Numerical Issues and Approximated Models for the Diagnosis of Transmission Pipelines

Zdzisław Kowalczuk and Marek Tatara

Abstract The chapter concerns numerical issues encountered when the pipeline flow process is modeled as a discrete-time state-space model. In particular, issues related to computational complexity and computability are discussed, i.e., simulation feasibility which is connected to the notions of singularity and stability of the model. These properties are critical if a diagnostic system is based on a discrete mathematical model of the flow process. The starting point of the study is determined by the partial differential equations obtained from the momentum and mass conservation laws by using physical principles. A realizable computational model is developed by approximation of the principal equations using the finite difference method. This model is expressed in terms of the recombination matrix A which is the key of the analysis by taking into account its possible singularity and stability. The nonsingularity of the matrix A for nonzero and finite, time and spatial steps is proven by the Lower-Upper decomposition. A feature of the discrete model allows the derivation of a nonsingular aggregated model, whose stability can be analyzed. By considering the Courant-Friedrichs-Lewy condition and data from experimental studies, numerical stability conditions are derived and limitations for the feasible discretized grid are obtained. Moreover, the optimal relationship between the time and space steps which ensures a maximum stability margin is derived. Because the inverse of matrix A, composed of four tridiagonal matrices, is required for the main diagnosis methods, two analytical methods for the inversion are discussed which reduce the system's initialization time and allow designing an accurate and fast diagnosis algorithm. By considering that each inversion method generates its particular structure, two different flow models are generated: one based on auxiliary variables and the other suitable if the stability condition of A is satisfied. The applicability of the two models is shown by considering the norm of the difference between their behaviors for a finer discretization grid. A similarity measure is proposed which considers the number of pipeline segments as well as the ratio between the time and spatial steps. Thus, the system's computational efficiency is improved and satisfactory results are shown with respect to the base model, if a highly dimensional model with the approximated diagonal matrix is considered.

3.1 Introduction

Model-based approaches for leak detection and identification (LDI) require a robust mathematical description of the flow process with the same behavior as the real pipeline. Throughout the years, many modeling methodologies have been developed for the leak detection, monitoring of parameters, and complexity analysis of pipeline installations. The dominant directions are the (linear) observer-based methods for leaks in a single pipeline (Billmann and Isermann, 1987; Kowalczuk and Gunawickrama, 1998, 2004) or multiple leaks (Torres et al., 2012), as well as the artificial neural network approaches (Belsito et al., 1998). Recently, Reddy et al. (2011) and Verde and Torres (2015) have reported efforts to describe complex pipeline systems.

Behind every discrete-time model derived for LDI purposes, however, there is a concern about the numerical stability and computability of the model. The specific practical issues relate to computational complexity and computability, therefore, the feasibility of simulation is discussed in this chapter. Section 3.2 starts with the principal (physical) equations of the continuous-time nature, derived from the momentum and mass conservation laws. The feasible models are developed by discretization of the principal equations using the finite difference method, and are analyzed considering possible singularity and stability. In Section 3.3, by LU decomposition, the proof of the nonsingularity of the recombination matrix for nonzero and finite time and spatial steps is presented. Once the invertibility of the matrix is proved, in Section 3.4 a new aggregated model, named nonsingular, is derived and its features are discussed. In Section 3.5 the question of the numerical stability of the model is analyzed. By considering the Courant-Friedrichs-Lewy condition and data obtained from experimental studies, one derives a necessary condition for numerical stability that limits the choice of the discretization grid. This allows the search for the optimal relationship between the steps of time and space, which ensures a maximum stability margin.

In Section 3.6, two analytical methods for the inversion of matrix A composed of four tridiagonal matrices are introduced, and the utility of the proposed models for diagnostic systems is discussed in Section 3.7. The analysis is based on the norm of the difference between both models and considers the cardinality of the pipeline segmentation as well as the ratio between the time and spatial steps.

3.1.1 Matrices' Notations

Symmetric matrix: A matrix M is symmetric if it is square and equal to its transposition: $M = M^T$.
Diagonal matrix: A matrix is diagonal if its elements out of the main diagonal are all zero.
Diagonal of a matrix: The set of elements of a matrix located in the diagonal, i.e., the elements set $m_{i,j}$ with $i = j$.
Subdiagonal of a matrix: The set of elements of a matrix directly under the main

diagonal (it is thus the first diagonal under the main diagonal).
Superdiagonal of a matrix: The set of elements of a matrix directly above the main diagonal (it is thus the first diagonal above the main diagonal).
Tridiagonal matrix: A matrix is tridiagonal if it has nonzero elements only on its main diagonal, subdiagonal, and superdiagonal.
Centrosymmetric matrix: A matrix M is called centrosymmetric if its elements satisfy the following condition:

$$m_{i,j} = m_{K-i+1,K-j+1} \qquad \text{for} \qquad i,j \in \{1,2,...,K\}$$

where K is the number of the matrix rows (or columns, since it is square). In other words, such matrix is symmetric around its center.
Row of a matrix: $r_i(M)$ denotes the *i-th* row of matrix M.

3.2 Base Model of the Flow Process

Consider the mathematical description of the pressure and flow rate for a flow process in transmission pipelines, which is expressed by the two equations obtained from the momentum and mass conservation laws (Billmann and Isermann, 1987):

$$\frac{\mathscr{A}}{v^2}\frac{\partial p}{\partial t} + \frac{\partial q}{\partial z} = 0 \tag{3.1}$$

$$\frac{1}{\mathscr{A}}\frac{\partial q}{\partial t} + \frac{\partial p}{\partial z} = -\frac{\lambda v^2}{2D\mathscr{A}^2}\frac{q|q|}{p} - \frac{gsin\alpha}{v^2}p \tag{3.2}$$

where $\mathscr{A}$ is the cross-sectional area $[m^2]$, v is the isothermal velocity of the sound in the fluid $[\frac{m}{s}]$, D is the diameter of the pipe $[m]$, q is the mass flow $[\frac{kg}{s}]$, p is the pressure $[Pa]$, t is the time $[s]$, z is the spatial coordinate $[m]$, λ is the dimensionless generalized friction factor, α is the inclination angle $[rad]$, and g is the gravitational acceleration $[\frac{m}{s^2}]$.

Since the practical operation of a model-based algorithm for pipeline diagnosis requires emulation of the underlying process behavior, the presented set of equations is discretized for the numerical implementation. The idea behind the discretization is to divide the pipeline into N segments of equal size, each one of length Δz, where the pressure at the end of each odd segment and the flow rate at the end of each even segment characterize the flow process. It is assumed that both variables, flow and pressure, are measured at the inlet and outlet of the pipeline. Such a discretization scheme is illustrated in Fig. 3.1.

The discrete-time model is simply obtained by introducing low-order central difference schemes:

$$\frac{\partial x}{\partial t} = \frac{3x_d^{k+1} - 4x_d^k + x_d^{k-1}}{2\Delta t} \tag{3.3}$$

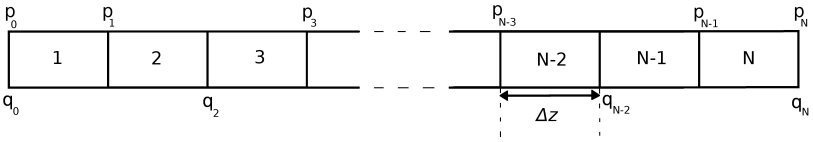


Fig. 3.1: Discretization scheme of a pipeline with N even segments

$$\frac{\partial x}{\partial z} = \frac{x_{d+1}^{k+1} - x_{d-1}^{k+1} + x_{d+1}^{k} - x_{d-1}^{k}}{4\Delta z} \tag{3.4}$$

where Δz is a spatial step, and Δt is a time step, subscripts and superscripts denote the number of the pipeline's segment and discrete-time step index, respectively.

