Improvement of the Measurement Update Step of EKF-SLAM

Zeyneb Kurt Yavuz, Sırma Yavuz,
Computer Engineering Department
Yildiz Technical University, Istanbul, Turkey
zeyneb@ce.yildiz.edu.tr, sirma@ce.yildiz.edu.tr

***Abstract*-In this study, the measurement update step of the Extended Kalman Filter (EKF)-based Simultaneous Localization and Mapping (SLAM) is improved. The computational complexity of the measurement uncertainty matrix inversion operation in the measurement update step is reduced via using Jacobi iteration method. It is observed that, the calculation of the measurement uncertainty matrix inverse by using Jacobi iteration method generates numerically more stable results than naive single and batch update operations. Moreover, it produces more accurate results than the results of Cholesky decomposition with less complexity.**

I. INTRODUCTION

The simultaneous localization and map building (SLAM) techniques tries to solve the problem for an autonomous vehicle to start in an unknown location in an unknown environment to incrementally build a map of this environment. The robot uses this map to compute its own location simultaneously. It can only sense noisy, probabilistic observations of its surroundings without knowing its exact location. During the robot moves, uncertainty is added into an already uncertain pose.

Among the methods proposed in literature to solve the SLAM problem, the methods based on Bayesian estimation theory has been the most successful ones. Many applications in the literature use Extended Kalman Filter (EKF) to solve nonlinear estimation problems such as position tracking, localization and SLAM [1-10]. However, the quadratic computational complexity of the EKF makes it difficult to apply in real time. Unscented Kalman Filter (UKF) is a more reliable estimator than EKF while the system model is highly nonlinear. The past of the UKF is relatively short compared to EKF. By approximating the probability density function, instead of the nonlinear function itself, UKF SLAM [11, 12] received a considerable attention. Yet it did not make any improvement to the computational complexity of the EKF. FastSLAM [13-17] utilizes particle filters and improves the computational complexity considerably compared to EKF and UKF. However in case of having insufficient particle numbers in the filter; FastSLAM can not estimate the state successfully.

In this study, reducing the complexity of the measurement update step of the EKF-based SLAM applications is aimed. Inverse of the measurement uncertainty matrix is calculated in the measurement update step of EKF. Minimum complexity of the matrix inversion operation is O($n^{2.4}$), achieved by Coppersmith–Winograd algorithm.

We present the Jacobi iteration method instead of matrix inversion process in EKF to reduce computational complexity. In EKF-based SLAM applications the innovation vector, the difference between the predicted and the actual observations, is taken into account for the state estimation via using batch or single update for each observation. The convergence of the measurement uncertainty matrix inversion by Jacobi iteration method produces numerically more stable and accurate results than the results of single and batch update approaches which use Cholesky decomposition with less complexity.

The measurement uncertainty matrix, whose inverse is calculated by Jacobi iteration method, is a symmetric and positive-definite matrix. The computational complexity of the inversion calculation of that special matrix by Jacobi iteration method is O(*N*), while the size of the measurement uncertainty matrix is *NxN*. However, the iterative solution of Jacobi method is converged to the inverse of the interested matrix in 8 of 10 experiments. Each of the EKF-based SLAM approaches with single and batch Cholesky updates, those use naive matrix inversion calculation; and batch update, that uses matrix inversion by Jacobi iteration, were run 10 times. The averages of the Root Mean Square Error (RMSE) results of the approaches are given in the experimental results.

In the second section of the paper, the EKF algorithm is described. In Third section the single and batch Cholesky updates and complexity of them are given. In forth section, the matrix inversion by Jacobi iteration method and the complexity analysis of the EKF algorithm which uses Jacobi iteration method for matrix inversion process are introduced. In the Fifth section the experimental results are illustrated and conclusion is given in the final section.

II. EKF-BASED SLAM ALGORITHM

EKF-based SLAM algorithm bases on Bayes Filters. The state of the system, the state of the robot and the environment, at time *t* is expressed by random variables x_t. In the probabilistic SLAM methods, the state of the robot and the environment can only be expressed through the conditional probability distributions of the sensor data. The probability distribution representing the uncertainty at each point in the time is called ***belief***, $bel(x_t)$. Bayes filters apply two update rules successively to estimate the system state [8].

