Available online at www.sciencedirect.com

ScienceDirect

C. R. Acad. Sci. Paris, Ser. I 346 (2008) 119–124

Comptes Rendus Mathematique

http://france.elsevier.com/direct/CRASS1/

Numerical Analysis

Computing the matrix sign and absolute value functions

Michaël Ndjinga [a,b]

[a] *Commissariat à l'énergie atomique, centre de Saclay, CEA/DEN/DM2S/SFME, 91191 Gif-sur-Yvette, France*

[b] *Laboratoire de mathématiques appliquées aux systèmes, École centrale Paris, 92295 Châtenay-Malabry cedex, France*

Received 17 May 2006; accepted after revision 22 November 2007

Presented by Olivier Pironneau

Abstract

We present two algorithms for the computation of the matrix sign and absolute value functions. Both algorithms avoid a complete diagonalisation of the matrix, but they however require some informations regarding the eigenvalues location. The first algorithm consists in a sequence of polynomial iterations based on appropriate estimates of the eigenvalues, and converging to the matrix sign if all the eigenvalues are real. Convergence is obtained within a finite number of steps when the eigenvalues are exactly known. Nevertheless, we present a second approach for the computation of the matrix sign and absolute value functions, when the eigenvalues are exactly known. This approach is based on the resolution of an interpolation problem, can handle the case of complex eigenvalues and appears to be faster than the iterative approach. ***To cite this article: M. Ndjinga, C. R. Acad. Sci. Paris, Ser. I 346 (2008).***

Résumé

Calcul du signe et de la valeur absolue d'une matrice. Nous présentons deux algorithmes pour le calcul des fonctions signe et valeur absolue matricielles. Ces algorithmes évitent une diagonalisation complète de la matrice, mais nécessitent des estimations de ses valeurs propres. Le premier algorithme consiste en une suite d'itérations polynomiales construite à partir d'approximations des valeurs propres, et convergeant vers le signe de la matrice lorsque les valeurs propres sont réelles. La convergence s'obtient en un nombre fini d'itérations lorsqu'elles sont connues exactement. Nous présentons cependant une seconde approche s'appuyant sur la résolution d'un problème d'interpolation polynomiale, pour le calcul des fonctions signe et valeur absolue matricielles dans le cas où les valeurs propres sont connues. Ce second algorithme se révèle plus rapide que le premier, et permet la prise en compte de matrices ayant des valeurs propres complexes. ***Pour citer cet article : M. Ndjinga, C. R. Acad. Sci. Paris, Ser. I 346 (2008).***

Version française abrégée

Les fonctions signe et valeur absolue matricielles apparaissent dans des domaines aussi divers que la simulation numérique d'équations aux dérivées partielles hyperboliques [7,11], la théorie du contrôle ou l'analyse matricielle [6]. Ces deux fonctions sont liées par la relation $|A| = A \times \operatorname{sign}(A)$, et leur définition (2) fait classiquement intervenir la décomposition spectrale de la matrice argument. Cette décomposition n'est en général pas triviale, et la recherche de

E-mail address: michael.ndjinga@cea.fr.

1631-073X/$ – see front matter
doi:10.1016/j.crma.2007.11.028

solutions algorithmiques s'impose. L'emploi d'algorithmes numériques effectuant une diagonalisation complète de la matrice peut présenter certaines difficultés. Le problème peut en effet se révèler mal conditionné, par exemple si deux vecteurs propres sont quasiment colinéaires. Ce cas singulier se présente souvent dans la simulation numérique de modèles moyennés d'écoulements diphasiques, et rend peu robuste l'utilisation des solveur de Riemann de type Roe [13], VFRoe [8] ou VFFC [4]. Nous proposons dans cette Note deux approches de calcul évitant une diagonalisation complète de la matrice, mais nécessitant des estimations de ses valeurs propres. Dans le cas de la simulation numérique d'écoulements diphasiques, ces estimations sont disponibles car le polynôme caractéristique de la matrice A admet une expression simple (voir par exemple [9,3]), et les deux algorithmes présentés dans cette Note peuvent donc être employés (voir [10]).