The substitution of (3.3) and (3.4) into the composed model (3.1) and (3.2) gives the following discretized set of equations for the flow process in the pipeline:

$$a p_d^{k+1} - b\left(q_{d-1}^{k+1} - q_{d+1}^{k+1}\right) = \frac{a}{3}\left(4p_d^k - p_d^{k-1}\right) + b\left(q_{d-1}^k - q_{d+1}^k\right) \tag{3.5}$$

$$b\left(p_{d+1}^{k+1} - p_{d-1}^{k+1}\right) + c q_d^{k+1} = b\left(p_{d-1}^k - p_{d+1}^k\right) + Y_d p_d^k + \left(\frac{4c}{3} + F_d^k\right) q_d^k - \frac{c}{3} q_d^{k-1} \tag{3.6}$$

with physical coefficients

$$a = \frac{3\mathscr{A}}{2\nu^2 \Delta t},\ b = \frac{1}{4\Delta z},\ c = \frac{3}{2\mathscr{A}\Delta t},\ Y_d = \frac{g sin\alpha_d}{\nu^2}$$

where α_d denotes the inclination angle of a d-th segment. The nonlinear function is approximated by

$$F_d^k \simeq -\frac{\lambda \nu^2}{D\mathscr{A}} \frac{|q_d^k|}{p_{d-1}^k + p_{d+1}^k}$$

since pressure is monitored at the ends of odd segments only. At the inlet and outlet ($d = 0$ and $d = N$) of the pipeline, the approximation is not required, because the pressure is known or measured, that is $F_d^k = -\frac{\lambda \nu^2}{2D\mathscr{A}} \frac{|q_d^k|}{p_d^k}$.

Thus, (3.5) and (3.6) can be represented by the compact state-space model

$$A\hat{x}^k = B\hat{x}^{k-2} + C\left(\hat{x}^{k-1}\right)\hat{x}^{k-1} + Du^{k-1} + Eu^k \tag{3.7}$$

where B and $C(\hat{x}^k - 1)$ are associated with the nonlinear dynamic of the state

$$\hat{x}^k = \left[\, q_0^k\ q_2^k\ q_4^k \cdots q_N^k \,\middle|\, p_1^k\ p_3^k\ p_5^k \cdots\ p_{N-1}^k \,\right]^T \in \mathbb{R}^{N+1}$$

and matrices D and E are associated with the input $u^k = \left[\, p_0^k\ p_N^k \,\right]^T \in \mathbb{R}^2$ (Gunawickrama, 2001). In particular, the matrices are given by

$$B = \frac{1}{3}\left[\begin{array}{ccccc:cccccc} -c & 0 & \cdots & 0 & 0 & & & & & \\ 0 & -c & \cdots & 0 & 0 & & & & & \\ \vdots & & \ddots & & \vdots & & & 0_{(\frac{N}{2}+1)\times(\frac{N}{2})} & & \\ 0 & 0 & \cdots & -c & 0 & & & & & \\ 0 & 0 & \cdots & 0 & -c & & & & & \\ \hdashline & & & & & -a & 0 & \cdots & 0 & 0 \\ & & & & & 0 & -a & \cdots & 0 & 0 \\ & & 0_{(\frac{N}{2})\times(\frac{N}{2}+1)} & & & \vdots & & \ddots & & \vdots \\ & & & & & 0 & 0 & \cdots & -a & 0 \\ & & & & & 0 & 0 & \cdots & 0 & -a \end{array}\right] \in \mathbb{R}^{(N+1)\times(N+1)} \tag{3.8}$$

$$C(\hat{x}^{k-1}) = \left[\begin{array}{ccccc:cccccc} H_0^k & 0 & \cdots & 0 & 0 & 2b & 0 & \cdots & 0 & 0 \\ 0 & H_2^k & \cdots & 0 & 0 & \Gamma_{2+} & \Gamma_{2-} & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & H_{N-2}^k & 0 & 0 & 0 & \cdots & \Gamma_{(N-2)+} & \Gamma_{(N-2)-} \\ 0 & 0 & \cdots & 0 & H_N^k & 0 & 0 & \cdots & 0 & +2b \\ \hdashline b & -b & 0 & \cdots & 0 & \frac{4a}{3} & 0 & \cdots & 0 & 0 \\ 0 & b & -b & \cdots & 0 & 0 & \frac{4a}{3} & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ 0 & \cdots & b & -b & 0 & 0 & 0 & \cdots & \frac{4a}{3} & 0 \\ 0 & \cdots & 0 & b & -b & 0 & 0 & \cdots & 0 & \frac{4a}{3} \end{array}\right] \in \mathbb{R}^{(N+1)\times(N+1)} \tag{3.9}$$

where $\Gamma_{d\pm} = \frac{Y_d}{2} \pm b$, $H_0^k = \frac{4c}{3} - \frac{\lambda v^2 \hat{x}^k_{(d/2+1)}}{2D\mathscr{A}u_1^k}$, $H_N^k = \frac{4c}{3} - \frac{\lambda v^2 \hat{x}^k_{(d/2+1)}}{2D\mathscr{A}u_2^k}$,

$$H_d^k = \frac{4c}{3} - \frac{\lambda v^2 \hat{x}^k_{(d/2+1)}}{D\mathscr{A}\left(\hat{x}^k_{(d/2+1+N/2)} + \hat{x}^k_{(d/2+2+N/2)}\right)} \text{ for } d = 2,4,...,N-2$$

.

$$D = \left[\begin{array}{cc} Y_0 + 2b & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & Y_N - 2b \\ \hdashline & \\ 0_{(\frac{N}{2})\times 2} & \end{array}\right] \in \mathbb{R}^{(N+1)\times 2}, \qquad E = \left[\begin{array}{cc} 2b & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & -2b \\ \hdashline & \\ 0_{(\frac{N}{2})\times 2} & \end{array}\right] \in \mathbb{R}^{(N+1)\times 2} \tag{3.10}$$

One can see from (3.7) that the singularity of A affects the model. This description is called the *singular state-space* model. The term *singular* emphasizes the specific form of the model in which A may be not invertible in general. The matrix A itself is named recombination matrix, and it is written by

$$A = \left[\begin{array}{c:c} A_1 & A_2 \\ \hdashline A_3 & A_4 \end{array}\right] = \left[\begin{array}{cccccc:ccccc} & & & & & & \searrow 2b & 0 & \cdots & 0 & 0 \\ & & & & & & -b & \searrow b & \cdots & 0 & 0 \\ & & & & & & \vdots & & \ddots & & \vdots \\ & & & D_{\frac{N}{2}+1}(c) & & & 0 & 0 & \cdots & \searrow b & 0 \\ & & & & & & 0 & 0 & \cdots & -b & \searrow b \\ & & & & & & 0 & 0 & \cdots & 0 & -2b \\ \hdashline -b & b & 0 & \cdots & 0 & 0 & 0 & & & & \\ 0 & -b & b & \cdots & 0 & 0 & 0 & & & & \\ \vdots & & & \ddots & & & \vdots & & D_{\frac{N}{2}}(a) & & \\ 0 & 0 & 0 & \cdots & -b & b & 0 & & & & \\ 0 & 0 & 0 & \cdots & 0 & -b & b & & & & \end{array}\right] \in \mathbb{R}^{(N+1)\times(N+1)} \tag{3.11}$$

where $D_W(\theta) \in \mathbb{R}^{W\times W}$ denotes a diagonal matrix with θ on the diagonal. Note that the upper right submatrix is non-square and belongs to $\mathbb{R}^{(\frac{N}{2}+1)\times(\frac{N}{2})}$.

3.3 Assessment of the Model's Singularity

According to Kowalczuk and Tatara (2013), to determine the effect of the selected discretization grid on the singularity in the model, the determinant of the recombination matrix A can be calculated by using the matrix LU factorization. To achieve this, the augmented recombination matrix is defined as

$$\bar{A} = \left[\, I_{N+1} \,\middle\vdots\, A \,\right] = \left[\, L \,\middle\vdots\, U \,\right] \tag{3.12}$$

where $I_{N+1} \in \mathbb{R}^{(N+1)\times(N+1)}$ is the proper identity matrix. The key to obtaining the determinant of $\bar{A}$ is the calculation of the lower and upper triangular matrices on the left-hand and right-hand sides of the augmented recombination matrix, respectively. Thus, the determinant is reduced to the product of the diagonal values of the matrices L and U (Kreyszig, 2006):

$$\det(A) = \det(L)\ \det(U) \tag{3.13}$$

To obtain the lower triangular matrix on the left-hand side of matrix

$$\bar{A} = \left[\begin{array}{c:cccccccccc} & c & 0 & \cdots & 0 & 0 & 2b & 0 & \cdots & 0 & 0 \\ & 0 & c & \cdots & 0 & 0 & -b & b & \cdots & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ & 0 & 0 & \cdots & c & 0 & 0 & 0 & \cdots & -b & b \\ & 0 & 0 & \cdots & 0 & c & 0 & 0 & \cdots & 0 & -2b \\ \mathbb{I}_{N+1} & -b & b & 0 & \cdots & 0 & a & 0 & \cdots & 0 & 0 \\ & 0 & -b & b & \cdots & 0 & 0 & a & \cdots & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ & 0 & \cdots & -b & b & 0 & 0 & 0 & \cdots & a & 0 \\ & 0 & \cdots & 0 & -b & b & 0 & 0 & \cdots & 0 & a \end{array}\right] \tag{3.14}$$

elementary row transformations are used.

