最終更新日: 2008 年 7 月 15 日

「C 言語による数値計算入門」(第 4 刷) 正誤表

	誤	正
p.28, ノルムの説明を追記. 分かりやすくするための措置	このベクトルノルムには次のようなものがあります. ただし, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^t \in \mathbb{C}^n$ とします. **1-ノルム** $\displaystyle \|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} \lvert x_i \rvert$ **2-ノルム** $\displaystyle \|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} \lvert x_i \rvert^2}$ ∞-**ノルム**または**最大値ノルム** $\displaystyle \|\boldsymbol{x}\|_\infty = \max_{1 \le i \le n} \lvert x_i \rvert$ ここで, 複素数 $x = a + bi$ に対して $\lvert x \rvert = \sqrt{a^2 + b^2}$ であることに注意して下さい.	このベクトルノルムには次のようなものがあります. ただし, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^t \in \mathbb{C}^n$ とします. また, 複素数 $x = a + bi$ に対して $\lvert x \rvert = \sqrt{a^2 + b^2}$ であることに注意して下さい. **1-ノルム** $\displaystyle \|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} \lvert x_i \rvert$ **2-ノルム** $\displaystyle \|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} \lvert x_i \rvert^2}$ ∞-**ノルム**または**最大値ノルム** $\displaystyle \|\boldsymbol{x}\|_\infty = \max_{1 \le i \le n} \lvert x_i \rvert$ なお, 2-ノルムは複素内積 $\displaystyle (\boldsymbol{x}, \boldsymbol{x}) = \sum_{i=1}^{n} x_i \bar{x}_i$ を使って, $\|\boldsymbol{x}\|_2 = \sqrt{(\boldsymbol{x}, \boldsymbol{x})}$ と書けます.

<table>
<tr><th></th><th>誤</th><th>正</th></tr>
<tr><td>p.30, 最大値ノルムの後</td><td>また，これらの行列ノルムとベクトルノルムの間には $\|A\| = \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$ という関係があります．</td><td>また，フロベニウスノルムを除くこれらの行列ノルムとベクトルノルムの間には $\|A\|_p = \max_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} (p = 1, 2, \infty)$ という関係があります．</td></tr>
<tr><td>p.44, 下から 3 行目</td><td>ます，(3.8) における</td><td>まず，(3.8) における</td></tr>
<tr><td>p.56, スケーリングの例を分かりやすいものに変更</td><td>$0 < |\varepsilon| \ll 1$ のとき，
$$\begin{bmatrix} 1 & \frac{1}{\varepsilon} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\varepsilon} + \varepsilon - 1 \\ 1 \end{bmatrix} \quad (3.16)$$
を解くことを考えます．この (3.16) に対して部分ピボット選択付きガウスの消去法の前進消去を実行すると次のようになります．
$$\begin{bmatrix} 1 & \frac{1}{\varepsilon} \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\varepsilon} + \varepsilon - 1 \\ 2 - \frac{1}{\varepsilon} - \varepsilon \end{bmatrix} \quad (3.17)$$
ここで，$0 < |\varepsilon| \ll 1$ のとき，$|2 - \varepsilon| \ll \left|-\frac{1}{\varepsilon}\right|$, $1 \ll \left|-\frac{1}{\varepsilon}\right|$ となるので，コンピュータ内では $1 - \frac{1}{\varepsilon} \approx -\frac{1}{\varepsilon}$, $2 - \frac{1}{\varepsilon} - \varepsilon \approx -\frac{1}{\varepsilon}$ となる可能性があります．このとき，(3.17) より
$$x_2 = 1, \quad x_1 = \frac{1}{\varepsilon} + \varepsilon - 1 - \frac{1}{\varepsilon}x_2 = \frac{1}{\varepsilon} + \varepsilon - 1 - \frac{1}{\varepsilon} = \varepsilon - 1 \approx -1$$
となりますが，(3.16) の真の解は $x_1 = \varepsilon$, $x_2 = 1 - \varepsilon$ なので x_1 は近似解とはいえません．
このとき，(3.17) にスケーリングを適用すると $s_1 = \max_{1\leq j\leq 2} |a_{1j}| = \left|\frac{1}{\varepsilon}\right|$ なので
$$\begin{bmatrix} \frac{1}{s_1} & \frac{1}{\varepsilon s_1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_1}(\frac{1}{\varepsilon} + \varepsilon - 1) \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 + \varepsilon^2 - \varepsilon \\ 1 \end{bmatrix}$$
