\[ L^2 – \text{PROJECTION METHOD FOR HYDRODYNAMIC EIGENVALUE PROBLEMS} \]

1. Diana Alina BISTRIAN

**ABSTRACT:** This paper reports a numerical method to compute eigenvalue problems arising in hydrodynamic instability problems. An \( L^2 \) – projection type algorithm is developed assessing both analytical methodology and implementation using symbolic and numerical conversions and applied to a particular case of the trailing vortices developed by delta wings aircrafts. The results of the swirling system instability are pointed out, together with the advantages of using the algorithm in flow control problems.

**KEYWORDS:** hydrodynamic instability, projection method, spectral algorithms

**INTRODUCTION**

Swirling flows behavior has long been an intensive subject of research. The amount of computational resources required to accurately simulate swirling systems is huge, so a complementary stability analysis is a critical requirement to predict the flow dynamics. Pozrikidis [10] offer an introductory course in fluid mechanics, covering the traditional topics in a way that unifies theory, computation, computer programming and numerical simulation. Resiga et al. [12] carried out an experimental and theoretical investigation of the processing helical vortex developed at the outlet of a Francis turbine runner, in order to elucidate the causes of a sudden drop in the draft tube pressure recovery coefficient at a discharge near the best efficiency operating point.

The objective of this paper is to present new instruments that provide relevant conclusions on the stability of swirling fluid systems assessing both an analytical methodology and numerical methods for a spatial stability investigation of the bending modes [1].

The mathematical model governing the linear spatial stability of the swirling fluid system, corresponding to the values of tangential wavenumber \( m \leq 1 \), in operator form is

\[
\begin{align*}
\Lambda_1(k,F,G,H,P) &= \frac{d}{dr}(rG) + kF + G + mH, \\
\Lambda_2(k,F,G,H,P) &= kUG - \omega G + \frac{mWG}{r} + \frac{2WH}{r} - \frac{d}{dr}P, \\
\Lambda_3(k,F,G,H,P) &= kHU - H\omega + \frac{m}{r}[HW + P] + \frac{W}{r}G + \frac{d}{dr}W, \\
\Lambda_4(k,F,G,H,P) &= kFU - F\omega + \frac{mFW}{r} + G\frac{d}{dr}U + kP, \\
\end{align*}
\]

\( \text{where } \{F,G,H,P\}(r) \text{ represent the complex amplitudes of the perturbations, } k \text{ is the complex axial wavenumber, } m \text{ is the integer tangential wavenumber, } \omega \text{ represents the temporal frequency, } U \text{ and } W \text{ represent the axial and the tangential velocity, respectively, both depending only on the radial coordinate } r. \)

For a given real \( \omega \), the system (1)-(4) is equivalent to the complex eigenvalue problem

\[
\Lambda_1 = \Lambda_2 = \Lambda_3 = \Lambda_4 = 0
\]

on the domain \( (0,r_{wall}) \) together with the boundary conditions in axis origin

\[
H = G = 0, \quad F = P = 0
\]

and the wall boundary conditions

\[
\begin{align*}
\frac{2W_{wall}}{r_{wall}}H - P' &= 0, \quad G = 0, \\
r_{wall}H(kU_{wall} - \omega) \pm HW_{wall} &\pm P = 0, \\
r_{wall}F(kU_{wall} - \omega) \pm FW_{wall} + kr_{wall}P &= 0.
\end{align*}
\]

where \( U_{wall} \) and \( W_{wall} \) are the axial and the tangential velocity respectively, calculated at domain limit \( r_{wall} \) and prime denotes the differentiation with respect to the radius.
The hydrodynamic stability model developed in the forthcoming sections of this paper involves spectral differentiation operators derived by means of shifted orthogonal expansions of the perturbation field. The hydrodynamic model presented in Section 2 is developed using an $L^2$ – projection method with operator scheme. The sophisticated boundary conditions motivated the use of this method, suitable for non-periodic problems with complicated boundary conditions.