The first elementary transformations of $\bar{A}$ are obtained by adding row $r_1\left(\bar{A}\right)$ multiplied by $d = \frac{b}{c}$ to the row $r_{(\frac{N}{2}+2)}\left(\bar{A}\right)$ and then by subtracting row $r_2\left(\bar{A}\right)$ multiplied by d from row $r_{(\frac{N}{2}+2)}\left(\bar{A}\right)$. This iterative operation should be repeated by increasing the rows' indexes each time by 1, until they reach $\frac{N}{2}$. Thus, the equivalent matrix can be written by

$$\bar{A} = \left[\begin{array}{ccccc:c:ccccc} & & & & & & 2b & 0 & \cdots & 0 & 0 \\ & & & & & & -b & b & \cdots & 0 & 0 \\ & I_{\frac{N}{2}+1} & & 0_{(\frac{N}{2}+1)\times\frac{N}{2}} & & D_{\frac{N}{2}+1}(c) & \vdots & & \ddots & & \vdots \\ & & & & & & 0 & 0 & \cdots & -b & b \\ & & & & & & 0 & 0 & \cdots & 0 & -2b \\ \hdashline l_{n,1} & \cdots & 1 & \cdots & 0 & & a+3bd & -bd & \cdots & 0 & 0 \\ & & & & & & -bd & a+2bd & -bd & \cdots & 0 \\ \vdots & & \ddots & & \vdots & 0_{\frac{N}{2}} & \vdots & & \ddots & & \vdots \\ l_{N,1} & l_{N,2} & \cdots & 1 & 0 & & 0 & \cdots & -bd & a+2bd & -bd \\ l_{N+1,1} & l_{N+1,2} & \cdots & l_{N+1,N} & 1 & & 0 & 0 & \cdots & -bd & a+3bd \end{array}\right], \tag{3.15}$$

where the coefficients $l_{j,k}$ for $j = \frac{N}{2}+2, \frac{N}{2}+3, ..., N+1$ and $k = 1, 2, ..., N$ are irrelevant for the determinant of A.

On the other hand, the upper triangular matrix on the right-hand side can be obtained by removing the coefficients $-bd$ of the subdiagonal on the right-hand matrix by starting with $n = \frac{N}{2}+2$ and by ending at $n = N$. This is achieved by adding to the row $r_{n+1}\left(\bar{A}\right)$ the row $r_n\left(\bar{A}\right)$ multiplied by $\frac{bd}{f_n}$, with f_n being the n-th coefficient of the main diagonal of the right-hand side matrix. Thus, (3.15) is equivalent to

$$\bar{A} = \left[\begin{array}{ccccc:c:ccccc} & & & & & & 2b & 0 & \cdots & 0 & 0 \\ & & & & & & -b & b & \cdots & 0 & 0 \\ & I_{\frac{N}{2}+1} & & 0_{(\frac{N}{2}+1)\times\frac{N}{2}} & & D_{\frac{N}{2}+1}(c) & \vdots & & \ddots & & \vdots \\ & & & & & & 0 & 0 & \cdots & -b & b \\ & & & & & & 0 & 0 & \cdots & 0 & -2b \\ \hdashline l'_{n,1} & \cdots & 1 & \cdots & 0 & & f_n & u_{n,n+1} & \cdots & 0 & 0 \\ & & & & & & 0 & f_{n+1} & u_{n+1,n+2} & \cdots & 0 \\ \vdots & & \ddots & & \vdots & 0_{\frac{N}{2}} & \vdots & & \ddots & & \vdots \\ l'_{N,1} & l'_{N,2} & \cdots & 1 & 0 & & 0 & \cdots & 0 & f_N & u_{N,N+1} \\ l'_{N+1,1} & l'_{N+1,2} & \cdots & l'_{N+1,N} & 1 & & 0 & 0 & \cdots & 0 & f_{N+1} \end{array}\right] \tag{3.16}$$

The coefficients f_j for $j = \frac{N}{2}+2, \frac{N}{2}+3, ..., N+1$ are obtained recursively with the initial condition $f_{\frac{N}{2}+2} = a+3bd$ and the final condition $f_{N+1} = a+3bd-\frac{b^2d^2}{f_N}$. The rest of coefficients are obtained with

$$f_j = a+2bd-\frac{b^2d^2}{f_{j-1}} \quad \text{for} \quad j = \frac{N}{2}+3, \frac{N}{2}+4, ..., N$$

Note that coefficients $l'_{j,k}$ for $j = \frac{N}{2}+2, \frac{N}{2}+3, \ldots N+1$ and $k = 1, 2, ..., N$ under the main diagonal of the left-hand side matrix have no effect on the $\det(\bar{A})$. Similarly coefficients $u_{j,k}$ for $j = n, ..., N$ and $k = j+1$ above the main diagonal of the right-hand side matrix have no effect on the $\det(\bar{A})$. As an example, if $N = 2$, there exists only one coefficient $f_3 = a+4bd$.

As a consequence, the $\det(A)$ is given by

$$\det(A) = \prod_{j=1}^{N+1} l_{j,j}u_{j,j} \tag{3.17}$$

which is generated by the diagonal coefficients of L and U, respectively. Since the diagonal elements of L are equal to 1, and U consists of $\frac{N}{2}+1$ diagonal elements with value c and $\frac{N}{2}$ diagonal elements of value f_j, the expression (3.17) is reduced to

$$\det(A) = (c)^{\frac{N}{2}+1} \prod_{j=\frac{N}{2}+2}^{N+1} f_j \tag{3.18}$$

and the general expression for a dimension N is written by

$$\det(A) = \sum_{i=0}^{\frac{N}{2}} C_i^N a^{\frac{N}{2}-i} b^{2i} c^{\frac{N}{2}-i+1} \tag{3.19}$$

with boundary values $C_0^N = 1$, for $N = 0, 2, 4, ...$ and $C_{\frac{N}{2}}^N = 2N$, for $N = 2, 4, 6, ...$ The coefficients C_i^N can be recursively calculated as

$$C_i^N = 2C_{i-1}^{N-2} + C_i^{N-2}, \quad \text{for } N = 4,6,8,\ldots \text{ and } i = 1 \tag{3.20}$$

$$C_i^N = 2C_{i-1}^{N-2} + C_i^{N-2} - C_{i-2}^{N-4}, \quad \text{for } N = 4,6,8,\ldots; \text{and } i = 2,3\ldots, \frac{N}{2} - 1 \tag{3.21}$$

The determinant values for $N = 2,4,6$ can be obtained by substituting the values of f_j into (3.18) and are reported in Table 3.1.

Table 3.1: Determinants of the recombination matrix for selected segmentation N

N	Determinant of matrix A
2	$ac^2 + 4b^2c$
4	$a^2c^3 + 6ab^2c^2 + 8b^4c$
6	$a^3c^4 + 8a^2b^2c^3 + 19ab^4c^2 + 12b^6c$

By analyzing (3.20) and (3.21), one can see that all of the coefficients are positive for $i > 0$ and $N > 0$. Moreover, since (3.19) depends on a, b, and c, with even exponentials, the only way to get a zero determinant is that any of them is equal to zero. By considering that the cross-section and the sound velocity are always positive, the only way to achieve singularity of the recombination matrix is if $\Delta z = \infty$ or $\Delta t = \infty$. The former is equivalent to an infinitely long pipeline, and the latter can be neglected by maintaining the numerical stability of the algorithm discussed in Section 3.5.

Note that the time and spatial steps are positive, and negative spatial steps could not affect the non singularity of A, since b always has an even exponent. Negative time steps also do not appear in the practical flow process. As a consequence, one can establish that the recombination matrix A is nonsingular for every finite time and spatial steps, and thus, invertible.