となります．そして，部分ピボット選択を行った後に前進消去を実行すると次のようになります．
$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 + \varepsilon^2 - \varepsilon \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 - \varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 + \varepsilon^2 - 2\varepsilon \end{bmatrix}$$
$1 - \varepsilon \approx 1$, $1 + \varepsilon^2 - 2\varepsilon \approx 1$ なので，$x_2 = 1$, $x_1 = 1 - x_2 = 0$ となりますが，こちらの方がスケーリングをしなかった場合よりも良い近似解であるといえます．</td><td>$0 < |\varepsilon| \ll 1$ のとき，
$$\begin{bmatrix} 1 & \frac{1}{\varepsilon} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{2}{\varepsilon} \\ 1 \end{bmatrix} \quad (3.16)$$
を解くことを考えます．この (3.16) に対して部分ピボット選択付きガウスの消去法の前進消去を実行すると次のようになります．
$$\begin{bmatrix} 1 & \frac{1}{\varepsilon} \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{2}{\varepsilon} \\ 1 + \frac{2}{\varepsilon} \end{bmatrix} \quad (3.17)$$
ここで，$0 < |\varepsilon| \ll 1$ のとき，$1 \ll \frac{2}{\varepsilon}$, $1 \ll \left|-\frac{1}{\varepsilon}\right|$ となるので，コンピュータ内では $1 - \frac{1}{\varepsilon} \approx -\frac{1}{\varepsilon}$, $1 + \frac{2}{\varepsilon} \approx \frac{2}{\varepsilon}$ となる可能性があります．このとき，(3.17) より
$$x_2 = -2, \quad x_1 = -\frac{2}{\varepsilon} - \frac{1}{\varepsilon}x_2 = -\frac{2}{\varepsilon} + \frac{2}{\varepsilon} = 0$$
となりますが，(3.16) の真の解は $x_1 = \frac{3}{1 - \varepsilon} \approx 3$, $x_2 = \frac{2 + \varepsilon}{\varepsilon - 1} \approx -2$ なので x_1 は近似解とはいえません．
このとき，(3.17) にスケーリングを適用すると $s_1 = \max_{1\leq j\leq 2} |a_{1j}| = \left|\frac{1}{\varepsilon}\right|$ なので
$$\begin{bmatrix} \frac{1}{s_1} & \frac{1}{\varepsilon s_1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{-2}{\varepsilon s_1} \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$
となります．そして，部分ピボット選択を行った後に前進消去を実行すると次のようになります．
$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 - \varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 - \varepsilon \end{bmatrix}$$
$1 - \varepsilon \approx 1$, $-2 - \varepsilon \approx -2$ なので，$x_2 = -2$, $x_1 = 1 - x_2 = 3$ となりますが，こちらの方がスケーリングをしなかった場合よりも良い近似解であるといえます．</td></tr>
</table>

	誤	正
p.57, 第 3.5 節の直前	以上のことより，与えられた連立一次方程式をあらかじめスケーリングした方が，解を求める過程で丸め誤差の影響を受けにくいことが分かります.	以上のことより，与えられた連立一次方程式をあらかじめスケーリングすると，比較的安定して解が求められることが分かります.
p.61, 下から 6 行目	$$\boldsymbol{x}_m^t A_m \boldsymbol{x}_m = \boldsymbol{x}^t A \boldsymbol{x} > 0 \qquad (3.21)$$	$$\boldsymbol{x}_m^t A_m \boldsymbol{x}_m = \boldsymbol{x}^t A \boldsymbol{x} > 0 \qquad (3.24)$$
p.62, 修正コレスキー分解法の直前	これは (3.21) に矛盾します.	これは (3.24) に矛盾します.
p.62, 行列 L の n 行 1 列要素	$\frac{l_{n1}}{l_{nn}}$	$\frac{l_{n1}}{l_{11}}$
p.62, 下から 5 行目	$$A = LL^t = (\tilde{L}\tilde{D})(\tilde{L}\tilde{D})^t = \tilde{L}\tilde{D}\tilde{L}^t\tilde{L}^t = \tilde{L}D\tilde{L}^t \qquad (3.22)$$ となります．この分解 (3.22) を**修正コレスキー分解**といいます.	$$A = LL^t = (\tilde{L}\tilde{D})(\tilde{L}\tilde{D})^t = \tilde{L}\tilde{D}\tilde{L}^t\tilde{L}^t = \tilde{L}D\tilde{L}^t \qquad (3.25)$$ となります．この分解 (3.25) を**修正コレスキー分解**といいます.
p.70, 上から 3 行目	を直接的に求めることはでません.	を直接的に求めることはできません.