Dongara [6] used the Chebyshev tau method to examine in detail a variety of eigenvalue problems arising in hydrodynamic stability studies, particularly those of Orr-Sommerfeld type. The orthogonality of Chebyshev functions was used by Bourne [4] to rewrite the differential equations as a generalized eigenvalue problem, assembling a very efficient projection based technique.

The classical approaches imply a transformation of the physical domain onto the standard interval for the definition of the Chebyshev polynomials. For the approach presented in this paper, instead of using classical Chebyshev polynomials, we used shifted Chebyshev polynomials, directly defined on the physical domain of the problem, preserving the orthogonality properties detailed in Section 2.1. The numerical algorithm was developed to work automatically for any number of expansion terms, using symbolic and numeric conversions, the implementation technique being detailed in Section 2.2. In Section 3 the algorithm is validated upon the model of a trailing vortex. An error analysis is performed in Section 4 and the algorithm is efficiently improved by a numerical procedure for eliminating the spurious eigenvalues form the hydrodynamic spectra. The main results of the paper are summarized in Section 5.

$L^2$ – PROJECTION ALGORITHM FOR BENDING MODES – Methodology

Following [4] the eigenvalue problem is transformed into a system of linear equations describing the hydrodynamic context for the cases $m = \pm 1$.

The difference between the classical tau method and the modified version proposed here is given by the selected spaces involved in the approximation process. An appropriate solution is sought in the truncated Chebyshev series form

$$
(F, G, H, P) = \sum_{k=1}^{N} (f_k, g_k, h_k, p_k) \cdot T_k^n,
$$

with $f_k, g_k, h_k, p_k, k = 1..N$ the sets of expansions coefficients to be found. The tau method is an algorithm implying in the first step expanding the residual function as a series of shifted Chebyshev polynomials.

Since the range $[0, r_{wall}]$ is more convenient to use than the standard definition interval of classical Chebyshev polynomials $[-1, 1]$ to discretize our hydrodynamic stability problem, where $r_{wall}$ represents the radial distance to the wall of the computational domain, we map the independent variable $r \in [0, r_{wall}]$ to the variable $\xi \in [-1, 1]$ by the linear transformation

$$
\xi = 2r_{wall}^{-1} - 1 \iff r = r_{wall} (\xi + 1) 2^{-1}.
$$

The shifted Chebyshev polynomials of the first kind $T_n^* (r)$ of degree $n - 1$ in $r$ on $[0, r_{wall}]$ are given by

$$
T_n^* (r) = T_n (\xi) = T_n (2r_{wall}^{-1} - 1).
$$

The shifted Chebyshev class is orthogonal in the Hilbert space $L^2_w (0, r_{wall})$, weighted by $w(r) = (1 - (2r_{wall}^{-1} - 1)^{1/2})^{-1/2}$. They have the next orthogonality properties

$$
\langle T_n^*, T_m^* \rangle_w = 0, \quad n \neq m, \quad n, m = 1..N,
$$

$$
\langle T_n^*, T_n^* \rangle_w = \frac{\pi}{r_{wall}^{n/2}}, \quad \lambda = \begin{cases} 
2 & \text{if } n = 1 \\
4 & \text{if } n = 2..N
\end{cases}
$$

with respect to the inner product $\langle u, v \rangle_w = \int_0^{r_{wall}} w(u \cdot v) dr$.

The differential model is reduced to a finite dimensional algebraic system in the expansion coefficients only by imposing the condition that each equation of the system to be orthogonal with respect to the inner product $\langle A_i, T_j^* \rangle$, $i = 1..4$, $j = 0..N - 2$ in the Hilbert space $L^2_w (0, r_{wall})$. We obtain a set of $4(N - 2)$ linear equations. The eight remaining equations are provided by the boundary conditions applied as side constraints.