978-1-4673-2695-7/12/$31.00 ©2012 IEEE

The predictive belief at time t is calculated just before the observation (z_t) is taken and uses the control data ($u_{1:t}$) by the time t [8]. This step is also called control update, and expressed as in (1).

$$\overline{bel(x_t)} = P(x_t \mid z_{1:t-1}, u_{1:t}) \qquad (1)$$

The state estimate given in (1) is corrected according to (2), using sensor measurements ($z_{1:t}$) and the control data ($u_{1:t}$) by the time t. This step is called as measurement update or the posterior belief of the system state and calculated whenever a sensor provides a new observation [8].

$$bel(x_t) = P(x_t \mid z_{1:t}, u_{1:t}) \qquad (2)$$

Kalman filters are the most widely used variant of Bayes filters. Standard Kalman filter assumes that the state transitions and measurement transitions are linear with added Gaussian noise. EKF overcomes the linearity assumption by linearizing the nonlinear state and measurement transition functions via Taylor expansion.

The nonlinear state transition function g and measurement transition function h is expressed as in (3) and (4) respectively with added noises ε_t and δ_t.

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \qquad (3)$$

$$z_t = h(x_t) + \delta_t \qquad (4)$$

EKF represents beliefs by the mean vector $\boldsymbol{\mu}_t$ and the covariance matrix $\boldsymbol{\Sigma}_t$ at time t.

The Taylor expansion of state transition function g is given as in (5) and (6). The Jacobian $\mathbf{G}_t$ is the value of first derivative of g at the point $\boldsymbol{\mu}_{t-1}$ [8].

$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) \approx g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + g'(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}) \qquad (5)$$

$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) = g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}) \qquad (6)$$

Similarly, the Taylor expansion of measurement transition function h is given as in (7) and (8). The Jacobian $\mathbf{H}_t$ is the value of first derivative of h at the point $\boldsymbol{\mu}_t$ [8].

$$h(\mathbf{x}_t) \approx h(\overline{\boldsymbol{\mu}_t}) + h'(\overline{\boldsymbol{\mu}_t})(\mathbf{x}_t - \overline{\boldsymbol{\mu}_t}) \qquad (7)$$

$$h(\mathbf{x}_t) = h(\overline{\boldsymbol{\mu}_t}) + \mathbf{H}_t(\mathbf{x}_t - \overline{\boldsymbol{\mu}_t}) \qquad (8)$$

The update rules corresponding to the prediction and correction steps of the Bayes filter and other details of the EKF algorithm can be found in [8-10].

In this study observation has two-dimensions, which are called the range and the bearing information of a laser scanner. Range is the distance between the robot and obstacles (landmarks); bearing is the angular difference between the robot and the obstacle. After taking the observation z_t, data association procedure is done to detect whether the measurements belong to an already existing landmark or to a new landmark. The measurements that belong to already existing landmarks cause update process in the filter. The single and batch update processes differ in terms of the calculating the innovation vectors and evaluating the effects of the associated landmarks on the update process. In the single update, the interested calculations and evaluations of the filter are handled one-by-one for each measurement value. In the batch update, they are handled together.

The pseudo code of the EKF-based SLAM algorithm [18] is given as follows:

```
1. [ $\overline{\boldsymbol{\mu}}_t, \overline{\Sigma}_t$ ] = Predict ( $\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}$ )
2. $\mathbf{z}_t$ = Get observations ( )
3. [ $\mathbf{z_f}, \mathbf{z_n}$ ] = Data associate ( $\overline{\boldsymbol{\mu}}_t, \overline{\Sigma}_t, \mathbf{z}_t$, R)
4. [ $\boldsymbol{\mu}_t, \Sigma_t$ ] = Update Map ( $\boldsymbol{\mu}_t, \Sigma_t, \mathbf{z_f}$, R)
5. [ $\boldsymbol{\mu}_t, \Sigma_t$ ] = Augment Map ( $\boldsymbol{\mu}_t, \Sigma_t, \mathbf{z_n}$, R)
```

where **R** is the covariance matrix of the observation noise; $\mathbf{z_f}$ is the vector of measurements that belong to associated (already existing) landmarks; $\mathbf{z_n}$ is the vector of measurements that belong to unseen landmarks.