p.75, 定理 4.2 の最後	(4.8) を**リプシッツ条件**という.	(4.8) を**リプシッツ条件**，L をリプシッツ定数という.
p.75, 脚注 2 の最後	無限区間 $(-\infty, \infty)$ などが挙げられます.	無限区間 $(-\infty, \infty)$ などが挙げられます．なお，コーシー列については次ページの脚注 3 を参照してください.
p.76, 下から 2 行目	つまり，縮小写像の原理の仮定 (4.9) が成り立っていれば	つまり，縮小写像の原理の仮定~~(4.9)~~が成り立っていれば
p.78, ニュートン法のアルゴリズムの while 文	`while` $(\lvert d\rvert > \varepsilon$ `or` $n < n_{\max})$	`while` $(\lvert d\rvert > \varepsilon$ `and` $n < n_{\max})$

	誤	正
p.80, 重複次数の推定，分かりやすくするための追記	$f(x)=0$ の解 $x=\alpha$ が m 重解，つまり，$f^{(l)}(\alpha)=0\ (0\leq l\leq m-1)$, $f^{(m)}(\alpha)\neq 0$ の場合を考えます．このとき，テイラーの公式より	$f(x)=0$ の解 $x=\alpha$ が m 重解，つまり，$f(x)$ が $(x-\alpha)^m$ で割り切れ，$(x-\alpha)^{m+1}$ で割り切れない場合を考えます．このとき，テイラーの公式より $f(x)$ を $(x-\alpha)^m$ で割った余りは $\displaystyle\sum_{l=0}^{m-1}\frac{f^{(l)}(\alpha)}{l!}(x-\alpha)^l$ なので，割り切れるためには $f^{(l)}(\alpha)=0(0\leq l\leq m-1)$ でなければなりません．また，$f(x)$ を $(x-\alpha)^{m+1}$ で割った余りは $\displaystyle\sum_{l=0}^{m}\frac{f^{(l)}(\alpha)}{l!}(x-\alpha)^l$ ですが，$(x-\alpha)^{m+1}$ では割り切れないので，直前の結果と合わせると $f^{(m)}(\alpha)\neq 0$ とならなければなりません．したがって，再びテイラーの公式より
p.81, 脚注に追記	となりがちです．	となりがちです．特に，初期値を少し変更したとき，近似解はあまり変わらないのに重複次数が大きく異なる場合は，桁落ちが発生している可能性があります．この場合は，ε をやや大きめに設定してください．
p.82, (4.17) の後に分かりやすくするための追記	α に**線形収束**するといいます．また，	α に**線形収束**するといいます．(4.17) より，$(x_{n+1}-\alpha)=A(x_n-\alpha)+\varepsilon_n(x_n-\alpha)$ で，$x_n-\alpha$ は有限値なので，$\displaystyle\lim_{n\to\infty}\varepsilon_n(x_n-\alpha)=0$ です．一方，$\eta_n\to 0(n\to\infty)$ ならば，適当な $\varepsilon_n\to 0(n\to\infty)$ となる ε_n を選んで $\eta_n=\varepsilon_n(x_n-\alpha)$ とできるので，$\{x_n\}$ が線形収束するための必要十分条件は，$(x_{n+1}-\alpha)=A(x_n-\alpha)+\eta_n, \lvert A\rvert<1, \eta_n\to 0(n\to\infty)$ となります． また，
p.82, 定理 4.3 の証明の 2 行目，分かりやすくするために追記	$f(\alpha)=0$ に注意すると，テイラーの公式より $$x_{n+1}-\alpha=x_n-\alpha-\frac{f(x_n)}{f'(x_n)}=\frac{-(f(x_n)+(\alpha-x_n)f'(x_n))}{f'(x_n)}=\frac{f''(\xi)}{2f'(x_n)}(\alpha-x_n)^2$$ が成り立ちます．	$f(\alpha)=0$ に注意すると，テイラーの公式より $0=f(\alpha)=f(x_n)+f'(x_n)(\alpha-x_n)+\frac{1}{2}f''(\xi)(\alpha-x_n)^2$ なので $$x_{n+1}-\alpha=x_n-\alpha-\frac{f(x_n)}{f'(x_n)}=\frac{-(f(x_n)+(\alpha-x_n)f'(x_n))}{f'(x_n)}=\frac{f''(\xi)}{2f'(x_n)}(\alpha-x_n)^2$$ が成り立ちます．

	誤	正
p.83，割線法の直前，分かりやすくするための追記	ちなみに，2 分法は (4.1) より，線形収束することが分かります．	ちなみに，2 分法は線形収束することが分かります．なぜなら，$a_n \to \alpha$ で，(4.1) より $b_{n+1}-a_{n+1} = \frac{1}{2}(b_n-a_n)$ なので，$b_{n+1}-\alpha = a_{n+1}+\frac{1}{2}(b_n-a_n)-\alpha = \frac{1}{2}(b_n-\alpha)+a_{n+1}-\frac{1}{2}\alpha-\frac{1}{2}a_n$ かつ，$a_{n+1}-\frac{1}{2}\alpha-\frac{1}{2}a_n \to 0 (n\to\infty)$ となるからです．