Introducing the notations

$$
F^{ij}_{ijkl} = \left( r^k (U^i)^{(j)} (T_i^*)_w \right), \quad F^{ij}_{ijkl} = \left( r^k (W^i)^{(j)} (T_i^*)_w \right)
$$

(13)
with \( d \) the derivation order, the first truncated \( 4(N - 2) \) equations of the hydrodynamic stability model become

\[
k \sum_{j=1}^{N} \left( f_{j} l_{ij}^{U} \right) + g_{r} r_{wall} + g_{s} \frac{2}{r_{wall}} l_{ij}^{U} + \sum_{j=1}^{N} g_{l} \frac{2(j - 1)}{r_{wall}} \left[ \sum_{r=j}^{N} 2 l_{ir}^{U} \right] + \sum_{j=1}^{N} g_{h} \frac{2(j - 1)}{r_{wall}} \left[ \sum_{r=j}^{N} 2 l_{ir}^{U} \right] = \sum_{r=1}^{N} 2 l_{ir}^{U} ,
\]

\[
+ \sum_{j=1}^{N} g_{k} \frac{2(j - 1)}{r_{wall}} \left[ \sum_{r=j}^{N} 2 l_{ir}^{U} \right] + m h_{r} r_{wall} c = 0 ,
\]

\[
k \sum_{j=1}^{N} g_{j} f_{j} r_{ij}^{U} - \omega g_{r} r_{wall} c + m \sum_{j=1}^{N} g_{j} f_{j} r_{ij}^{U} + 2 \sum_{j=1}^{N} h_{j} f_{j} r_{ij}^{U} - p_{r} \frac{2}{r_{wall}} A_{n} - \sum_{j=1}^{N} p_{j} \frac{2(j - 1)}{r_{wall}} \left[ \sum_{r=j}^{N} 2 A_{n} \right] = 0 ,
\]

\[
k \sum_{j=1}^{N} h_{j} f_{j} r_{ij}^{U} - \omega \sum_{j=1}^{N} h_{j} f_{j} r_{ij}^{U} + m \sum_{j=1}^{N} h_{j} f_{j} r_{ij}^{U} + m p_{r} r_{wall} c + \sum_{j=1}^{N} g_{j} \left( f_{j} r_{ij}^{U} + l_{ij}^{U} \right) = 0 ,
\]

\[
k \left( \sum_{j=1}^{N} g_{j} f_{j} r_{ij}^{U} + p_{r} r_{wall} c \right) - \omega f_{j} r_{wall} c + m \sum_{j=1}^{N} g_{j} \left( f_{j} r_{ij}^{U} + l_{ij}^{U} \right) = 0 ,
\]

for \( i = 1..N - 2 \), where the number \( c \) being defined as \( c = \frac{\pi}{2} \), \( i = 1 \) and \( A \in M_{N-2} \) is square \( (N-2) \times (N-2) \) matrix, with \( A_{11} = r_{wall} \pi / 2 \), \( A_{nn} = r_{wall} \pi / 4 \), \( m = 2..N - 2 \), \( A_{nn} = 0 \), \( m \neq n \). Similarly, we translate the boundary conditions in linear equations that complete the system.

For the case \( m = \pm 1 \) that we investigate here, the boundary conditions provide the seven equations set:

\[
\sum_{j=1}^{N} (-1)^{k+1} g_{k} \pm \sum_{j=1}^{N} (-1)^{k+1} h_{k} = 0 ,
\]

\[
\sum_{j=1}^{N} (-1)^{k+1} f_{k} = \sum_{j=1}^{N} (-1)^{k+1} p_{k} = 0 ,
\]

\[
\sum_{j=1}^{N} g_{k} = 0 ,
\]

\[
k \sum_{j=1}^{N} f_{k} + \frac{1}{r_{wall}} \sum_{j=1}^{N} h_{k} = 0 ,
\]

\[
-\frac{2 W_{wall}}{r_{wall}} \sum_{j=1}^{N} h_{k} + p_{2} \frac{2}{r_{wall}} \sum_{j=1}^{N} f_{k} \frac{2(k - 1)}{r_{wall}} \left[ \sum_{j=1}^{N} 2 + \sum_{j=1}^{N} p_{k} \frac{2(k - 1)}{r_{wall}} \sum_{j=1}^{N} 2 + 1 \right] = 0 ,
\]

