing in the brackets. Correct formula is

$$u^h_{new} = S_h^{\nu_2}[S_h^{\nu_1} u_0^h + I_H^h A_H^{-1} I_h^H(-A_h S_h^{\nu_1} u_0^h)].$$

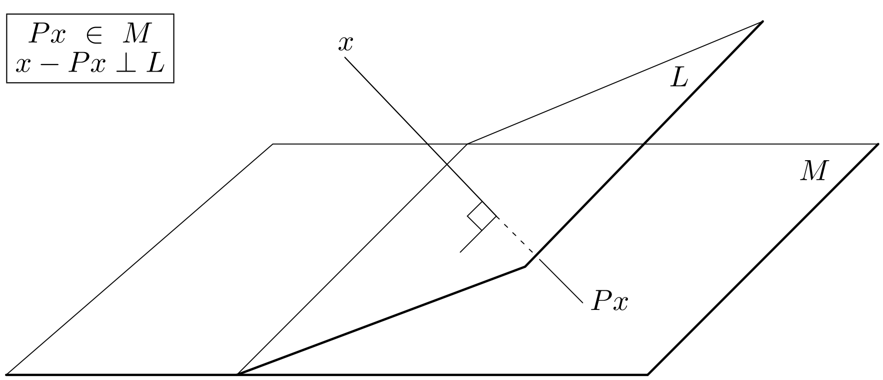
5. Page 426, Line -15: $7/3\eta n$ should be replaced by $(4/3)\eta n$.
6. Page 209, Line 2 of Algorithm 6.24 should be:

   2. For $j = p, p+1, \ldots, m+p-1$, Do:

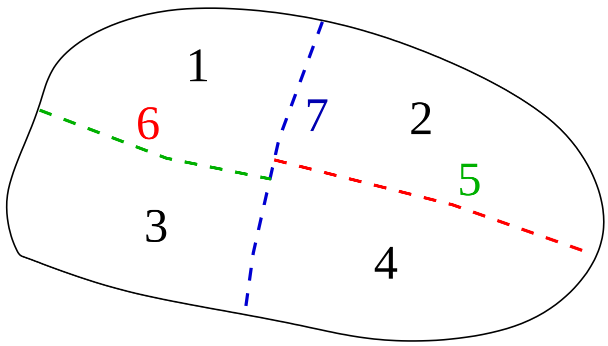
7. Page 191, Lines -1 and -2 and Page 192, line 2: $-\gamma_m$ should be $+\gamma_m$. Also in Line 5. of Algorithm 6.19, $-\gamma_j$ should be $+\gamma_j$.

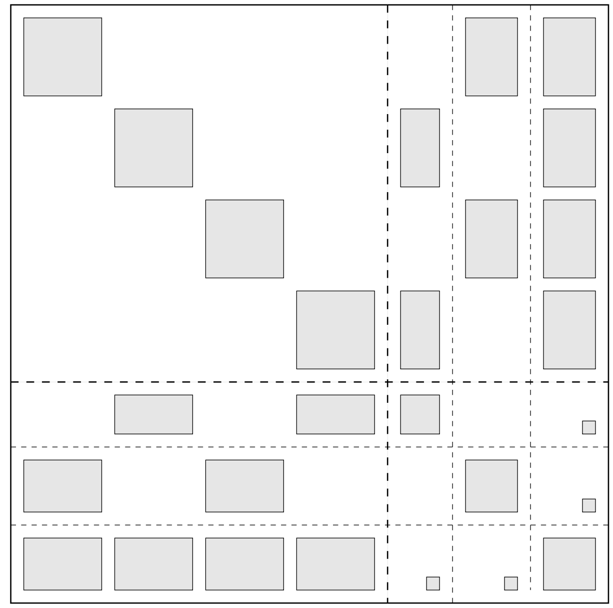
## 2 Problems with figures

1. In Figure 1.1, a big diagonal across the base rectangle got inserted (this was not in my original figure). Here is the original figure:



2. In Figure 3.11 (p. 95). 5 and 6 need to be interchanged in the left figure. The correct figure is:



## 3 Section 9.6

This section contains a few errors (Thanks to Kees Vuik for finding these) and it is best to rewrite the section.