p. 84, (4.20) の後，分かりやすくするための追記	分子で桁落ちを起こす可能性が高いので (4.20) を使わない方がいいでしょう．	分子で桁落ちを起こす可能性が高いので (4.20) を使わない方がいいでしょう．実際，(4.19) だと右辺第 2 項で桁落ちが起こったとしても，x_n の情報は残りますが，(4.20) では，その情報すらなくなってしまう可能性があります．
p.86, (4.23)	$$f(x_n)-f(\alpha) = \frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}(f(x_n)-\alpha)-\frac{1}{2}(\alpha-x_{n-1})(\alpha-x_n)f''(\xi) \quad (4.23)$$	$$f(x_n)-f(\alpha) = \frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}(x_n-\alpha)-\frac{1}{2}(\alpha-x_{n-1})(\alpha-x_n)f''(\xi) \quad (4.23)$$
p.88, 連立非線形方程式に対するニュートン法のアルゴリズムの while 文	`while` $(\lvert d\rvert > \varepsilon$ `or` $k < k_{\max})$	`while` $(\lvert d\rvert > \varepsilon$ `and` $k < k_{\max})$
p.95	これより，$$\beta \log\lVert M\rVert + \log\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \approx \log\varepsilon$$ となりますが，通常，収束するときには $\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \le 1$ が期待できるので $\log\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \le 0$ となります．よって，反復回数を $\beta\log\lVert M\rVert \ge \log\varepsilon$，つまり，$$\beta \ge \frac{\log\varepsilon}{\log\lVert M\rVert} \quad (5.4)$$ と見積もることができます．ここで，$\varepsilon < 1$ なので (5.4) は $\lVert M\rVert < 1$ のときに成立することに注意してください．	これより，$$\beta \log\lVert M\rVert + \log\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \approx \log\varepsilon$$ となりますが，通常，$\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \ge 1$ となることが多いので，$\log\lVert \boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\rVert \ge 0$ と考えられます．よって，反復回数を $\beta\log\lVert M\rVert \le \log\varepsilon$，と見積もることができます．ここで，$\lVert M\rVert < 1$ より $\log\lVert M\rVert < 0$ であることに注意すると，結局，$$\beta \ge \frac{\log\varepsilon}{\log\lVert M\rVert} \quad (5.4)$$ が得られます．

	誤	正
p.96, ヤコビ法のアルゴリズムの while 文	`while` $(\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| > \varepsilon$ `or` $k < k_{\max})$	`while` $(\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| > \varepsilon$ `and` $k < k_{\max})$
p.103 の下から 2 行目	$\|y_2\| = \left\|\frac{a_{21}}{a_{22}}\right\| \|y_1\| + \sum_{j=3}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| \|x_j\| < \left\|\frac{a_{21}}{a_{22}}\right\| + \sum_{j=3}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| = \sum_{j=1, j\neq 2}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| < 1$	$\|y_2\| \leq \left\|\frac{a_{21}}{a_{22}}\right\| \|y_1\| + \sum_{j=3}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| \|x_j\| < \left\|\frac{a_{21}}{a_{22}}\right\| + \sum_{j=3}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| = \sum_{j=1, j\neq 2}^{n} \left\|\frac{a_{2j}}{a_{22}}\right\| < 1$