3.4 Aggregated Model

Once the invertibility of A has been shown, the base model (3.7) can be represented in the nonsingular state-space form

$$\hat{x}^k = A^{-1}\left(B\hat{x}^{k-2} + C\left(\hat{x}^{k-1}\right)\hat{x}^{k-1} + Du^{k-1} + Eu^k\right) \tag{3.22}$$

According to Kowalczuk and Tatara (2013), by defining an aggregated state vector $\tilde{x}^k = \left[\hat{x}^{k\,T}\ \hat{x}^{k-1\,T}\right]^T$ and an augmented input vector, $\tilde{u}^k = \left[u^{k\,T}\ u^{k-1\,T}\right]^T$, the flow process model can be rewritten in the form of state-space equation:

$$\tilde{x}^k = A_c\tilde{x}^{k-1} + B_c\tilde{u}^k \tag{3.23}$$

where

$$A_c = \left[\begin{array}{c|c} A^{-1}C\left(\tilde{x}^{k-1}\right) & A^{-1}B \\ \hline I & 0 \end{array}\right] \tag{3.24}$$

$$B_c = \left[\begin{array}{c|c} A^{-1}E & A^{-1}D \\ \hline 0 & 0 \end{array}\right] \tag{3.25}$$

Note that matrix A_c is a function of the state vector $\tilde{x}^{k-1}$, and matrix B_c is dependent on the friction factor λ. Thus, A_c must be recalculated during the operation of the algorithm. Since (3.23) represents a regular state-space form, it can be analyzed with well-known methods from control theory, but the recombination matrix A must be inverted at each Δt. For this task, the following matrix inversion lemma (MIL) can be applied at each time step (Brogan, 1991):

$$A^{-1} = \left[\begin{array}{c|c} A_1' & A_2' \\ \hline A_3' & A_4' \end{array}\right] = \left[\begin{array}{c|c} (A_1 - A_2A_4^{-1}A_3)^{-1} & -A_1^{-1}A_2(A_4 - A_3A_1^{-1}A_2)^{-1} \\ \hline -A_4^{-1}A_3(A_1 - A_2A_4^{-1}A_3)^{-1} & (A_4 - A_3A_1^{-1}A_2)^{-1} \end{array}\right] \tag{3.26}$$

3.5 Selection of the Discretization Grid

The numerical stability problem in discretized differential equations is connected with the choice of discretization grid, which determines information's propagation speed between the nodes of the algorithm. According to Strikwerda (2007), the Courant-Friedrichs-Lewy condition (CFL) establishes that the propagation speed of information in a numerical algorithm $\frac{\Delta z}{\Delta t}$ must be greater or equal to the information exchange velocity in the corresponding differential equations for maintaining stability. This is only a necessary condition, but it is not sufficient and gives only a lower boundary for the velocity. At the same time, discretization must allow online simulation. Since the highest speed present in the flow process is the sound velocity v, the following inequality must be satisfied:

$$\frac{\Delta z}{\Delta t} \geqslant v \tag{3.27}$$

In addition, rewriting it as an equality, one obtains

$$\Delta t = \mu \frac{\Delta z}{v} \tag{3.28}$$

where μ is a coefficient within the range $\langle 0, 1\rangle$, binding the discretization steps.

According to Kowalczuk and Tatara (2016), there exists a coefficient μ_{opt}, which maximizes the stability margin s_m for the discrete-time system on the z-plane (domain of the Z-transform) for the specific physical parameters of a pipeline. The stability margin is calculated as

$$s_m = 1 - e_{max} \tag{3.29}$$

where e_{max} is the largest absolute eigenvalue of the state transition matrix A_c.

The distribution of the stability margin versus the *stability tuner*, μ, for a few values of a pipeline's length is shown in Fig. 3.2.

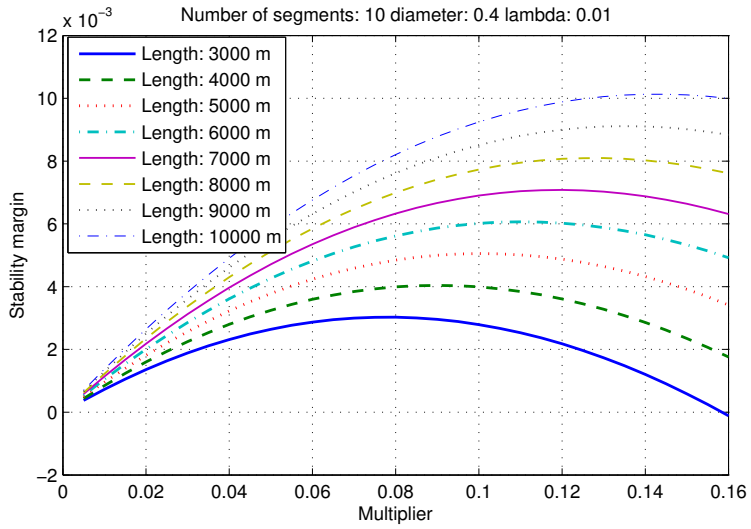


Fig. 3.2: Distribution of the stability margin versus coefficient μ for eight lengths of the pipeline shown in $[m]$. Experimental setup: $N = 10$, $D = 0.4$ $[m]$, $\lambda = 0.01$, $\nu = 304$ $[\frac{m}{s}]$, $p_{inlet} = 3.2$ $[MPa]$, and $p_{outlet} = 3.0$ $[MPa]$

The eigenvalues are calculated after the transition response to diminish the influence of initial values on the stability margin. As one can see from Fig. 3.2, the stability margin is in the magnitude order of 10^{-2}, which emphasizes the need for a detailed numeric stability analysis. Such a system can be easily destabilized, and special attention must be paid during the discretization phase.

To find a maximum stability margin for a physical flow parameter, such as pipeline dimension or fluid properties, the influence of each of them on the estimation of μ_{opt} must be studied. In particular, numerical optimization was performed by using the Hooke and Jeeves (1961) algorithm to find the values of μ_{opt} (the dependent variable) for specified flow parameters. Six experiments were conducted where one of the parameters was an independent variable, and the others were fixed.

The toolbox *curve fitting tool* (cftool) of The MathWorks, Inc. (2012) was used to fit the curve to the experimental data. By assuming the common fitting curve

$$\mu_{opt} = C_1 p_p^{C_2} \tag{3.30}$$

where C_1 and C_2 are the coefficients to be optimized and p_p is the physical flow parameter which could be: N, D, L, λ, the mean pressure p_m in the pipeline and the pressure drop p_m along the pipeline. The quality of the fitting is evaluated via

the R-squared coefficient of determination R^2. The closer it is to 1, the better the curve fits the experimental data (Walpole et al., 2012). The adjusted functions are presented in Fig. 3.3, where each subplot refers to a different physical parameter as the independent variable.

By interpreting the identified parameters of (3.30) for a global case as follows: (1) the term $p_p^{C_2}$ combined into the single parameter ξ which includes all the physical parameters with a common power $\frac{1}{2}$; and (2) the individual scaling factor C_1 which is united in a global scaling coefficient C_3; the simple optimal stability adjustment is proposed

$$\mu_{opt} = C_3 \sqrt{\frac{p_d L \lambda}{p_m D}} \frac{1}{N} = C_3 \xi \tag{3.31}$$

where C_3 is the sought coefficient. For this task, the MATLAB's cftool is used to determine the value of the global scaling coefficient C_3 in (3.31).

The data set for the experiment is generated by 10 pipelines with randomly generated parameters. The fitting result is shown in Fig. 3.4, where the experimental data, aggregated to the single parameter ξ, are indicated by the symbol $\times$. The fitted value of C_3=0.36 is characterized by the coefficient of determination R^2=0.97. Note in Fig. 3.4 that some experimental data are slightly displaced with respect to the fitting curve. The reason for this should be attributed to the most 'severe' mathematical operation consisting in rounding the power coefficient 0.44 (the case reflected in Fig. 3.3c) to 0.5 (the case implemented by Eq. (3.31)). In both cases, the issue of fitting the μ_{opt} to the pressure drop along the pipeline is considered. Nevertheless, the relatively high value of the computed determination coefficient R^2 shows that the curve fits the experimental data satisfactorily, and thus the obtained expression can be practically used for approximating μ_{opt}.