\[
k U_{wall} \sum_{j=1}^{N} h_{k} + \left( \frac{W_{wall}}{r_{wall}} - \omega \right) \sum_{j=1}^{N} h_{k} \pm \frac{1}{r_{wall}} \sum_{j=1}^{N} p_{k} = 0 ,
\]

\[
k \left( \frac{U_{wall}}{r_{wall}} + \sum_{j=1}^{N} f_{k} \right) + \left( \frac{W_{wall}}{r_{wall}} - \omega \right) \sum_{j=1}^{N} f_{k} = 0 .
\]

For this case, the additional relation in obtained from the second equation of the mathematical model (2), taking the inner product \( \left( A_{2}, T_{N-1}^{*} \right) \)

The eigenvalue problem is written as a system of \( 4N \) equations with the matrix formulation

\[ kM \vec{s} = \vec{M} \vec{s} , \vec{s} = (\vec{f}, \vec{g}, \vec{h}, \vec{p}) , \] with \( \vec{s} = (\vec{f}, \vec{g}, \vec{h}, \vec{p}) , \vec{e} = (\vec{f}, \vec{g}, \vec{h}, \vec{p}) . \]

The method has the obvious advantage that the highest degree of the Chebyshev polynomials multiplying the residual in the method inner-product is only \( N - 2 \).

**IMPLEMENTATION TECHNIQUE USING SYMBOLIC AND NUMERIC CONVERSIONS**

The recurrence relation for \( T_{n}^{*} \) has the form

\[
T_{n}^{*} (r) = 2 \left( 2 r r_{wall}^{-1} - 1 \right) \cdot T_{n-1}^{*} (r) - T_{n-2}^{*} (r) , n = 3,4,\ldots
\]
in which the initial conditions are \( T^*_n (r) = 1, T^*_n (r) = \frac{2r}{r_{wall}} - 1 \). The use of a recurrence relation significantly increases the elapsed time to generate the shifted Chebyshev polynomials. To improve the performance of the numerical algorithm, we introduce in our code the equivalent polynomial relation

\[
T^*_n (r) = \frac{1}{2} \left[ \left( r + \sqrt{r^2 - 1} \right)^n - 1 \right] + \frac{1}{2} \left[ \left( r - \sqrt{r^2 - 1} \right)^n - 1 \right], \quad r = 2r_{wall}^{-1} - 1
\]

(26)

to automatically generate the shifted Chebyshev polynomials \( \{ T^*_n (\xi) \}_{k=0} \) on \([0, r_{wall}]\).

A modified Chebyshev-Gauss grid \( \Xi = \{ \xi_i \}_{j=0}^{j=N} \) in \([-1, 1]\) was constructed

\[
\xi_{j+1} = \cos \left( \pi + \frac{j\pi}{N-1} \right), \quad \xi_{j+1} \in [-1, 1], \ j = 0..N-1,
\]

(27)

mapped into the physical range of our problem by the simple linear transformation

\[ r = r_{wall} (\xi_i + 1) / 2, \ i = 1..N. \]

The collocation nodes clustered near the boundaries diminishing the negative effects of the Runge phenomenon [5]. Another aspect is that the convergence of the interpolation function on the clustered grid towards unknown solution is extremely fast. The numerical algorithm was developed to work automatically for any number of expansion terms, using the routines of a high level language such Matlab.

Using the sym2string [15] function the symbolic integrands are converted into a Matlab equation strings.

The symbol @(x) is concatenated to the integrands by using the Matlab strcat [16] procedure and the integrals are delivered as Matlab functions by eval [17] function.

The integrals are evaluated on the physical domain \([0, r_{wall}]\) using quad [18] Matlab function that numerically evaluates the integrals using an recursive adaptive Simpson quadrature.

We list in Appendix the code sequences for generating the inner products \( I^U_{ijkl} \).

The generalized eigenvalue problem was solved during numerical simulations by an Arnoldi type algorithm implemented in stpms [19] Matlab’s procedure.