p.104, 上から 2 行目	なので，$\|y_i\| < 1 (j < i)$ に注意すると $\|y_i\| \leq \sum_{j=1}^{i-1} \left\|\frac{a_{jj}}{a_{ii}}\right\| \|y_i\| + \sum_{j=i+1}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| \|x_j\| < \sum_{j=1}^{i-1} \left\|\frac{a_{ij}}{a_{ii}}\right\| + \sum_{j=i+1}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| = \sum_{j=1, j\neq i}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| < 1$	なので，$\|y_j\| < 1 (j < i)$ に注意すると $\|y_i\| \leq \sum_{j=1}^{i-1} \left\|\frac{a_{jj}}{a_{ii}}\right\| \|y_j\| + \sum_{j=i+1}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| \|x_j\| < \sum_{j=1}^{i-1} \left\|\frac{a_{ij}}{a_{ii}}\right\| + \sum_{j=i+1}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| = \sum_{j=1, j\neq i}^{n} \left\|\frac{a_{ij}}{a_{ii}}\right\| < 1$
p.108 の (5.21)，読者の利便性を高めるため，脚注を追記	また，この対偶より，SOR 法は収束するならば， $0 < \omega < 2 \quad (5.21)$ となることが分かります．ただし，	また，この対偶より，SOR 法は収束するならば， $0 < \omega < 2 \quad (5.21)$ となることが分かります[4]．ただし， †4 行列 A が n 次実対称正定値の場合，(5.21) が成り立てば SOR 法が収束する (つまり，逆が成立する) ことが知られています．例えば，文献 [18] を参照してください．
p.109, 1 行目	直接的にはには関係ありませんが，	直接的には~~には~~関係ありませんが，

	誤	正
p.114, (5.27) の証明の中程	$i < k$ のとき， $\begin{aligned}(\boldsymbol{r}_{k+1}, \boldsymbol{r}_i) &= (\boldsymbol{r}_i, \boldsymbol{r}_{k+1}) \\ &= (\boldsymbol{r}_i, \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= -\alpha_k(\boldsymbol{r}_i, A\boldsymbol{p}_k) \quad \text{帰納法の仮定より} \\ &= -\alpha_k(\boldsymbol{p}_i - \beta_{i-1}\boldsymbol{p}_{i-1}, A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= 0 \quad \text{帰納法の仮定より}\end{aligned}$ $i = k$ のとき $\begin{aligned}(\boldsymbol{r}_{k+1}, \boldsymbol{r}_k) &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{r}_k, A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{p}_k - \beta_{k-1}\boldsymbol{p}_{k-1}, A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{p}_k, A\boldsymbol{p}_k) \quad \text{帰納法の仮定より}\end{aligned}$	$i < k$ のとき， $\begin{aligned}(\boldsymbol{r}_{k+1}, \boldsymbol{r}_i) &= (\boldsymbol{r}_i, \boldsymbol{r}_{k+1}) \\ &= (\boldsymbol{r}_i, \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= -\alpha_k(\boldsymbol{r}_i, A\boldsymbol{p}_k) \quad \text{帰納法の仮定より} \\ &= -\alpha_k(\boldsymbol{p}_i - \beta_{i-1}\boldsymbol{p}_{i-1}, A\boldsymbol{p}_k) \quad (5.26)\text{ より} \\ &= 0 \quad \text{帰納法の仮定より}\end{aligned}$ $i = k$ のとき $\begin{aligned}(\boldsymbol{r}_{k+1}, \boldsymbol{r}_k) &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{r}_k, A\boldsymbol{p}_k) \quad (5.24)\text{ より} \\ &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{p}_k - \beta_{k-1}\boldsymbol{p}_{k-1}, A\boldsymbol{p}_k) \quad (5.26)\text{ より} \\ &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{p}_k, A\boldsymbol{p}_k) \quad \text{帰納法の仮定より}\end{aligned}$