III. SINGLE AND BATCH UPDATE IN EKF ALGORITHM

If the effects of the measurements of the associated landmarks on the update process of the filter are taken into account one-by-one for each measurement, it is called single update process; if they are handled together, it is called batch update. Pseudo codes and complexity analysis of two updating approaches are given in the following subsections.

A. Single Update Approach

The associated measurements are handled one-by-one in the single update process. The difference between the actual and the predicted measurements is called innovation vector. The size of the innovation vector is two for only one measurement, because each measurement has two-dimensions, which are called range and bearing information. The measurement estimation uncertainty matrix is a *2x2* matrix for only one measurement.

While N is the number of associated measurements, pseudo code of the single update approach:

```
1. For i=1 to N
    a. [ $\mathbf{z_p}$ , $\mathbf{H}_t$ ] = h ( $\overline{\boldsymbol{\mu}}_t$ ) //h is the observ. model
    b. $\mathbf{v} = \mathbf{z_p} - \mathbf{z_f}$ // v is the innovation vector
```

```
        c.  $\mathbf{S_t} = \mathbf{H_t}\bar{\boldsymbol{\Sigma}}_t\mathbf{H_t^T} + \mathbf{R}$ // meas. uncert. matrix
        d.  $\mathbf{K_t} = \bar{\boldsymbol{\Sigma}}_t\mathbf{H_t^T}\mathbf{S_t^{-1}}$ // Kalman gain
        e.  $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K_t}\mathbf{v}$ // update of the mean vec.
        f.  $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K_t}\mathbf{H_t})\bar{\boldsymbol{\Sigma}}_t$ // update of the cov.
    2.  End for
```

The Kalman gain, $\mathbf{K_t}$, is calculated by using the inverse of the measurement uncertainty matrix, $\mathbf{S_t}$. To calculate the inverse of $\mathbf{S_t}$; Cholesky decomposition can be used as in [18].

The complexity of the inversion of a *2x2* matrix is *$O(2^{2.4})$;* therefore the computational complexity of the $\mathbf{S_t}$'s inversion operation is *$O(Nx2^{2.4})$* in the single update for all observations, while *N* is the number of the associated measurements.

B. Batch Update Approach

The associated measurements are handled together within a joint matrix in the batch update process. The innovation vector is a 2*Nx1* vector when the number of the associated measurements is *N,* because each measurement has two-dimensions (features). And also the measurement estimation uncertainty matrix is a 2*N*x2*N* matrix.

Pseudo code of the batch update approach:

```
    1.  For i=1 to N
            a.  $[\mathbf{z_p},\ \mathbf{H_t}(\mathbf{2i-1:2i,2i-1:2i})] = \mathrm{h}(\bar{\boldsymbol{\mu}}_t)$
            b.  $\mathbf{v(2i\text{-}1{:}2i)} = \mathbf{z_p} - \mathbf{z_f}$ // v is innovation vec.
            c.  $\mathbf{RR(2i-1:2i,2i-1:2i)} = \mathbf{R}$
    2.  End for
    3.  $\mathbf{S_t} = \mathbf{H_t}\bar{\boldsymbol{\Sigma}}_t\mathbf{H_t^T} + \mathbf{RR}$ // meas. uncertainty matrix
    4.  $\mathbf{K_t} = \bar{\boldsymbol{\Sigma}}_t\mathbf{H_t^T}\mathbf{S_t^{-1}}$ // Kalman gain
    5.  $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K_t}\mathbf{v}$ // update of the mean vec.
    6.  $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K_t}\mathbf{H_t})\bar{\boldsymbol{\Sigma}}_t$ // update of the cov.
```

To calculate the inverse of the measurement uncertainty matrix, $\mathbf{S_t}$; Cholesky decomposition can be used as in [18].

