

# Calculation of the Orr-Sommerfeld stability equation for the plane Poiseuille flow

Trinh Anh Ngoc, Tran Vuong Lap Dong

**Abstract**—The stability of plane Poiseuille flow depends on eigenvalues and solutions which are generated by solving Orr-Sommerfeld equation with input parameters including real wavenumber  $\alpha$  and Reynolds number  $R$ . In the research of this paper, the Orr-Sommerfeld equation for the plane Poiseuille flow was solved numerically by improving the Chebyshev collocation method so that the solution of the Orr-Sommerfeld equation could be approximated even and odd polynomial by relying on results of proposition 3.1 that is proved in detail in section 2. The results obtained by this method were more economical than the modified Chebyshev collocation if the comparison could be done in the same accuracy, the same collocation points to find the most unstable eigenvalue. Specifically, the present method needs 49 nodes and only takes 0.0011s to create eigenvalue  $c_1^{49} = 0.23752648\ 505 + 0.00373967\ 557i$  while the modified Chebyshev collocation also uses 49 nodes but takes 0.0045s to generate eigenvalue  $c_1^{49} = 0.23752648\ 526 + 0.00373967\ 555i$  with the same accuracy to eight digits after the decimal point in the comparison with  $c_{exact} = 0.23752648882 + 0.00373967062i$ , see [4], exact to eleven digits after the decimal point.

**Keywords**—Orr-Sommerfeld equation, Chebyshev collocation method, plane Poiseuille flow, even polynomial, odd polynomial

## 1. INTRODUCTION

In this paper, we reconsidered the problem of the stability of plane Poiseuille flow by using odd polynomial and even polynomial to approximate the solution of the Orr-Sommerfeld equation. This approach was also described by Orszag [1], J.J. Dongarra, B. Straughan, D.W. Walker [5] but the goal of this paper was to describe how to

Received 11-01-2018; Accepted on 24-07-2018; Published 20-11-2018

Trinh Anh Ngoc, University of Science, VNU-HCM  
Tran Vuong Lap Dong, University of Science, VNU-HCM;  
Hoang Le Kha high school for the gifted

\*Email: tranvuonglapdong@gmail.com

implement in the efficient approach by using Chebyshev collocation method [6]. We obtained results require considerably less computer time, computational expense and storage to achieve the same accuracy, about finding an eigenvalue which had the largest imaginary part, than were required by the modified Chebyshev collocation method [3].

About the plane Poiseuille flow we wished to study numerically the stream flow of an incompressible viscous fluid through a channel and driven by a pressure gradient in the  $x$  - direction. We used units of the half-width of the channel and units of the undisturbed stream velocity at the centre of the channel to measure all lengths and velocities. In the Poiseuille case, the undisturbed primary flow was  $U(y) = 1 - y^2$  only depended on the  $y$  -coordinate, the side walls were at  $y = \pm 1$ , the Reynolds number was  $R = 1/\nu$ , where  $\nu$  was the kinematic viscosity.

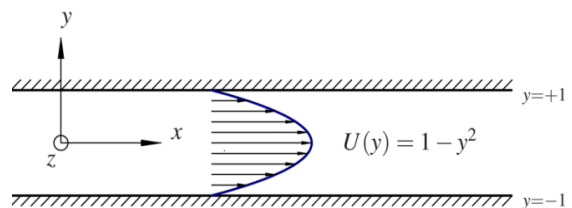


Fig. 1. The plane Poiseuille flow

We assume a two-dimensional disturbance having the form

$$\psi = \phi(y)e^{i\alpha(x-ct)} \quad (1)$$

where  $i$  was the imaginary unit,  $\alpha$  was a real wavenumber,  $c$  was the complex wave velocity. The velocity perturbation equations might be obtained by the linearization of the Navier-Stokes equations which were reducible to the well-known Orr-Sommerfeld for the  $y$ -dependent function  $\phi(y)$ .

$$\frac{i}{\alpha R} (\phi^{(4)} - 2\alpha^2 \phi'' + \alpha^4 \phi) + (1 - y^2)(\phi'' - \alpha^2 \phi) + 2\phi = c(\phi'' - \alpha^2 \phi) \quad (2)$$

With boundary conditions

$$\phi(\pm 1) = \phi'(\pm 1) = 0 \quad (3)$$

According to (1), the real part of the temporal growth rate was  $e^{c_i t}$ ,  $c = c_r + ic_i$ , therefore if there existed  $Im(c) > 0$  then amplitude of the disturbance velocity grew exponentially with time.