p.123, 下から 2 行目	したがって，データに誤差が含まれていたとしても，それを吸収する働きをもち，その結果，データのおおよそ特徴を把握することができます．	したがって，データに誤差が含まれていたとしても，~~それを吸収する働きをもち，その結果，~~データのおおよその特徴を把握することができます．
p.124, バンデルモンドの行列式	$\det V = \prod_{i>k}(x_i - x_k) \neq 0$ です．よって，	$\det V = \prod_{0 \leq i < j \leq n}(x_j - x_i) \neq 0$ です．ここで，上式の右辺は，$0 \leq i < j \leq n$ を満たす組 (i, j) のすべてに対して，$(x_j - x_i)$ の積をとることを意味します． よって，
p.125, プログラム 6.2 の 4 行目	void input_vector3(double *b, char c, int m, int n, FILE *fin);	void input_vector3(double *b, ~~char c,~~ int m, int n, FILE *fin);
p.126, 中程	input_vector3(x, 'x', 0, n, fin); /* ベクトル x の入出力 */ input_vector3(y, 'y', 0, n, fin); /* ベクトル y の入出力 */	input_vector3(x, ~~'x',~~ 0, n, fin); /* ベクトル x の入出力 */ input_vector3(y, ~~'y',~~ 0, n, fin); /* ベクトル y の入出力 */
p.127, プログラム 6.2 の最後の関数	void input_vector3(double *b, char c, int m, int n, FILE *fin);	void input_vector3(double *b, ~~char c,~~ int m, int n, FILE *fin);

	誤	正
p.131，ニュートン補間のアルゴリズム	1. データ $(x_i, f_i)(i = 0, 1, \ldots, n)$ と補間点 x の入力 2. $i = 0, 1, \ldots, n$ に対して (6.22) を計算する	1. データ $(x_i, f_i)(i = 0, 1, \ldots, n)$ と補間点 x の入力 2. ~~$i = 0, 1, \ldots, n$ に対して~~ (6.22) を計算する
p.164，上から 5 行目	$k_3 = f(x + \alpha_1 h, y + (\beta k_1 + \gamma_1 k_2)h)$	$k_3 = f(x + \alpha_1 h, y + (\beta_1 k_1 + \gamma_1 k_2)h)$
p.165, 中程	(8.17) に (8.19) と (8.21)〜(8.24) を代入して $O(h^5)$ の項を無視する[5]と	(8.17) の左辺に (8.19) を代入し，(8.17) の右辺に (8.21)〜(8.24) を代入した後，$O(h^5)$ の項を無視[5]して，両辺を h で割ると，
p.165, 脚注	つまり，h^4 の項まで一致させる，ということです．	つまり，h^4 の項まで一致させる，ということです．また，こうして得られる関係式は，h の値と f の形に依存しないようにしておきたいものです．そこで，h の各次数および f の導関数の形ごとに分けて，(8.25) のように，f, hDf, h^2D^2f, h^2f_yDf, h^3D^3f, $h^3f_yD^2f$, $h^3f_y^2Df$, h^3Df_yDf ごとに係数を比較します．
p.167, 8.5 節の直前	とすると次数 4 のグンゲ・クッタ公式 (8.15)(8.16) が得られます．	とすると次数 4 のルンゲ・クッタ公式 (8.15)(8.16) が得られます．
p.186，べき乗法のアルゴリズムの前，分かりやすくするための措置	となるので，小さな正数 ε に対して $\\|\boldsymbol{v}\\|_2^2 - \|\lambda^{(k)}\|^2 < \varepsilon$ が成立すれば，$\lambda^{(k)}$ を近似固有値と見なせることが分かります．ただし，$\\|\cdot\\|_2$ はベクトルの 2-ノルムです． 以上をまとめると次のようなアルゴリズムが得られます．ここで，$\boldsymbol{x}^{(0)}$ は正規化された初期ベクトルです．	となるので，小さな正数 ε に対して $\\|\boldsymbol{v}\\|_2^2 - \|\lambda^{(k)}\|^2 < \varepsilon^2$ が成立すれば，$\lambda^{(k)}$ を近似固有値と見なせることが分かります．ただし，$\\|\cdot\\|_2$ はベクトルの 2-ノルムです． 以上をまとめると次のようなアルゴリズムが得られます．ここで，$\boldsymbol{x}^{(0)}$ は正規化された初期ベクトルです．また，収束判定条件では ε^2 の代わりに ε を利用しています．ε は任意に設定できる正数なので，結局，ε^2 も任意に設定された数です．したがって，ε^2 を ε に代えても何ら問題はありません．

	誤	正
p.186，べき乗法のアルゴリズム	`Input` A，$\boldsymbol{x}^{(0)}$，ε `do`	`Input` A，$\boldsymbol{x}^{(0)}$，ε $k \leftarrow 0$ `do`