The application returns an acceptably accurate approximation of the spectrum and relevant information on perturbation amplitudes for stable or unstable modes, the maximum amplitude of the most unstable mode and the critical distance where the perturbation is the most amplified.

**NUMERICAL RESULTS OF THE INSTABILITY INVESTIGATION**

The basic flow under consideration for the validation of the proposed methods is the Batchelor vortex case or the q-vortex [2] that trails on the tip of each delta wing of the airplanes. The properties of the Batchelor vortex were pointed out in [9] using a shooting method. In order to compare our results with the ones from [9] numerical evaluations of the axial wavenumber \( k \) of the most amplified perturbed wave, obtained for various values of the spectral parameter \( N \). In Table 1 these values are presented in comparison with the ones from Olendradru et al. [9] and the ones obtained using the collocation POD (Proper Orthogonal Decomposition) method in our previous investigation [3].

The numerical results obtained by collocation are better, however, the projection method allowed us to investigate the hydrodynamic models with sophisticated boundary conditions.

The figure 1 illustrate the behavior of the perturbation amplitudes for bending modes, computed with the critical axial wavenumbers listed in Table 1, in comparison with the results obtained in [9].

Following Tadmor [13], when differencing analytic functions using Chebyshev pseudospectral methods, the error committed is expected to decay to zero at an exponential rate. The convergence behavior of the algorithm with respect to the number of expansion terms is shown in Table 2 in comparison with ones obtained using the radial boundary adapted technique in our previous investigations [3].
Clearly the numerical computation costs were less expansive in the projection method approach since the number of terms in the approximations was significantly reduced. In fact, in comparison with the boundary adapted collocation method this number was more than twenty times reduced. In consequence, with a reduced by far computational time, we can obtain accurate results in an acceptable agreement with existing ones.

**ERROR ANALYSIS AND ELIMINATING THE SPURIOUS EIGENVALUES**

Although the projection method is a very efficient technique, the inclusion of the boundary conditions as equations in the system of the generalized eigenvalue problem have been observed to be one case of spurious eigenvalues.

The spurious eigenvalues, which are not always easy to identify, may lead one to a false conclusion regarding the stability of the fluid system, thus the elimination of them is of great importance. These are values returned by the algorithm which do not satisfy the eigenvalue problem. The spurious eigenvalues problems have been the attention of much study recently. Gardner et al. [7] and McFadden et al. [8] describe the tau methods to avoid spurious eigenvalues and in Dongara [6] the occurrence of the spurious eigenvalues is assessed in application to the Benard convection problem.

We implement in our numerical procedure a code sequence that identifies if an eigenvalue of the spectra is spurious or not. First the algorithm provides the entire spectra, then calculates the residual vector of the eigenvalue problem

\[ E^k = k \cdot T \cdot \bar{v} - \Psi \cdot \bar{v}, \quad \bar{v} = (f, g, h, p)^T \]  \hspace{1cm} (28)

with

\[ (f, g, h, p)^T = (\{f_i\}_{i=1}^N, \{g_i\}_{i=1}^N, \{h_i\}_{i=1}^N, \{p_i\}_{i=1}^N)^T \]  \hspace{1cm} (29)

for each eigenvalue of the spectra.

**Figure 1.** Absolute values of eigenfunction amplitudes computed at:

- a) \( m = 1, \quad a = 0, \quad q = 0.7, \quad \omega = 0.0425, \quad k_{\omega} = (0.49, 0) \) using \( N = 7 \) expansion terms.
- b) \( m = -1, \quad a = -1.268, \quad q = 0.6, \quad \omega = -0.78, \quad k_{\omega} = (0.57, -1.358) \) using \( N = 8 \) expansion terms.

**Table 2.** Comparison of the convergence behavior of the algorithm assessing POD collocation method and \( L^2 \)-projection method.