## 4 The Concus, Golub, and Widlund Algorithm

When the matrix is nearly symmetric, we can think of preconditioning the system with the symmetric part of $A$. This gives rise to a few variants of a method known as the CGW method, from the names of the three authors Concus and Golub [88], and Widlund [312] who proposed this technique in the middle of the 1970s. Originally, the algorithm was not viewed from the angle of preconditioning. Writing $A = M - N$, with $M = \frac{1}{2}(A + A^H)$, the authors observed that the preconditioned matrix

$$M^{-1}A = I - M^{-1}N$$

is equal to the identity matrix, plus a matrix which is skew-Hermitian with respect to the $M$-inner product. It is not too difficult to show that the tridiagonal matrix corresponding to the Lanczos algorithm, applied to $A$ with the $M$-inner product, has the form

$$T_m = \begin{pmatrix} 1 & -\eta_2 & & & \\ \eta_2 & 1 & -\eta_3 & & \\ & . & . & . & \\ & & \eta_{m-1} & 1 & -\eta_m \\ & & & \eta_m & 1 \end{pmatrix}. \tag{1}$$

As a result, a three-term recurrence in the Arnoldi process is obtained, which results in a solution algorithm that resembles the standard preconditioned CG algorithm (Algorithm 9.1).

A version of the algorithm can be derived easily. The developments in Section 6.7 relating the Lanczos algorithm to the Conjugate Gradient algorithm, show that the vector $x_{j+1}$ can be expressed as

$$x_{j+1} = x_j + \alpha_j p_j.$$

The preconditioned residual vectors must then satisfy the recurrence

$$z_{j+1} = z_j - \alpha_j M^{-1} A p_j$$

and if the $z_j$'s are to be $M$-orthogonal, then we must have $(z_j - \alpha_j M^{-1} A p_j, z_j)_M = 0$. As a result,

$$\alpha_j = \frac{(z_j, z_j)_M}{(M^{-1} A p_j, z_j)_M} = \frac{(r_j, z_j)}{(A p_j, z_j)}.$$

Also, the next search direction $p_{j+1}$ is a linear combination of $z_{j+1}$ and $p_j$,

$$p_{j+1} = z_{j+1} + \beta_j p_j.$$

Since $M^{-1} A p_j$ is orthogonal to all vectors in $\mathcal{K}_{j-1}$, a first consequence is that

$$(A p_j, z_j) = (M^{-1} A p_j, p_j - \beta_{j-1} p_{j-1})_M = (M^{-1} A p_j, p_j)_M = (A p_j, p_j).$$

In addition, $M^{-1} A p_{j+1}$ must be $M$-orthogonal to $p_j$, so that $\beta_j = -(M^{-1} A z_{j+1}, p_j)_M / (M^{-1} A p_j, p_j)_M$. The relation $M^{-1} A = I - M^{-1} N$, the fact that $N^H = -N$, and that $(z_{j+1}, p_j)_M = 0$ yield,

$$(M^{-1} A z_{j+1}, p_j)_M = -(M^{-1} N z_{j+1}, p_j)_M = (z_{j+1}, M^{-1} N p_j)_M = -(z_{j+1}, M^{-1} A p_j)_M.$$

Finally, note that $M^{-1} A p_j = -\frac{1}{\alpha_j}(z_{j+1} - z_j)$ and therefore we have (note the sign difference with the standard PCG algorithm)

$$\beta_j = -\frac{(z_{j+1}, z_{j+1})_M}{(z_j, z_j)_M} = -\frac{(z_{j+1}, r_{j+1})}{(z_j, r_j)}.$$