p.190，定理 9.3，相似変換をより一般的な表現に変更	行列 A の固有値を λ, 対応する固有ベクトルを $\boldsymbol{x}$ とすると，正則行列 M による行列 MAM^{-1} および $M^{-1}AM$ の固有値は λ であり，対応する固有ベクトルは，それぞれ $M\boldsymbol{x}$ および $M^{-1}\boldsymbol{x}$ である． 特に，M が直交行列ならば，MAM^t と M^tAM は A と同じ固有値 λ を持ち，対応する固有ベクトルはそれぞれ $M\boldsymbol{x}$ と $M^t\boldsymbol{x}$ である． なお，MAM^t および M^tAM を A の**相似変換**という．	行列 A の固有値を λ, 対応する固有ベクトルを $\boldsymbol{x}$ とすると，正則行列 M による行列 MAM^{-1} および $M^{-1}AM$ の固有値は λ であり，対応する固有ベクトルは，それぞれ $M\boldsymbol{x}$ および $M^{-1}\boldsymbol{x}$ である．特に，M が直交行列ならば，MAM^t と M^tAM は A と同じ固有値 λ を持ち，対応する固有ベクトルはそれぞれ $M\boldsymbol{x}$ と $M^t\boldsymbol{x}$ である． なお，$AX = XB$ かつ X が正則行列のとき，A と B の固有値が等しいならば，A と B は相似であるといい，B は A の**相似変換**という．したがって，M が直交行列のときは，MAM^t および M^tAM は相似変換である．
p.191, 補題 9.1 の証明の最後	なお，$\boldsymbol{x}\boldsymbol{y} = 2(\boldsymbol{u}^t\boldsymbol{x})\boldsymbol{u}$ なので，$\boldsymbol{u}^t\boldsymbol{x} \neq 0$ となることに注意してください．	なお，$\boldsymbol{x} - \boldsymbol{y} = 2(\boldsymbol{u}^t\boldsymbol{x})\boldsymbol{u}$ なので，$\boldsymbol{u}^t\boldsymbol{x} \neq 0$ となることに注意してください．
p.191, 定理 9.4 の直前	A が非対称行列の場合は，	A が n 次実行列の場合は，
p.191, 定理 9.4	A を n 次実非対称行列とすれば，	A を n 次実~~非対称~~行列とすれば，
p.192，下から 2 行目	そして， $$\boldsymbol{u}_1 = \begin{bmatrix} 0 \\ \boldsymbol{v}_1 \end{bmatrix}, \quad P_1 = E_n - 2\boldsymbol{u}_1\boldsymbol{u}_1^t = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & Q_1 \end{bmatrix}$$ とおくと，	そして， $$\boldsymbol{u}_1 = \begin{bmatrix} 0 \\ \boldsymbol{v}_1 \end{bmatrix}, \quad P_1 = E_n - 2\boldsymbol{u}_1\boldsymbol{u}_1^t = \begin{bmatrix} 1 & \boldsymbol{0}^t \\ \boldsymbol{0} & Q_1 \end{bmatrix}$$ とおくと，
p.192, 定理 9.4 の証明の後	実際に，実非対称行列をヘッセンベルグ行列に	実際に，実~~非対称~~行列をヘッセンベルグ行列に
p.193，ハウスホルダー変換のアルゴリズムの前に追記	また，(9.22) における $\boldsymbol{v}_k$ の計算において $\boldsymbol{b}_k \approx s_k\boldsymbol{e}_k$ のとき桁落ちが発生します．これを防ぐために，s_k の符号は $\boldsymbol{b}_k$ の第 1 要素と異符号にします．	また，(9.22) における $\boldsymbol{v}_k$ の計算において $\boldsymbol{b}_k \approx s_k\boldsymbol{e}_k$ のとき桁落ちが発生します．これを防ぐために，s_k の符号は $\boldsymbol{b}_k$ の第 1 要素 $\boldsymbol{b}_{k,1}$ と異符号にします．なお，$\boldsymbol{b}_{k,1} = 0$ のときは，$\boldsymbol{b}_{k,1} - s_k$ の演算で桁落ちは起こりません．

	誤	正
p.193，プログラム 9.2 の 5 行目，少し汎用性を高めるための措置	`#define N 4`	`#define N 4` `#define EPS pow(10.0, -15.0)`
p.194，プログラム 9.2 の下から 6 行目，上記の変更に伴う措置	`if ( fabs(ss) <= 0.0 ) continue;`	`if ( fabs(ss) < EPS ) continue;`
p.197,(9.34) の上	A の正則性より Q_1, R_1 も正則となるので，	A の正則性より Q_2, R_1 も正則となるので，

	誤	正