La première section présente une approche iterative du calcul du signe de la matrice A, dans le cas où celle ci est à valeurs propres réelles. On construit grâce au Lemme 2.1, une suite de matrices A_k ayant pour limite $\mathrm{sign}(A)$. Si les valeurs propres λ_i de la matrice A sont connues précisément, la suite A_k converge en un nombre fini d'itérations (Théorème 2.2). Dans ce cas, le nombre de multiplications de matrices requis est $2(n_d - 1)$, où n_d est le nombre de valeurs propres distinctes non nulles. Cet algorithme admet une extension dans le cas où seules des bornes λ_{i+} et λ_{i-} telles que $0 < \lambda_{i-} < |\lambda_i| < \lambda_{i+}$ sont disponibles (Théorème 2.3). La convergence de cette extension ne s'effectue cependant plus en un nombre fini d'itérations.

Dans la deuxième section, nous présentons une approche par interpolation pour le calcul des fonctions signe et valeur absolue matricielles. Le point de départ est le Théorème 3.1 : si un polynôme P_{int} interpole la fonction valeur absolue (respectivement, la fonction signe) sur les points de support définis par le spectre de A, alors $P_{\mathrm{int}}(A) = |A|$ (respectivement, $P_{\mathrm{int}}(A) = \mathrm{sign}(A)$). Cette approche permet de traiter le cas des matrices admettant des valeurs propres complexes. Nous réduisons alors le problème de la détermination de P_{int} à un problème d'interpolation plus simple grace aux formules (13) et (14). En construisant le polynôme d'interpolation de Lagrange Q_+ satisfaisant (14), l'évaluation de $|A|$ fait appel à moins de $\max\{n_+, n_-\}$ multiplications de matrices, où n_+ (respectivement, n_-) est le nombre de valeurs propres de partie réelle positive (respectivement négative) ou nulle. Cet algorithme est donc plus rapide que le premier.

1. Introduction

In this Note, we use the following extensions of the sign and absolute value functions to complex numbers $z \in \mathbb{C} \setminus i\mathbb{R}^*$

$$\mathrm{sign}(z) = \begin{cases} -1 & \text{if } \mathrm{Re}(z) < 0, \\ 0 & \text{if } z = 0, \\ +1 & \text{if } \mathrm{Re}(z) > 0, \end{cases} \qquad |z| = z \times \mathrm{sign}(z) = \begin{cases} -z & \text{if } \mathrm{Re}(z) < 0, \\ 0 & \text{if } z = 0, \\ +z & \text{if } \mathrm{Re}(z) > 0. \end{cases} \tag{1}$$

Note that $|z|$ is a complex number different from z modulus. Other extensions of the sign and absolute value functions to complex numbers are possible, but the general principle of the interpolation approach (Theorem 3.1) would still be relevant.

Consider a real $n \times n$ diagonalisable matrix A, with eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C} \setminus i\mathbb{R}^*$, left eigenvectors $\mathbf{l}_1, \ldots, \mathbf{l}_n$ and right eigenvectors $\mathbf{r}_1, \ldots, \mathbf{r}_n$. One can normalise the eigenvectors in such a way that $\mathbf{l}_i \cdot \mathbf{r}_i = 1$. Then the following decomposition holds

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{l_i} \otimes \mathbf{r_i}.$$

The sign and absolute value $\mathrm{sign}(A)$ and $|A|$ are real matrices sharing the same left and right eigenvectors as A, with associated eigenvalues $\mathrm{sign}(\lambda_1), \ldots, \mathrm{sign}(\lambda_n)$ for $\mathrm{sign}(A)$, and $|\lambda_1|, \ldots, |\lambda_n|$ for $|A|$:

$$\mathrm{sign}(A) = \sum_{i=1}^{n} \mathrm{sign}(\lambda_i) \mathbf{l}_i \otimes \mathbf{r}_i, \qquad |A| = \sum_{i=1}^{n} |\lambda_i| \mathbf{l}_i \otimes \mathbf{r}_i = A \times \mathrm{sign}(A). \tag{2}$$

The matrix sign and absolute value functions arise in many engineering and mathematical fields such as the numerical analysis of hyperbolic systems of PDEs [7], control theory or matrix analysis [6]. The Roe approximate Riemann solver [11], successfully applied to the numerical simulation of single phase flows, requires an efficient computation of the matrix absolute value function. Recent characteristic based methods, designed for the numerical simulation of multiphase flow averaged models, also involve the computation of the matrix sign or absolute value functions (see [13,8,4]). However these models may lack hyperbolicity [9], and one has to consider the general case of real matrices with complex eigenvalues.