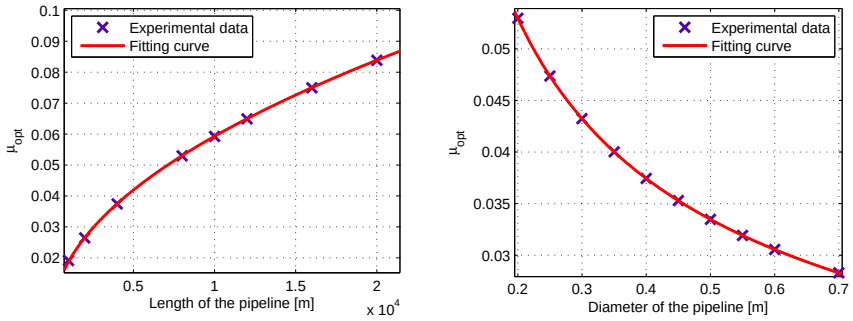
Thus the formula connecting the time and spatial steps with the maximal stability margin is

$$\mu_{opt} = 0.36 \sqrt{\frac{p_d L \lambda}{p_m D}} \frac{1}{N} \tag{3.32}$$

This stability margin is applicable to models (3.23) and (3.7). For models constructed with other assumptions, (3.32) does not guarantee stability; however, the optimization process can be reproduced analogically, obtaining the result fitted to the analyzed model.

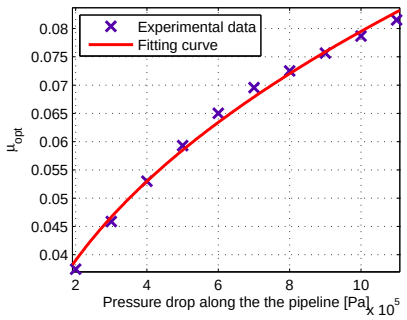
3.6 Analytic Inversion of the Recombination Matrix

In the case of flow process model, there are two main reasons for applying efficient computational analytic inversion matrix methods. The first relates to the Gauss-Jordan elimination method, which has the time complexity $O(n^3)$, which implies

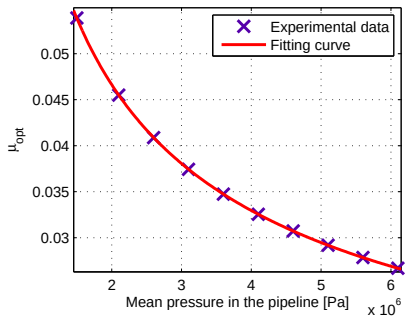


(a) C_1=5.92·10^{-4}, C_2=0.5, R^2=1

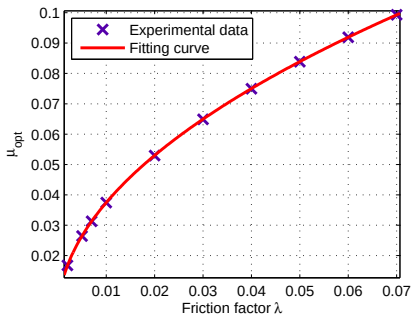
(b) C_1=0.024, C_2=-0.5, R^2=1



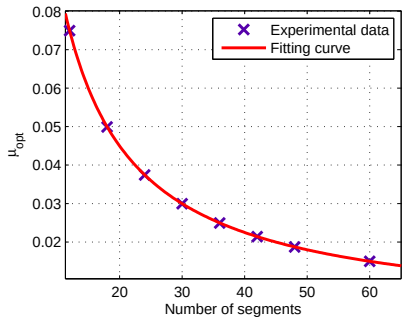
(c) C_1=1.75·10^{-4}, C_2=0.44, R^2=0.994



(d) C_1=67.03, C_2=-0.5, R^2=1



(e) C_1=0.38, C_2=0.5, R^2=1



(f) C_1=0.90, C_2=-1, R^2=1

Fig. 3.3: Results of adjusted function (3.30) for μ_{opt} by varying the following: (a) length of the pipeline, (b) diameter of the pipeline, (c) difference between the inlet and the outlet pressure, (d) mean pressure in the pipeline, (e) friction factor and (f) number of the pipeline's segments. The used fixed parameters for not independent variables are: N=12, D=0.4 [m], λ=0.01, ν=304 [$\frac{m}{s}$], p_{inlet}=4.1 [MPa], and p_{outlet}=3.9 [MPa]

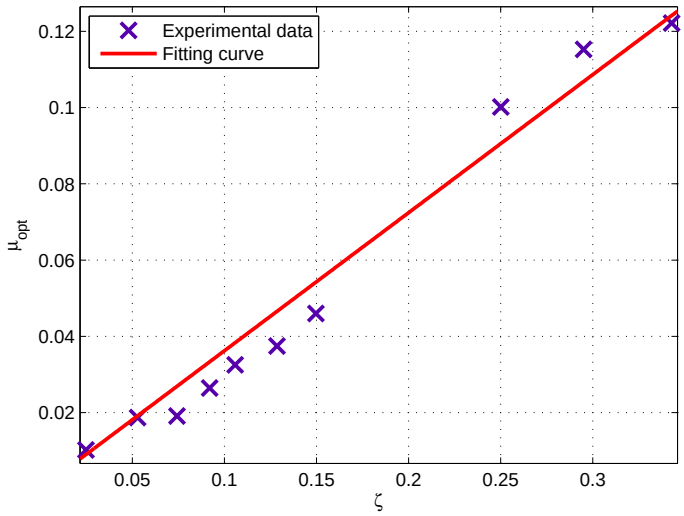


Fig. 3.4: Adjusting of (3.31) to the experimental data

that the computation time strongly increases with the dimension of the model. The second is caused by a property of the recombination matrix which has a ratio of its largest to its smallest singular values of order 10^9, and as a consequence matrix A is very sensitive to numerical errors.

The above facts motivated the inverse problem formulation within the framework of the tridiagonal matrix with the matrix A divided into four submatrices

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = \left[\begin{array}{cccccc:cccc} & & & & & & 2b & 0 & \cdots & 0 & 0 \\ & & & & & & -b & b & \cdots & 0 & 0 \\ & & & & & & \vdots & & \ddots & & \vdots \\ & & & D_{\frac{N}{2}+1}(c) & & & 0 & 0 & \cdots & -b & b \\ & & & & & & 0 & 0 & \cdots & 0 & -2b \\ \hdashline -b & b & 0 & \cdots & 0 & 0 & 0 & & & & \\ 0 & -b & b & \cdots & 0 & 0 & 0 & & & & \\ \vdots & & & \ddots & & \vdots & & & D_{\frac{N}{2}}(a) & & \\ 0 & 0 & 0 & \cdots & -b & b & 0 & & & & \\ 0 & 0 & 0 & \cdots & 0 & -b & b & & & & \end{array}\right] \tag{3.33}$$

Since matrix A consists of two diagonal submatrices and two submatrices having elements only on two diagonals, by (3.26) the matrix A^{-1} can be represented as

$$A^{-1} = \left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right]^{-1} = \left[\begin{array}{c|c} A'_1 & A'_2 \\ \hline A'_3 & A'_4 \end{array}\right] \tag{3.34}$$

with

$$A'_1 = \left(D_{\frac{N}{2}+1}(c) - A_2 D_{\frac{N}{2}}(a^{-1}) A_3\right)^{-1} \tag{3.35}$$

$$A'_4 = \left(D_{\frac{N}{2}}(a) - A_3 D_{\frac{N}{2}+1}\left(c^{-1}\right) A_2\right)^{-1} \tag{3.36}$$

$$A'_2 = D_{\frac{N}{2}+1}\left(-c^{-1}\right) A_2 A'_4 \tag{3.37}$$

$$A'_3 = D_{\frac{N}{2}}\left(-a^{-1}\right) A_3 A'_1 \tag{3.38}$$

From the set of matrices, one can see that A'_2 and A'_3 are dependent on matrices A'_4 and A'_1, which have to be determined first. Thus, (3.35) and (3.36) can be directly calculated which results in the following tridiagonal matrices:

$$A'_1 = \begin{bmatrix} c+\frac{2b^2}{a} & -\frac{2b^2}{a} & 0 & \cdots & 0 & 0 & 0 \\ -\frac{b^2}{a} & c+\frac{2b^2}{a} & -\frac{b^2}{a} & \cdots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{b^2}{a} & c+\frac{2b^2}{a} & -\frac{b^2}{a} \\ 0 & 0 & 0 & \cdots & 0 & -\frac{2b^2}{a} & c+\frac{2b^2}{a} \end{bmatrix}^{-1} \tag{3.39}$$

$$A'_4 = \begin{bmatrix} a+\frac{3b^2}{c} & -\frac{b^2}{c} & 0 & \cdots & 0 & 0 & 0 \\ -\frac{b^2}{c} & a+\frac{2b^2}{c} & -\frac{b^2}{c} & \cdots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{b^2}{c} & a+\frac{2b^2}{c} & -\frac{b^2}{c} \\ 0 & 0 & 0 & \cdots & 0 & -\frac{b^2}{c} & a+\frac{3b^2}{c} \end{bmatrix}^{-1} \tag{3.40}$$