<table>
<thead>
<tr>
<th>N</th>
<th>Axial wavenumber ( k_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>POD collocation [3]</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>165</td>
</tr>
<tr>
<td>( L^2 )-Projection</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

**Figure 2.** Residual of the eigenvalue problem and corresponding histogram.

© copyright FACULTY OF ENGINEERING - HUNEDOARA, ROMANIA 171
A true value of \( k \) must satisfy the eigenvalue problem. We evaluate the \( L^2 \) norm of the vector with respect to a given tolerance \( \varepsilon \). If the condition

\[
\left( \sum \text{abs}(E^k) \right)^{1/2} > \varepsilon
\]

holds, the eigenvalue \( k \) is declared spurious and discarded from the spectra. Figure 2 presents the residual of the eigenvalue problem, solved using the QZ algorithm implemented in the high level computing platform Matlab and the corresponding histogram.

**APPENDIX**

Table 3. The function polcevs.m generates symbolically the \( n^\text{th} \) Chebyshev polynomial defined on domain \([0, r_{\text{wall}}]\).

```
function Tc1 = polcevs(N1,rmax1)
syms x N rmax
Tc1=((2.*x./rmax-1+sqrt((2.*x./rmax-1).^2-1)).^(N-1)+...+(2.*x./rmax-1-sqrt((2.*x./rmax-1).^2-1)).^(N-1))/2;
Tc2=subs(Tc1,N,N1); %replace N by N1
Tc1=subs(Tc2,rmax,rmax1); %replace rmax by rmax1
```

Examples:
```
>> polcevs(1,3)
an = 1
>> polcevs(2,3)
an = 2/3*x1
>> polcevs(3,3)
an = 1/2*(2/3*x1+(2/3*x1)^2-1)/(1/2)^2+1/2*(2/3*x1-(2/3*x1)^2-1)/(1/2)^2
>> polcevs(4,3)
an = 1/2*(2/3*x1+(2/3*x1)^2-1)/(1/2)^3+1/2*(2/3*x1-(2/3*x1)^2-1)/(1/2)^3
>> polcevs(5,3)
an = 1/2*(2/3*x1+(2/3*x1)^2-1)/(1/2)^4+1/2*(2/3*x1-(2/3*x1)^2-1)/(1/2)^4
```

Table 4. Functions to generate symbolically the integrands

```
function Tmn=produsJ(M1,N1,rmax,q)
syms x
Tx=polcevs(N1,rmax);Ty=polcevs(M1,rmax);
W=1./sqrt(1-((2.*x./rmax-1).^2-1));
U=a+exp(-x.^2);
Tmn=U.*Tx.*Ty.*W;
```

Integrals: \( r^1 W(r) \cdot T_n(r) \cdot T_m(r) \cdot w(r) \)

```
function Tmn=produsK(M1,N1,rmax,q)
syms x
Tx=polcevs(N1,rmax);Ty=polcevs(M1,rmax);
W=1./sqrt(1-((2.*x./rmax-1).^2-1));
Ws=1.*exp(-x.^2);/(x.^2);
Tmn=Ws.*(Tx.*Ty.*W);
```

Integrals: \( r^1 W(r) \cdot T_n(r) \cdot T_m(r) \cdot w(r) \)

```
function Tmn=produsL(M1,N1,rmax,q)
syms x
Tx=polcevs(N1,rmax);Ty=polcevs(M1,rmax);
W=1./sqrt(1-((2.*x./rmax-1).^2-1));
rU=x.*(a+exp(-x.^2));
Tmn=rU.*Tx.*Ty.*W;
```

Integrals: \( r^1 U(r) \cdot T_n(r) \cdot T_m(r) \cdot w(r) \)

```
function Tmn=produsM(M1,N1,rmax,q)
syms x
Tx=polcevs(N1,rmax);Ty=polcevs(M1,rmax);
W=1./sqrt(1-((2.*x./rmax-1).^2-1));
Ww=1.*exp(-x.^2);(x.^2);
Tmn=Ww.*Tx.*Ty.*W;
```