The classical definitions of $|A|$ and $\mathrm{sign}(A)$ (2) involve the left and right eigenvectors of A. In the general case, the matrix A has non trivial eigenvalues and eigenvectors, and one should consider numerical algorithms to obtain its spectral decomposition. However, in the case of close eigenvalues or nearly parallel eigenvectors, the diagonalisation process may be ill conditioned (see [12]). Moreover, for a dense non symmetric matrix, a full diagonalisation can be costly in terms of computational time ([5] gives an estimate $25n^3$ flops for the QR method).

We propose two alternative computations of $|A|$ and $\mathrm{sign}(A)$ that do not involve the determination of A eigenvectors. Both algorithms however require some informations regarding the eigenvalues location. In the multiphase flow discipline (see [2] for an introduction) the characteristic polynomial generally admits a simple expression, and such estimates are available (see for example [9] and [3]). Otherwise, estimates of the eigenvalues may be obtained through Gershgorin Disc's localisation techniques.

In Section 2, it is assumed that all the eigenvalues are real, and we build a sequence of polynomial iterations that converges to $\mathrm{sign}(A)$ (Theorem 2.3). This sequence is based on estimates of the eigenvalues, and converges within a finite number of steps if the eigenvalues are exactly known (Theorem 2.2). Section 3 proposes a second approach based on the resolution of an interpolation problem. This approach can handle the case of complex eigenvalues, and requires less matrix multiplications than the first.

2. An iteration scheme for the matrix sign function

The aim of the Iteration Scheme is to build a sequence of polynomials $P_{k,k\in\mathbb{N}}$ such that the sequence of matrices $A_0 = A$, $A_{k+1} = P_k(A_k)$, converges to $\mathrm{sign}(A)$. As the eigenvectors are conserved by polynomial iterations, it is sufficient to require that for any eigenvalue λ of A, the sequence $\lambda_0 = \lambda$, $\lambda_{k+1} = P_k(\lambda_k)$ converges to $\mathrm{sign}(\lambda)$. For any matrix A with real eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{R}$, we define

$$\Lambda_A = \{|\lambda_i|, \lambda_i \neq 0\}, \quad n_d = \#\Lambda_A, \quad \lambda_m(A) = \min(\Lambda_A), \quad \lambda_M(A) = \max(\Lambda_A).$$

The following lemma details the construction of an iteration polynomial:

Lemma 2.1 *(Iteration polynomials). Consider a finite set of real numbers Λ, such that* $\min(\Lambda) = 1$*, and define* $\lambda_M = \max(\Lambda)$*. There is a third degree polynomial P_Λ satisfying*

$$P_\Lambda(-x) = -P_\Lambda(x), \quad \forall x \in \mathbb{R}, \tag{3}$$

$$P_\Lambda(1) = P_\Lambda(\lambda_M) = 1, \tag{4}$$

$$1 < P_\Lambda(\lambda) < \lambda, \quad \forall \lambda \in \Lambda \setminus \{1, \lambda_M\}. \tag{5}$$

Proof. $P_\Lambda = (1 + a_M)x - a_M x^3$ with $a_M = \frac{1}{\lambda_M(\lambda_M+1)}$ satisfies the first two requirements. Moreover, $P_\Lambda(x) - x = a_M x(1 - x^2)$ and thus $\forall x > 1$, $P_\Lambda(x) - x < 0$. Finally $P_\Lambda(x) - 1 = -a_M(x-1)(x-\lambda_M)(x+\lambda_M+1)$ and thus, $1 < x < \lambda_M \Rightarrow P_\Lambda(x) - 1 > 0$. Hence P_Λ satisfies the three requirements. □

If A satisfies $\min(\Lambda_A) = 1$ then according to Lemma 2.1, the largest eigenvalue in modulus of A is replaced by its sign in $P_\Lambda(A)$, while the other eigenvalues come closer to their sign. Repeating the operation, we obtain the following Iteration Scheme:

Theorem 2.2 *(Iteration Scheme). Given a non zero diagonalisable matrix A with real eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{R}$, the sequence*

$$\begin{cases} A_0 = \frac{1}{\lambda_m(A)} A, \quad A_{k+1} = P_{\Lambda_k}(A_k), \\ \Lambda_0 = \Lambda_{A_0}, \quad \Lambda_{k+1} = P_{\Lambda_k}(\Lambda_k), \end{cases} \tag{6}$$

converges to the fixed point $\begin{cases} A_{n_d-1} = \text{sign}(A), \\ \Lambda_{n_d-1} = \{1\}. \end{cases}$

Proof. As $\lambda_m(A) > 0$, the matrices A and A_0 share the same sign. From Lemma 2.1 we deduce that $\text{sign}(A_k) = \text{sign}(A)$. Besides, Eq. (4) implies that if $\#\Lambda_k > 1$ then $\#\Lambda_{k+1} < \#\Lambda_k$. Hence Λ_{n_d-1} necessarily reduces to $\{1\}$ and as $\Lambda_{n_d-1} = \Lambda_{A_{n_d-1}}$ we conclude that $A_{n_d-1} = \text{sign}(A_{n_d-1}) = \text{sign}(A)$, which is a fix point. □

It is therefore possible to perform an exact computation of $\text{sign}(A)$ using at most $(n_d - 1)$ iterations of the sequence (6), that is $2(n_d - 1)$ matrix multiplications. As the Iteration Scheme relies only on the knowledge of the eigenvalues and performs in a finite number of steps, it can handle the case of matrices with non diagonalisable Jordan blocks. However, it does not extend to the case where the matrix A presents complex eigenvalues. The set of convergence of polynomial iterations on the complex plane is known to have a fractal structure, and it thus seems difficult to give a good criterion for the Iteration Scheme to convergence.

2.1. Working with estimates of the eigenvalues

In this section we build an Iteration Scheme in the case where only upper and lower bounds, λ_{i+} and λ_{i-} such that $0 < \lambda_{i-} < |\lambda_i| < \lambda_{i+}$ are available for any non zero eigenvalue $\lambda_i \in \Lambda_A$. Define

$$\tilde{\Lambda}_A = \bigcup_{i=1}^{n_d} [\lambda_{i-}, \lambda_{i+}], \quad \lambda_m(A) = \min(\tilde{\Lambda}_A), \quad \lambda_M(A) = \max(\tilde{\Lambda}_A).$$

Given a compact set $\tilde{\Lambda} \subset [1, +\infty[$ we use the notation $\text{ext}(\tilde{\Lambda})$ for the set of its extremal points.

Theorem 2.3. *The sequence*

$$\begin{cases} A_0 = \frac{1}{\lambda_m(A)} A, \quad \tilde{\Lambda}_0 = \tilde{\Lambda}_{A_0}, \\ A_{k+1} = P_{\text{ext}(\tilde{\Lambda}_k)}(A_k), \quad \tilde{\Lambda}_{k+1} = P_{\text{ext}(\tilde{\Lambda}_k)}(\tilde{\Lambda}_k) \end{cases} \tag{7}$$

converges to the fixed point $A_\infty = \text{sign}(A)$, $\Lambda_\infty = \{1\}$.

Proof. From Lemma 2.1 we know that the sequence $\lambda_M(A_k)$ is strictly decreasing and bounded below by 1, thus converging to a limit $\lambda_M^\infty \geqslant 1$, which is a fixed point of $P_\Lambda^\infty = (1 + a_M^\infty)x - a_M^\infty x^3$. However, the only fixed points of P_Λ^∞ are 1, −1 and 0, hence $\lambda_M^\infty = 1$. □

Using Theorem 2.3, it is possible to compute the sign of a matrix from the knowledge of some estimates of the eigenvalues. However, unlike Theorem 2.2, convergence is not obtained within a finite number of steps, although numerical convergence can be attained quickly, depending on the precision of the estimates.