The optimal complexity of the inverse calculation of a 2*N*x2*N* matrix is *$O((2N)^{2.4})$*; hence the computational complexity of the $\mathbf{S_t}$'s inversion operation is *$O(N^{2.4}x2^{2.4})$* in the batch update for whole observations.

IV. Matrix inversion by Jacobi iteration Method

The linear equation system, which has *N* variables and *N* equations, is given in (9) [19]:

$$\mathbf{Ax} = \mathbf{b} \tag{9}$$

The solution of (9) is obtained as,

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}\,. \tag{10}$$

If **b** is chosen as an unit matrix $\mathbf{I}$, the solution of (9) is the inverse of **A.** Due to the computational load involved in taking the inverse of **A,** rearrange (9) as

$$(\mathbf{D} + \mathbf{L} + \mathbf{U}+)\mathbf{x} = \mathbf{I}\,, \tag{11}$$

where **D**, **L** and **U** are the diagonal matrix, lower and upper triangular matrices respectively, yields

$$\mathbf{x} = \mathbf{D}^{-1}\left[(\mathbf{D} - \mathbf{A})\mathbf{x} + \mathbf{I}\right]. \tag{12}$$

Now assign initial matrix $\mathbf{x}_0$ to the $\mathbf{x}$ on the right-hand side and calculate $\mathbf{x}_1$ on the left-hand side. in general case, we can write

$$\mathbf{x}_{k+1} = \mathbf{D}^{-1}\left[(\mathbf{D} - \mathbf{A})\mathbf{x}_k + \mathbf{I}\right]. \tag{13}$$

If **A** is a dominant matrix, the absolute values of the eigenvalues of the matrix $\mathbf{D^{-1}(D - A)}$ are less than 1. When this condition exists, $\mathbf{x}_k$ approaches the inverse of **A**. A dominant matrix has the absolute value of the diagonal entry in a row which is larger than or equal to the sum of the absolute values of all the other entries in that row. If that condition is satisfied, inverse of the matrices, which have even large size; can be calculated in a few iterations by using Jacobi iteration method [19].

The measurement uncertainty matrices, those are generated in this study, are symmetric and positive-definite matrixes. They also provide the above conditions generally. The sufficient iteration number is *M* for an *MxM* matrix to converge the solution by Jacobi iteration method in this study.

If Jacobi iteration method is used for the matrix inversion in the single update, the complexity is O(2*N*) where *N* is the number of the associated measurements. *O(2)* is the complexity of calculating the inverse of a 2x2 matrix by Jacobi iteration method. This is a bit smaller than the complexity of the naive single update, it is not significant. However in the naive batch update for all of the measurements, calculating the inverse of the measurement uncertainty matrix costs O($N^{2.4}$x$2^{2.4}$); while it costs O(2*N*) by using Jacobi iteration method for matrix inversion in the batch update. It is significantly less than the complexity of naive batch update.

All of the operations, except the inverse matrix calculation, are the same in the naive update approaches and the approaches which use Jacobi iteration method for inverse matrix calculation. The complexity results for both single and batch update are the same, O(2*N*), when Jacobi iteration method is used for matrix inversion. The difference between the single updates, which uses and does not use Jacobi iteration method, is not significant. Hence the results of the single update that uses Jacobi iteration method are not given in the experimental results.

TABLE I
COMPLEXITY ANALYSIS OF THE APPROACHES

Approach	Single Cholesky update	Batch Cholesky update	Batch update that uses Jacobi iteration method for matrix inverse
Complexity	O($2^{2.4}$xN)	O($M^{2.4}$)=O($2^{2.4}$x$N^{2.4}$)	O(M) = O(2xN)

The computational complexity of the methods are given in Table I when the number of the associated landmarks is N and $M = 2xN$.