3.6.1 Tridiagonal Matrix Inversion Method

Da Fonseca and Petronilho (2001) determined the analytical expression of the inverse matrix for a general tridiagonal matrix with the general structure

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \gamma_1 & \alpha_2 & \beta_2 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ 0 & & & \gamma_{n-1} & \alpha_n \end{bmatrix} \tag{3.41}$$

The authors showed that the coefficients of T^{-1} are given by

$$T^{-1}(i,j) = t_{i,j} = \begin{cases} \frac{(-1)^{i+j}\beta_i \cdots \beta_{j-1}\theta_{i-1}\phi_{j+1}}{\theta_n} & \text{for} \quad i < j \\ \frac{\theta_{i-1}\phi_{j+1}}{\theta_n} & \text{for} \quad i = j \\ \frac{(-1)^{i+j}\gamma_j \cdots \gamma_{i-1}\theta_{j-1}\phi_{i+1}}{\theta_n} & \text{for} \quad i > j \end{cases} \tag{3.42}$$

with the parameters θ_i and ϕ_i calculated recursively as

$$\theta_i = \alpha_i\theta_{i-1} - \beta_{i-1}\gamma_{i-1}\theta_{i-2} \quad \text{for} \quad i = 2,3,\ldots,n; \quad \theta_0 = 1, \quad \theta_1 = \alpha_1 \tag{3.43}$$

$$\phi_i = \alpha_i\phi_{i+1} - \beta_i\gamma_i\phi_{i+2} \quad \text{for} \quad i = n-1, n-2, \ldots, 1; \quad \phi_{n+1} = 1, \quad \phi_n = \alpha_n \tag{3.44}$$

Because the matrix (3.39) has a centrosymmetric structure, $\phi_i = \theta_{n+1-i}$, and its inverse is also centrosymmetric, thus, only half of the coefficients of A_1' must be calculated by using (3.42). Therefore, the general matrix is obtained:

$$A_1' = \begin{bmatrix} a'_{1,1} & a'_{1,2} & \cdots & a'_{1,n-1} & a'_{1,n} \\ a'_{n-1,n} & a'_{2,2} & & a'_{2,n-1} & a'_{2,n} \\ \vdots & & \ddots & & \vdots \\ a'_{2,n} & a'_{2,n-1} & & a'_{2,2} & a'_{n-1,n} \\ a'_{1,n} & a'_{1,n-1} & & a'_{1,2} & a'_{1,1} \end{bmatrix} \tag{3.45}$$

Similar reasoning applies to (3.40). Moreover, this matrix is symmetric not only centrosymmetric. Thus, the number of elements to be calculated is only $n^2/4$. Hence, matrix A_4' has the following form

$$A_4' = \begin{bmatrix} a''_{1,1} & a''_{1,2} & \cdots & a''_{1,n-1} & a''_{1,n} \\ a''_{1,2} & a''_{2,2} & & a''_{2,n-1} & a''_{1,n-1} \\ \vdots & & \ddots & & \vdots \\ a''_{1,n-1} & a''_{2,n-1} & & a''_{2,2} & a''_{1,2} \\ a''_{1,n} & a''_{1,n-1} & \cdots & a''_{1,2} & a''_{1,1} \end{bmatrix} \tag{3.46}$$

where the values of $a''_{i,j}$ are calculated with (3.42). Once the matrices A_1' and A_4' are evaluated, one can use them to calculate (3.37) and (3.38), resulting in

$$A_2' = \frac{b}{c}\left[\begin{array}{c|c|c|c} -2a''_{1,1} & -2a''_{1,2} & \cdots & -2a''_{1,n-1} \\ a''_{1,1}-a''_{1,2} & a''_{1,2}-a''_{2,2} & \cdots & a''_{1,n-1}-a''_{2,n-1} \\ \vdots & & \ddots & \\ a''_{1,n-1}-a''_{1,n} & a''_{2,n-1}-a''_{1,n-1} & \cdots & a''_{2,2}-a''_{1,2} \\ 2a''_{1,n} & 2a''_{1,n-1} & \cdots & 2a''_{1,2} \end{array} \begin{array}{c} -2a''_{1,n} \\ a''_{1,n}-a''_{1,n-1} \\ \vdots \\ a''_{1,2}-a''_{1,1} \\ 2a''_{1,1} \end{array}\right] \tag{3.47}$$

$$A_3' = \frac{b}{a}\left[\begin{array}{c|c|c|c} a'_{1,1}-a'_{n-1,n} & a'_{1,2}-a'_{2,2} & \cdots & a'_{1,n-1}-a'_{2,n-1} \\ a'_{n-1,n}-a'_{n-2,n} & a'_{2,2}-a'_{n-2,n-1} & \cdots & a'_{2,n-1}-a'_{3,n-1} \\ \vdots & & \ddots & \\ a'_{3,n}-a'_{2,n} & a'_{3,n-1}-a'_{2,n-1} & \cdots & a'_{n-2,n-1}-a'_{2,2} \\ a'_{2,n}-a'_{1,n} & a'_{2,n-1}-a'_{1,n-1} & \cdots & a'_{2,2}-a'_{1,2} \end{array} \begin{array}{c} a'_{1,n}-a'_{2,n} \\ a'_{2,n}-a'_{3,n} \\ \vdots \\ a'_{n-2,n}-a'_{n-1,n} \\ a'_{n-1,n}-a'_{1,1} \end{array}\right] \tag{3.48}$$

Matrices A_2' and A_3' are anti-centrosymmetric, which reduces again the number of the required mathematical operations. Thus, the four submatrices which form matrix A^{-1} are known and can be assembled into one. The state-space model expressed in term of the A^{-1} will be called the analytical model of band matrix inversion (AMBMI). A disadvantage of this method is that for high-order systems, the inverse cannot be computed because some θ_i and ψ_i have values higher than the computer's numerical representation range (over 10^{300}). Therefore, this model is recommended only for a lower number of segments.

3.6.2 Diagonal Approximation Model

To avoid the disadvantage of the analytical inversion matrix, Kowalczuk and Tatara (2016) suggested approximating the tridiagonal matrices (3.39) and (3.40) with their diagonal counterparts. To make such approximation feasible, the coefficients' values on the subdiagonal and superdiagonal must be significantly lower than the ones on the main diagonal. This sufficient condition satisfies if

$$|c| \gg \left|4\frac{b^2}{a}\right| \tag{3.49}$$

By substituting physical parameters of the pipeline in (3.49), the inequality

$$|\frac{3}{2\mathscr{A}\Delta t}| \gg |4\frac{1}{16\Delta z^2}\frac{2\nu^2\Delta t}{3\mathscr{A}}| \tag{3.50}$$

is obtained. Since Δt, Δz and ν are positive, the above equation can be simplified to

$$\Delta t^2 \ll 9\frac{\Delta z^2}{\nu^2} \tag{3.51}$$

which is equivalent to

$$\Delta t \ll 3\frac{\Delta z}{\nu} \tag{3.52}$$

By incorporating the CFL condition (3.27) in (3.52), the restriction on μ is reduced to

$$\mu \ll 3 \tag{3.53}$$

By considering the practical assumption that μ must be at least two orders of magnitude fewer than the value obtained with (3.53), one suggests the following stability tuner's condition

$$\mu < 0.03 \tag{3.54}$$

By comparing inequality (3.54) with the Courant-Friedrichs-Lewy result discussed by Dick (2012), who says that μ must be lower than 0.90 to 0.95 for stationary flows (depending on flow parameters), it is clear that (3.54) is more restrictive because it has been obtained by considering an approximated model. Moreover, the CFL criterion solely defines a necessary condition; thus the values may not assure numerical stability.