3. An interpolation scheme for the matrix absolute value function

In this section, we consider a diagonalisable matrix A, with eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C} \setminus i\mathbb{R}^*$. Let P_A be its characteristic polynomial, Λ the set of its eigenvalues, and $n_d = \#\Lambda$ the number of distinct eigenvalues. For any polynomial $P \in \mathbb{R}[X]$, the matrix $P(A)$ shares the same eigenvectors as A with respective eigenvalues $P(\lambda_i)$. Hence we deduce the following theorem:

Theorem 3.1 *(Interpolating polynomial). Given a polynomial* $P_{\text{int}} \in \mathbb{R}[X]$, *we have the following equivalences*

$$P_{\text{int}}(A) = |A| \iff P_{\text{int}}(\lambda_i) = |\lambda_i|, \quad \forall \lambda_i \in \Lambda, \tag{8}$$

$$P_{\text{int}}(A) = \text{sign}(A) \iff P_{\text{int}}(\lambda_i) = \text{sign}(\lambda_i), \quad \forall \lambda_i \in \Lambda. \tag{9}$$

The computation of the matrix sign and absolute value functions can thus be regarded as interpolation problems with support points in the set Λ. In the following, we detail the computation of the polynomial P_{int} interpolating the absolute value function over Λ (Eq. (8)). Existence and uniqueness of such a polynomial P_{int} in $\mathbb{R}_{n_d-1}[X]$ is given by the well known Lagrange interpolation theorem (see [12]). However, one can first reduce the interpolation problem (8) on Λ to an interpolation problem on a smaller set. Define

$$\Lambda_+ = \{\lambda_i \in \Lambda, \text{Re}(\lambda_i) \geqslant 0\}, \qquad \Lambda_- = \{\lambda_i \in \Lambda, \text{Re}(\lambda_i) < 0\}, \tag{10}$$

$$n_+ = \#\Lambda_+, \qquad n_- = \#\Lambda_-, \tag{11}$$

$$P_+(X) = \prod_{\lambda_i \in \Lambda_+} (X - \lambda_i), \qquad P_-(X) = \prod_{\lambda_i \in \Lambda_-} (X - \lambda_i). \tag{12}$$

Lemma 3.2 *(Reduced form of the interpolating polynomial). $P_{\text{int}} \in \mathbb{R}[X]$ satisfies* (8) *if and only if there are two polynomials Q_+, $Q_- \in \mathbb{R}[X]$ such that*

$$P_{\text{int}}(X) = X + P_+(X)Q_+(X) = -X + P_-(X)Q_-(X). \tag{13}$$

Proof. This result comes from the fact that from the definition (1), $|\lambda| = \pm\lambda$ depending whether $\lambda \in \Lambda_+$ or $\lambda \in \Lambda_-$. Hence $P_{\text{int}}(\lambda) = |\lambda|$ on Λ if and only if the polynomials $P_{\text{int}}(X) - X$ and $P_{\text{int}}(X) + X$ equal zero respectively on Λ_+ and Λ_-. □

Therefore, in order to determine P_{int}, it is sufficient to build one of the two polynomials Q_+ and Q_- in (13). In the following we assume, up to a change of A into $-A$ that $n_+ \geqslant n_-$. The polynomial Q_+ then has lower degree than Q_- and we now focus on its determination.

Theorem 3.3 *(Q_+ is an interpolating polynomial). $P_{\text{int}}(X) = X + P_+(X)Q_+(X)$ satisfies* (8) *if and only if Q_+ satisfies*

$$Q_+(\lambda_i) = \frac{-2\lambda_i}{P_+(\lambda_i)}, \quad \forall \lambda_i \in \Lambda_-. \tag{14}$$

Proof. For any eigenvalue $\lambda \in \Lambda_+$ we have $P_+(\lambda) = 0$ and thus $P_{\text{int}}(\lambda) = |\lambda|$ is automatically satisfied on Λ_+. For $\lambda \in \Lambda_-$, we have the equivalence $P_{\text{int}}(\lambda) = |\lambda| \iff Q_+(\lambda) = \frac{-2\lambda_i}{P_+(\lambda_i)}$. □

From Theorem 3.3, Q_+ is a Lagrange interpolating polynomial as P_{int}, with a smaller set of support points $\Lambda_+ \subset \Lambda$. The polynomial $Q_+ \in \mathbb{R}_{n_- -1}[X]$ satisfying (14) can be determined using the divided differences method (see for example [12]). In order to evaluate $P_{\text{int}}(A)$, one can first compute the powers $A^2, \ldots, A^{n_+}$ to obtain separately $P_+(A)$ and $Q_+(A)$ as linear combinations, and use Eq. (13) to conclude. The computation of $|A|$ thus requires an overall $\max\{n_+, n_- - 1\}$ matrix multiplications.