V. EXPERIMENTAL RESULTS

Experiments are performed by using SLAM software of Bailey and Nieto [18]. Single and batch updates, which use Cholesky update, are run 10 times. Batch update, which uses Jacobi iteration for inverse matrix calculation, is also run 10 times. Averages of the Root Mean Square Errors (RMSE) for robot position during the exploration of each approach are calculated and given in this section.

The inverse matrix calculation of the interested matrix by Jacobi iteration could not converged in all the experiments. It is observed that, the solution obtained by Jacobi iteration method converged to matrix inverse and the filter is concluded successfully with the ratio of 8/10 (in terms of experiment numbers); it diverged with the ratio 2/10. When the filter, which uses Jacobi iteration for matrix inversion, diverged; it was run more than 10 experiments to acquire 10 successful experiment results.

Maximum likelihood data association is used in the application.

The environment, that the experiments are implemented in, is shown in the Fig.1. The blue stars are landmarks, green dots are waypoints those are used for calculate steering angle.

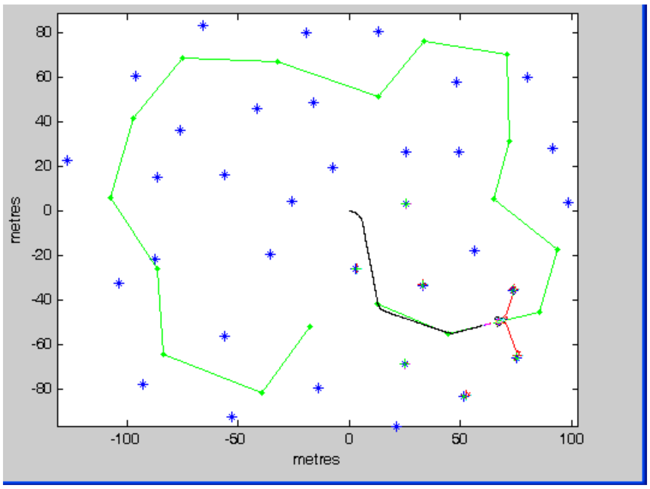

Fig. 1. Used environment in the experiments.

RMSE results of the robot position with respect to approaches are given in Table II. Robot closed the loop once in the environment.

TABLE II
RMSE RESULTS OF THE APPROACHES FOR ONE LOOP IN THE ENVIRONMENT

Approach	Single Cholesky update	Batch Cholesky update	Batch update that uses Jacobi method for matrix inverse
RMSE	4.68184	3.91332	3.21603

From the results of the experiments, it is clear that, the minimum RMSE result is achieved by the batch update approach which use Jacobi iteration method for inverse matrix calculation. Although inverse matrix calculation by Jacobi iteration does not converged every time, it is better than a commonly chosen approach, Cholesky update, when it converges.

The effect of the number of loop closings in the environment to RMSE results of the approaches is also examined. Hence Jacobi iteration method is decided as whether it is confident or not, when the number of the loop closing increases. When robot closed the loop twice in the environment, the RMSE results of the robot position with respect to approaches are given in Table III.

TABLE III
RMSE RESULTS OF THE APPROACHES FOR TWO LOOPS IN THE ENVIRONMENT

Approach	Single Cholesky update	Batch Cholesky update	Batch update that uses Jacobi method for matrix inverse
RMSE	3.9766	3.7661	3.0791

The RMSE value for each approach decreases when the number of loop closings is two in the environment. Batch update approach, that uses Jacobi iteration, is obviously seen as the best resulting approach in the terms of RMSE of the robot position.

VI. CONCLUSION

In this study, the computational complexity of the calculation of the measurement uncertainty matrix’s inverse, which is done in the measurement update process of EKF-SLAM, is examined. That computational complexity is decreased via using Jacobi iteration method for inverse matrix calculation.

The approach, that uses Jacobi iteration method for inverse matrix calculation, is more accurate and generates less RMSE value than the approaches which use Cholesky single and batch updates, with respect to experimental results. Furthermore, each approach generates less RMSE value of the robot position, when the number of the loop closings in the environment increases. In this case, the approach that uses Jacobi iteration method also gives the least RMSE results. However, the solution obtained by Jacobi iteration method cannot converge to the interested matrix inverse for each experiment because the interested spectral radius is not less than 1 or the interested matrix is not a dominant matrix.