Integrals: \( W(r) \cdot T_n(r) \cdot T_m(r) \cdot w(r) \)

```
function Tmn=produsN(M1,N1,rmax,q)
syms x
Tx=polcevs(N1,rmax);Ty=polcevs(M1,rmax);
W=1./sqrt(1-((2.*x./rmax-1).^2-1));
Ww=1.*exp(-x.^2);(x.^2);
Tmn=Ww.*Tx.*Ty.*W;
```

Integrals: \( W(r) \cdot T_n(r) \cdot T_m(r) \cdot w(r) \)
Integrals: \( r \cdot W^r(r) \cdot T^n(m) \cdot T^m(r) \cdot W(r) \)

function Tmn=produsO(M1,N1,rmax,q)

\[
T_x = \text{policevs}(N1,rmax); 
T_y = \text{policevs}(M1,rmax); 
W = 1/\sqrt{1 - (2.*x./rmax - 1)^2}; 
Wdr = x.*(q.*((2.*exp(-x.^2)) + (exp(-x.^2)-1)/(x.^2))); 
Tmn = Wdr.*T_x.*T_y.*W;
\]

Integrals: \( U^r(r) \cdot T^n(m) \cdot T^m(r) \cdot W(r) \)

function Tmn=produsP(M1,N1,rmax,q)

\[
T_x = \text{policevs}(N1,rmax); 
T_y = \text{policevs}(M1,rmax); 
W = 1/\sqrt{1 - (2.*x./rmax - 1)^2}; 
Uder = -2.*x.*exp(-x.^2); 
Tmn = Uder.*T_x.*T_y.*W;
\]

The integrals

\[
I_{ij} = \int_0^{r_{max}} r T_i T_j w dr \\
J_{ij} = \int_0^{r_{max}} UT_i T_j w dr \\
K_{ij} = \int_0^{r_{max}} \frac{W}{r} T_i T_j w dr \\
L_{ij} = \int_0^{r_{max}} r U T_i T_j w dr \\
M_{ij} = \int_0^{r_{max}} W T_i T_j w dr \\
O_{ij} = \int_0^{r_{max}} r W' T_i T_j w dr \\
P_{ij} = \int_0^{r_{max}} U' T_i T_j w dr 
\]

are numerically evaluated by means of function integrala.m, given in Table 4.

Table 4. Function integrala.m.

function valoareint = integrala(functie,lims,lim2)

tt = sym2str(functie);
ft = strcat('@(x)', tt);
fn = eval(f);
valoareint = quad(fn, lims, lim2);

The results are retained in seven square \( N \times N \) matrices. (see Table 5).

Table 5. Sequence for construction of the evaluation matrices.

I=zeros(N); J=zeros(N); K=zeros(N); L=zeros(N); M=zeros(N); O=zeros(N); P=zeros(N);
for i=1:N
for j=1:N
(i,j)=integrala(produs(i,j,rmax),0,rmax);
J(i,j)=integrala(produs(i,j,rmax,a),0,rmax);
K(i,j)=integrala(produsK(i,j,rmax,q),0,rmax);
L(i,j)=integrala(produsL(i,j,rmax,a),0,rmax);
M(i,j)=integrala(produsM(i,j,rmax,q),0,rmax);
O(i,j)=integrala(produsO(i,j,rmax,a),0,rmax);
P(i,j)=integrala(produsP(i,j,rmax,a),0,rmax);
end
end

CONCLUSIONS

This paper reports a numerical spectral approach based on shifted Chebyshev polynomials for numerically solving hydrodynamic eigenvalue problems with sophisticated boundary conditions. Numerical results showed that the use of this method improved the computational time and the obtained critical values of the eigenparameter involved are fairly accurate. The numerical results have been compared with other existing spatial investigations [3] and [9]. The collocation method proved to be more accurate, however the projection method was less expensive with respect to the numerical implementation costs, i.e. numerical results were obtained for a much smaller number of terms in the discretization. The results are very useful not only from their numerical values point of view, but also for their physical interpretation in fluid dynamics in flow control problems.

REFERENCES


15. www.mathworks.com/matlabcentral/fileexchange/authors/30690