Remark 1. The determination of Q_+ through the interpolation problem (14) requires the knowledge of the eigenvalues. However, in the cases where the characteristic polynomial P_A is easily available, there is no actual need to compute the eigenvalues, but only the decomposition $P_A = P_+P_-$. Subtracting the two equations in (13), gives a closed form for Q_+: $P_+Q_+ = -2X \bmod P_-$. This equation can be solved by computing the inverse of P_+ in the quotient space $\mathbb{R}[X]/P_-$ using Euclid algorithm.

4. Application

We implemented the Iteration and Interpolation Schemes in a multiphase flow dynamic code and obtained very satisfactory results [10]. We compared the computational time of the new methods to a straightforward approach based on the computation of the eigenvectors, and the use of formula (2). We considered a classical multiphase flow benchmark, the Ransom water faucet problem [1], and the matrix A had size 8×8 with 6 real distinct eigenvalues. We performed 3 runs of 10 000 computations of the matrix absolute value function using successively the QR method [14], the Iteration and Interpolation Schemes. This corresponds to one time step of a Roe approximate Riemann solver on a 10 000 cell mesh.

Table 1
Computation times for the various methods

Algorithm for the computation of $\vert A\vert$	Eigen decomposition	Iteration Scheme	Interpolation Scheme
CPU time for 10 000 computations	0.27 s	0.35 s	0.16 s

As can be seen on Table 1, the Interpolation Scheme is the fastest and has been chosen as the default algorithm. Note, however, that the results obtained using the eigen decomposition algorithm did not enable us to perform a reliable numerical simulation over several time steps. This comes from the fact that in many cells eigenvectors are nearly parallel. The results given by the QR method are then inaccurate and generate exceptions in the simulation. Thus only the iteration and interpolation approaches developed in this Note allowed the numerical simulation of our two-fluid model.

References

[1] J.M. Delhaye, G. Hewitt, M. Zuber, Numerical Benchmark Tests in Multiphase Science and Technology, vol. 3, Hemisphere, Washington, DC/New York, 1987.
[2] D.A. Drew, S.L. Passman, Theory of Multicomponents Fluids, Springer-Verlag, New York, 1999.
[3] T. Gallouët, J.-M. Hérard, N. Seguin, Numerical modeling of two-phase flows using the two-fluid two-pressure approach, Math. Models Methods Appl. Sci. 14 (5) (2004).
[4] J.M. Ghidaglia, A. Kumbaro, G. Le Coq, On the numerical solution to two fluid models via a cell centered finite volume method, Eur. J. Mech. B/Fluids 20 (6) (2001).
[5] G.H. Golub, C.F. Van Loan, Matrix Computation, third ed., The John Hopkins University Press, 1996.
[6] C.S. Kenney, A.J. Laub, The matrix sign function, IEEE Trans. Automat. Control 40 (8) (1995) 1330–1348.
[7] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[8] J.M. Masella, I. Faille, T. Gallouët, On an approximate Godunov scheme, Int. J. Comput. Fluid Dynamics 12 (1999) 133–149.
[9] M. Ndjinga, Influence of interfacial pressure on the hyperbolicity of the two-fluid model, C. R. Acad. Sci. Paris, Ser. I 344 (2007) 407–412.
[10] M. Ndjinga, A. Kumbaro, P. Laurent-Gengoux, F. De Vuyst, 2006, Fast and robust computation of the matrix absolute value function. Application to Roe solver for the numerical simulation of two-phase flow models, in: 14th International Conference on Nuclear Energy, 17–20 July 2006, Miami, USA.
[11] P.L. Roe, Approximate Riemannn solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981).
[12] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, 2002.
[13] I. Toumi, A. Kumbaro, An approximate linearized Riemann solver for a two-fluid model, J. Comput. Phys. 2 (1996) 124.
[14] J.H. Wilkinson, C. Reinsch, Handbook for Automatic Computation, vol. II, Linear Algebra, Springer-Verlag, New York, 1971.