Using Jacobi iteration method in a system model, which provides the convergence conditions for each iteration, can significantly decrease the computational complexity of that system.

REFERENCES

[1] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, "A Framework for Vision Based Bearing Only 3D SLAM", IEEE International Conference on Robotics and Automation (ICRA), pp. 1944-1950, 2006.

[2] T. Lemaire and S. Lacroix, (2007), "SLAM with Panoramic Vision", Journal of Field Robotics, Vol. 24, no. 1-2, pp. 91 - 111, Feb 2007.

[3] J.-H. Kim, and M.J. Chung, "SLAM with omni-directional stereo vision sensor" In Proceedings of the International Conference on Intelligent Robots and Systems, Vol. 51, no. 1, pp. 31-43, January, 2008.

[4] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez and M. Olivares, "Visual 3-D SLAM from UAVs", Journal of Intelligent and Robotic Systems, Vol. 55 (4-5), pp. 299-321, Aug 2009.

[5] D. Scaramuzza, R. Siegwart and A. Martinelli, (2009), "A Robust Descriptor for Tracking Vertical Lines in Omnidirectional Images and Its Use in Mobile Robotics", International Journal of Robotics Research, Vol. 28, no. 2, pp. 149-171, Feb 2009.

[6] J. Sola, A. Monin, M. Devy and T. Vidal-Calleja, “Fusing Monocular Information in Multicamera SLAM”, IEEE Transactions on Robotics, Vol. 24, no. 5, pp. 958-968, 2008.

[7] A.P. Gee, D. Chekhlov, A. Calway and W. Mayol-Cuevas, "Discovering Higher Level Structure in Visual SLAM", IEEE Transactions on Robotics, Vol. 24, no. 5, pp. 980–990, October 2008.

[8] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, The MIT Press, 2005.

[9] P. Abbeel, A. Coates, M. Montemerlo, A. Ng, and S. Thrun, "Discriminative training of Kalman filters". In Proceedings of Robotics Science and Systems, Cambridge, MA, 2005. MIT Press.

[10] Q.-H. Meng, Y.-C. Sun, and Z.-L. Cao, “Adaptive Extended Kalman Filter (AEKF)-based Mobile Robot Localization Using Sonar”, Journal of Robotica, Vol 18(5), pp. 459–473, September 2000.

[11] S. J. Julier and J. K. Uhlmann, ”Unscented filtering and nonlinear estimation,” Proceedings of the IEEE, Vol. 92, No. 3, March 2004, pp401-422.

[12] R. Martinez-Cantin and J. A. Castellanos, “Unscented slam for large-scale outdoor environments”, IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems, 2005, pp 328-333.

[13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem", In the Proceedings of the AAAI National Conference of Artificial Intelligence, Canada, 2002.

[14] C. Stachniss, D. Hahnel, and W. Burgard, W., “Exploration with active loop-closing for FastSLAM”, Proceedings of International Conference on Intelligent Robots and Systems (IROS), pp. 1505-1510, Oct 2004.

[15] J. Z. Sasiadek, A. Monjazeb and D. Necsulescu, “Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM”, 16th Mediterranean Conference on Control and Automation, pp. 517 - 522, 25-27 June 2008, Ajaccio, France.

[16] M. Montemerlo, and S. Thrun , “Simultaneous localization and mapping with unknown data association using FastSLAM”, in the Proceedings of the 2003 IEEE International Conference on Robotics & Automation (ICRA), pp. 1985 - 1991, 14-19 September 2003, Taipei, Taiwan.

[17] T. Bailey, J. Nieto and E. Nebot, “Consistency of the FastSLAM algorithm”, IEEE International Conference on Robotics and Automation (ICRA), pp. 424 - 429, 15-19 May 2006, Orlando, FL.

[18] http://www-personal.acfr.usyd.edu.au/tbailey/software/index.html

[19] J. M. Ortega, Numerical analysis: a second course, SIAM, 1990.