By considering (3.54), the two submatrices of the inverted recombination matrix can be approximated as

$$A_1' \approx \breve{A}_1' = \begin{bmatrix} c+\frac{2b^2}{a} & 0 & 0\cdots 0 & 0 & 0 \\ 0 & c+\frac{2b^2}{a} & 0\cdots 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0\cdots 0 & c+\frac{2b^2}{a} & 0 \\ 0 & 0 & 0\cdots 0 & 0 & c+\frac{2b^2}{a} \end{bmatrix}^{-1} \tag{3.55}$$

and

$$A_4' \approx \breve{A}_4' = \begin{bmatrix} a+\frac{3b^2}{c} & 0 & \cdots\ 0\ 0 & 0 & \\ 0 & a+\frac{2b^2}{c} & 0\ \cdots 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0\ \cdots 0 & a+\frac{2b^2}{c} & 0 \\ 0 & 0 & 0\ \cdots 0 & 0 & a+\frac{3b^2}{c} \end{bmatrix}^{-1} \tag{3.56}$$

The inverses of these matrices are obtained by substituting their reciprocals on the main diagonal. Therefore, the inverse recombination matrix

$$A^{-1} \approx \left[\begin{array}{ccccc:ccccc}
\frac{a}{\sigma} & 0 & \cdots & 0 & 0 & -\frac{2b}{3b^2+ca} & 0 & \cdots & 0 & 0 \\
0 & \frac{a}{\sigma} & & 0 & 0 & \frac{b}{3b^2+ca} & -\frac{b}{\sigma} & 0 & 0 & 0 \\
 & & & & & 0 & \frac{b}{\sigma} & -\frac{b}{\sigma} & 0 & 0 \\
\vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\
 & & & & & 0 & 0 & \frac{b}{\sigma} & -\frac{b}{\sigma} & 0 \\
0 & 0 & & \frac{a}{\sigma} & 0 & 0 & 0 & 0 & \frac{b}{\sigma} & -\frac{b}{3b^2+ca} \\
0 & 0 & \cdots & 0 & \frac{a}{\sigma} & 0 & 0 & \cdots & 0 & \frac{2b}{3b^2+ca} \\
\hdashline
\frac{b}{\sigma} & -\frac{b}{\sigma} & 0 & \cdots & 0 & 0 & \frac{c}{3b^2+ca} & 0 & \cdots & 0 & 0 \\
0 & \frac{b}{\sigma} & -\frac{b}{\sigma} & & 0 & 0 & 0 & \frac{c}{\sigma} & & 0 & 0 \\
0 & 0 & \frac{b}{\sigma} & & 0 & 0 & & & & & \\
\vdots & & \ddots & & \vdots & & \vdots & & \ddots & & \vdots \\
0 & 0 & & -\frac{b}{\sigma} & 0 & 0 & & & & & \\
0 & 0 & & \frac{b}{\sigma} & -\frac{b}{\sigma} & 0 & 0 & 0 & & \frac{c}{\sigma} & 0 \\
0 & 0 & \cdots & 0 & \frac{b}{\sigma} & -\frac{b}{\sigma} & 0 & 0 & \cdots & 0 & \frac{c}{3b^2+ca}
\end{array}\right] \tag{3.57}$$

is obtained by using (3.26) with $\sigma = 2b^2 + ca$. This matrix constitutes an explicit form of the inverted recombination matrix. Since this inversion is performed after the estimation of the friction factor which could change with the operation condition, or during the initialization, the inversion algorithm consumes short computational time. Since (3.57) is now a sparse matrix, another advantage is that the online computation of (3.24) and (3.25), and the premultiplication by a sparse matrix is not time-consuming. The model obtained with this inverse method for the recombination matrix is called the analytic model of diagonal approximation (AMDA). Certainly, the model is an approximation of the base model; thus its accuracy and calculation speed must be examined.

3.6.2.1 Note on the Model's Dimension

By considering the maximum stability margin given in (3.32) with the CFL condition (3.54) for the AMDA model, one deduces that

$$\mu_{opt} = 0.36\sqrt{\frac{p_d L\lambda}{p_m D}}\frac{1}{N} < 0.03 \tag{3.58}$$

Thus, if the above condition is satisfied, the AMDA model is a good approximation for the base model (3.7), and the stability tuner μ_{opt} for AMDA can be computed by

using (3.32). The formula (3.58) can be rewritten in a practical form, which gives us information about the minimal cardinality of the AMDA model as

$$N > 12\sqrt{\frac{p_d L\lambda}{p_m D}} \tag{3.59}$$

As a consequence, if a properly designed AMDA's dimension is selected beyond its minimum, one can apply (3.32) to calculate μ_{opt}, which, in turn, satisfies (3.58) and assures a maximum stability margin.

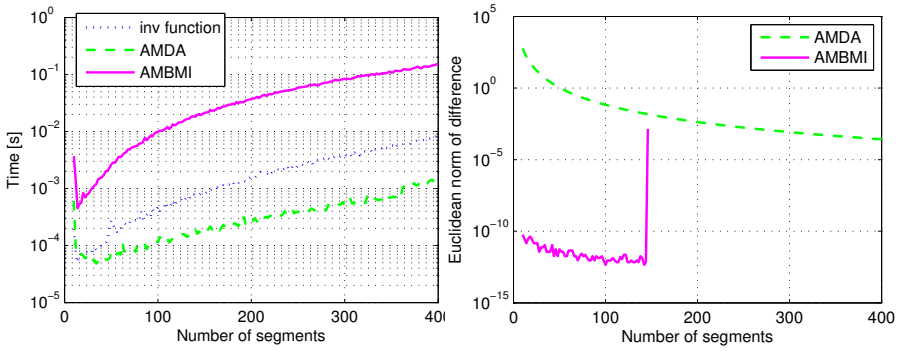
Note that μ_{opt} differs slightly from model to model. When the accuracy of the inversion improves, however, the optimal value of the coefficient for the AMDA model approaches the value for the principal base model; certainly, the segmentation (3.59) must be rounded up to the nearest even integer.

3.7 Analysis of the Models

Based on the two methods AMBMI and AMDA for inverting the recombination matrix which have been presented in the previous section, one can say that the former method generates values beyond the acceptable computer representation range, making it impossible to compare accuracy and convergence with other models from a practical point of view. On the other hand, the AMDA model for lower (rough) segmentation produces an inverse which is not similar to the numerically inverted matrix considering the Euclidean norm of the variables as a measure. For this reason, computation accuracy and speed will be assessed in detail only for the AMDA model, if one compares it with the base model. The inversion time for the three methods has been evaluated with respect to the model's dimension, and the results are shown in Fig. 3.5a. The inversion accuracy is assessed in terms of the Euclidean norm of the difference between the matrices inverted by the AMDA or AMBMI methods and the MATLAB inv function. The results are shown in Fig. 3.5b. The experimental data used are L=4000 [m], v=304.23 [$\frac{m}{s}$], D=0.4 [m], and λ=0.01. Each calculation was repeated 10 times, and the results are averaged.

A more detailed analysis has been conducted for the AMDA model as compared to the inv function. The models are analyzed in terms of the Euclidean norm of the difference between the two inverted matrices (which should ideally be equal to zero). The inversion time of the matrix gained from the two methods has been presented as a function of μ and the number of segments N. Finally, the models have been compared in Fig. 3.6 in terms of the mean squared error between the mass flow estimates (at the inlet and outlet) obtained by both of them.

Note that in Fig. 3.6 the axes in subplots (a, c, e,) have a log-log scale with N ranging from 10 to 400. For subplots (b, d, f) N=100 and μ is calculated by using (3.32). Subplots a and b describe with a continuous line the Euclidean norm of the difference between the inverted recombination matrices by AMDA and by inv functions; the dashed line is the Euclidean norm of the difference between state tran-



(a) (b)

Fig. 3.5: Comparison of AMDA and AMBMI models. (a) inversion time; continuous line is used for the AMBMI method, the discontinuous line for the AMDA model, and the dots line for the MATLAB inv function. (b) accuracy measure, the continuous line denotes the norm of the difference between the AMBMI and inv function, and dashed line denotes the difference between the functions AMDA and inv.

sition matrices obtained by AMDA and inv functions. The subplots c and d describe the numerical inversion time results with continuous line for the inv function, and dashed line for the analytic inversion time (AMDA). Finally, subplots e and f show the result of the mean quadratic difference between mass flow estimated by the two analyzed models. All the presented results were obtained with mean values of 10 runs.