p.201, (9.39)	$r_{ii} \leftarrow \sqrt{a'^2_{ii} + a^2_{i+1,i}}$ $\sin\theta_i \leftarrow \dfrac{a_{i+1,i}}{r_{ii}}, \quad \cos\theta_i \leftarrow \dfrac{a'_{ii}}{r_{ii}}$ $a'_{ij} \leftarrow a'_{ij}\cos\theta_i + a_{i+1,j}\sin\theta_i \quad (j = i+1, i+2, \ldots, n)$ $a_{i+1,j} \leftarrow -a'_{ij}\sin\theta_i + a_{i+1,j}\cos\theta_i \quad (j = i+1, i+2, \ldots, n)$	$r_{ii} \leftarrow \sqrt{a'^2_{ii} + a^2_{i+1,i}}$ $\sin\theta_i \leftarrow \dfrac{a_{i+1,i}}{r_{ii}}, \quad \cos\theta_i \leftarrow \dfrac{a'_{ii}}{r_{ii}}$ $r_{ij} \leftarrow a'_{ij}\cos\theta_i + a_{i+1,j}\sin\theta_i \quad (j = i+1, i+2, \ldots, n)$ $a'_{i+1,j} \leftarrow -a'_{ij}\sin\theta_i + a_{i+1,j}\cos\theta_i \quad (j = i+1, i+2, \ldots, n)$ $a'_{ij} \leftarrow r_{ij} \quad (j = i+1, i+2, \ldots, n)$
p.201, 中程	A_k を求めた時点で，十分小さな数 $\varepsilon > 0$ に対し $\lvert a_{j,j-1}\rvert > \varepsilon$ を満たす最大値を m とすると，	A_k を求めた時点で，十分小さな数 $\varepsilon > 0$ に対し $\lvert a_{j,j-1}\rvert > \varepsilon$ を満たす j の最大値を m とすると，
p.202，QR 法のアルゴリズム	$m \leftarrow m-1$ としてループを抜ける	$m \leftarrow m-1$ として，while の条件式 $m > 1$ の評価を行なう
p.202，プログラム 9.3, プログラム 9.2 の変更に伴う措置	`#define N 4`	`#define N 4` `#define EPS pow(10.0, -15.0)`

	誤	正
p.206, 補題 9.3 の証明	$A = QR$ において，R は上三角行列なので $R^{-1} = [r'_{ij}]$ も上三角行列です．よって，$Q = AR^{-1}$ を比較すると	$A = QR$ において，R は上三角行列なので $R^{-1} = [r'_{ij}]$ も上三角行列です．よって，$Q = AR^{-1}$ の両辺の各成分をを比較すると
p.206, 補題 9.4 の証明	このとき，$A\boldsymbol{x} = -\boldsymbol{x}$ なので $$\\|\boldsymbol{x}\\| \leq \\|A\boldsymbol{x}\\| \leq \\|A\\| \cdot \\|\boldsymbol{x}\\|$$ となり，$\\|A\\| \geq 1$ となりますが，これは	このとき，$A\boldsymbol{x} = -\boldsymbol{x}$ なので $$\\|\boldsymbol{x}\\| = \\|A\boldsymbol{x}\\| \leq \\|A\\| \cdot \\|\boldsymbol{x}\\|$$ となり，$\\|A\\| \geq 1$ となりますが，これは
p.209，中程	したがって，QR 分解の一意性と補題 9.6 より， $$Q_1Q_2\cdots Q_k = Q\tilde{Q}_kD_2^{-1}D_1^{-k}$$ $$R_1R_2\cdots R_k = D_1^kD_2\tilde{R}_kR\Lambda^kU$$ となります．これを (9.38) に代入すると， $$A_{k+1} = (Q\tilde{Q}_kD_2^{-1}D_1^k)^tA(Q\tilde{Q}_kD_2^{-1}D_1^{-k})$$ を得ます．そして，$A = X\Lambda X^{-1}$, $X = QR$ より，	したがって，QR 分解の一意性と補題 9.6 より， $$Q_1Q_2\cdots Q_k = Q\tilde{Q}_kD_2^{-1}D_1^{-k}$$ $$R_kR_{k-1}\cdots R_1 = D_1^kD_2\tilde{R}_kR\Lambda^kU$$ となります．これを (9.38) に代入すると， $$A_{k+1} = (Q\tilde{Q}_kD_2^{-1}D_1^{-k})^tA(Q\tilde{Q}_kD_2^{-1}D_1^{-k})$$ を得ます．そして，$A = X\Lambda X^{-1}$, $X = QR$ より，
p.210, 中程	が成り立ちます．ここで，近似固有値 $\hat{\lambda}_i$ が真の固有値 λ_i に十分近い，つまり $$\lvert\lambda_i - \hat{\lambda}_i\rvert < \lvert\lambda_j - \hat{\lambda}_j\rvert, \quad i \neq j$$ を満たすと仮定すれば，	が成り立ちます．ここで，近似固有値 $\hat{\lambda}_i$ が真の固有値 λ_i に十分近い，つまり $$\lvert\lambda_i - \hat{\lambda}_i\rvert < \lvert\lambda_j - \hat{\lambda}_i\rvert, \quad i \neq j$$ を満たすと仮定すれば，
p.211，プログラム 9.4, プログラム 9.2 の変更に伴う措置	`#define N 4`	`#define N 4` `#define EPS pow(10.0, -15.0)`