3.8 Conclusions

In the case of inversion time, the AMDA outperforms other methods for almost each segmentation, while AMBMI is the slowest one. Note, however, that the MATLAB inv function is effectively optimized, which may not be available in a field computer running the diagnostic algorithm. Note also that for $N \approx 140$ segments, the AMBMI model is not computable at all, though, because of the lower segmentation it results in a better approximation of inversion. The AMBMI method is, however, limited to a lower order due to the high values of the auxiliary variables. The upper boundary for the use of this model depends on the physical parameters of the flow; therefore it is difficult to give a number. A safe upper boundary seems to be $N \approx 50$, however.

As seen in Figs. 3.6a and 3.6b, the norm difference between the two recombination matrices decreases by decreasing the coefficient μ or by increasing the order of the model. Since the state transition matrices depend on the recombination matrix, the norm difference is smaller in every case. The inversion time is almost inde-

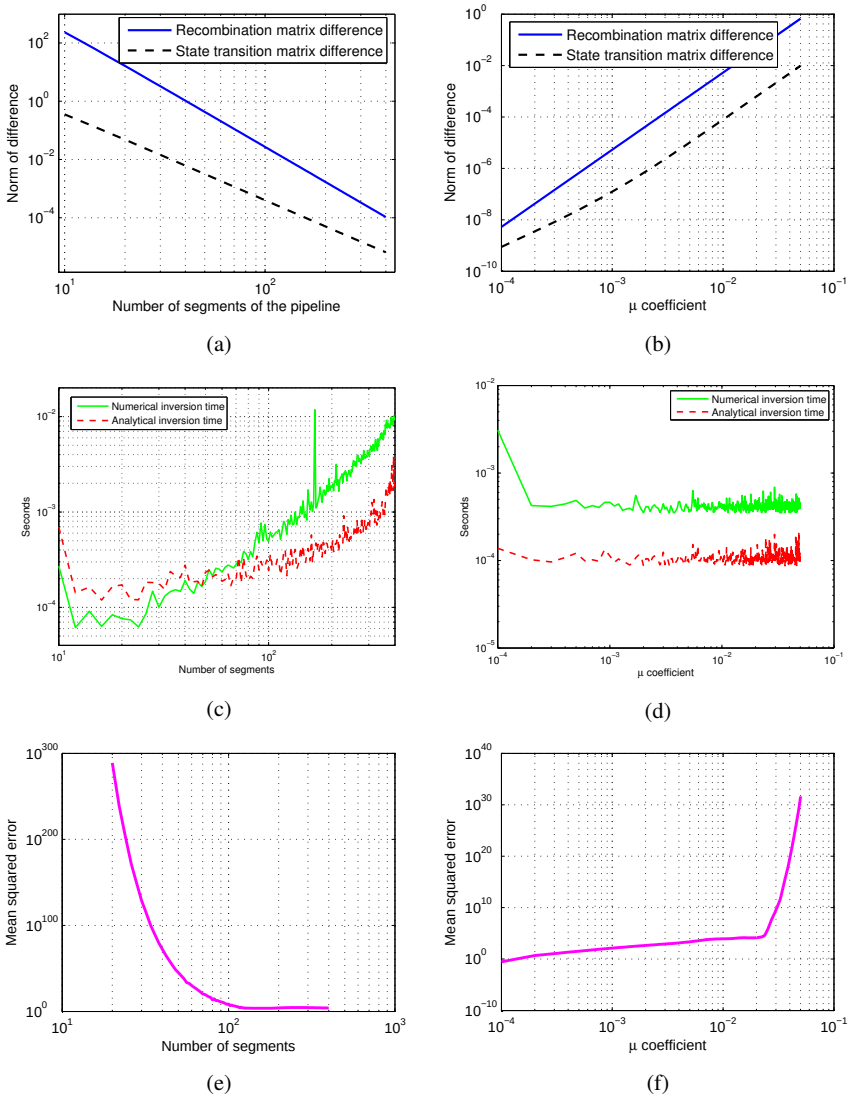


Fig. 3.6: Numerical results of the AMDA inversion method as compared to the MATLAB inv function for a pipeline with L=2000 [m], v=304.23 [$\frac{m}{s}$], D=0.4 [m], p_{inlet}=3.2 [MPa], p_{outlet}=3.0 [MPa], and $\lambda = 0.1$

pendent from the coefficient μ for both methods; however, increasing the model's order increases the time necessary for inversion. For lower segmentation, the analytical approach is slower, but for the number of segments above approximately 50, the analytic approach outperforms the numerical one. Figs. 3.6e and 3.6f show the

mean quadratic difference between the two models. Clearly, for lower segmentation, the difference level is significant and decreases when one increases segmentation. Above 100 segments, the error line becomes flat, and this region seems to constitute the preferred operation range for the model. In the case of the coefficient μ, the error is almost flat for μ <0.02 and rapidly increases with μ above this limit.

In summary, the recommendations for using the AMDA model are that the number of segments should be higher than 100, for the following reasons: to benefit from the low quadratic error of the mass flow estimates and from faster inversion, while the coefficient μ should remain below 0.02. The model should also be tested in an actual LDI algorithm, however, to assess the resulting accuracy of the target estimates of leak parameters.

References

Belsito, S., Lombardi, P., Andreussi, P., & Banerjee, S. (1998). Leak detection in liquefied gas pipelines by artificial neural networks. *AIChE*, 44(12), 2675–2688.

Billmann, L. & Isermann, R. (1987). Leak detection methods for pipelines. *Automatica*, 23(3), 381–385.

Brogan, W. (1991). *Modern control theory*. Boston: Prentice Hall.

Da Fonseca, C. & Petronilho, J. (2001). Explicit inverses of some tridiagonal matrices. *Linear Algebra and its Applications*, 325, 7–21.

Dick, M. (2012). *Stabilization of the gas flow in networks: Boundary feedback stabilization of quasilinear hyperbolic systems on networks*. Ph.D. thesis. Erlangen-Nürnberg: Friedrich-Alexander-Universität.

Gunawickrama, K. (2001). *Leak detection methods for transmission pipelines*. Ph.D. thesis. Gdask: Gdask Univeristy of Technology.

Hooke, R., & Jeeves, T. A. (1961). *"Direct search" solution of numerical and statistical problems* Journal of the Association for Computing Machinery (ACM). 8 (2): 212229.

Kowalczuk, Z., & Gunawickrama, K. (1998). Detection of leakages in industry pipelines using a cross-correlation approach. *Pomiary Automatyka Kontrola*, 44(4), 140–146.

Kowalczuk, Z., & Gunawickrama, K. (2004). *Detection and localisation of leaks in transmission pipelines*, chapter 21, (pp. 821–864). New York: Springer.

Kowalczuk, Z., & Tatara, M. (2013)p Analytical modeling of flow processes: Analysis of computability of a state-space model. In *XI International Conference on Diagnostics of Processes and Systems* (pp. 74.1–12). Lagów, Lubuski.

Kowalczuk, Z. & Tatara, M. (2016). *Approximate models and parameter analysis of the flow process in transmission pipelines* (pp. 239–252). Springer.

Kreyszig, E. (2006). *Advanced engineering mathematics*. (Vol. 9). Columbus: John Wiley and Sons Inc.

Reddy, H., Narasimhan, S., Bhallamudi, S.M., & Bairagi, S. (2011). Leak detection in gas pipeline networks using an efficient state estimator. Part-I: Theory and simulations. *Computers and Chemical Engineering*, 35(4), 651–661.

Strikwerda, J. (2007). *Finite difference schemes and partial differential equations*. SIAM.

MathWorks. (2012). *MATLAB and Curve Fitting Toolbox release 2012b* Natick, Massachusetts, United States: Technical Report.

Torres, L., Besançon, G., & Verde, C. (2012). Leak detection using parameter identification. In *8th IFAC symposium SAFEPROCESS-2012*. Mexico City, Mexico.

Verde, C. and Torres, L. (2015). Referenced model based observers for locating leaks in a branched pipeline. In *The 9th international federation of automatic control (IFAC) symposium SAFEPROCESS*, (pp. 1066–1071). Paris: IFAC.

Walpole, R., Myers, R., Myers, S., & Ye, K. (2012). *Probability and statistics for engineers and scientists*. (9th ed.). Boston: Prentice Hall.