## 2. MATERIALS AND METHODS

Proposition 3.1 Suppose that we seek an approximate eigenfunction of (2)-(3) of the form

$$\phi_N(y) = a_N y^N + a_{N-1} y^{N-1} + \dots + a_1 y + a_0.$$

then  $\phi_N(y)$  was an odd function or an even function; corresponding to

$A = \{c | \phi_N(y) \text{ was an odd function}\}$

or  $S = \{c | \phi_N(y) \text{ was an even function}\}$ ,

respectively. Furthermore, if there existed  $c_0 \in A \cap S$  then the approximate eigenfunction of (2)-(3) was the sum of odd function and even function, corresponding to eigenvalue  $c_0$ .

Proof. Assuming that a solution of (2)-(3) could be expanded in a polynomial series as follows

$$\phi(y) = \sum_{k=0}^{+\infty} a_k y^k.$$

Then, the second and fourth derivatives of the function  $\phi(y)$  were

$$\phi''(y) = \sum_{k=0}^{+\infty} (k+1)(k+2)a_{k+2}y^k,$$

$$\phi^{(4)}(y) = \sum_{k=0}^{+\infty} (k+1)(k+2)(k+3)(k+4)a_{k+4}y^k.$$

Hence

$$\begin{aligned} & \frac{i}{\alpha R} (\phi^{(4)} - 2\alpha^2 \phi'' + \alpha^4 \phi) = \\ & \sum_{k=0}^{+\infty} \frac{i}{\alpha R} [(k+1)(k+2)(k+3)(k+4)a_{k+4} \\ & - 2\alpha^2(k+1)(k+2)a_{k+2} + \alpha^4 a_k] y^k. \end{aligned}$$

$$\phi'' - \alpha^2 \phi = \sum_{k=0}^{+\infty} [(k+1)(k+2)a_{k+2} - \alpha^2 a_k] y^k.$$

$$\begin{aligned} (1-y^2)(\phi'' - \alpha^2 \phi) &= \sum_{k=0}^{+\infty} [(k+1)(k+2)a_{k+2} - \\ & ((k-1)k + \alpha^2)a_k + \alpha^2 a_{k-2}] y^k. \end{aligned}$$

We could substitute these into (2), then the right-hand side of (2) was

$$\begin{aligned} & \sum_{k=0}^{+\infty} \left\{ \frac{i}{\alpha R} (k+1)(k+2)(k+3)(k+4)a_{k+4} + \right. \\ & \left. [(1-c)(k+1)(k+2) - \frac{2i\alpha}{R}(k+1)(k+2)] a_{k+2} + \right. \\ & \left. \left[ \frac{i\alpha^3}{R} - (k-1)k - (1+c)\alpha^2 + 2 \right] a_k - \alpha^2 a_{k-2} \right\} y^k \\ & = 0. \quad (4) \end{aligned}$$

Usually, it was not practical to attempt to sum the infinite series in (4), hence we replaced (4) by the finite sum with  $k = 0, 1, \dots, N-4$  and equate coefficients of  $y^k$  for  $k = 0, \dots, N-4$ , we got

$$\begin{aligned} & \frac{i}{\alpha R} (k+1)(k+2)(k+3)(k+4)a_{k+4} + \\ & [(1-c)(k+1)(k+2) - \frac{2i\alpha}{R}(k+1)(k+2)] a_{k+2} + \\ & \left[ \frac{i\alpha^3}{R} - (k-1)k - (1+c)\alpha^2 + 2 \right] a_k - \alpha^2 a_{k-2} = 0, \quad (5) \end{aligned}$$

Beside, the boundary condition (3) were also replaced by the finite sum as expansions in  $a_0, \dots, a_N$ , as follows

$$\sum_{\substack{k=0 \\ k \equiv 0 \pmod{2}}}^N a_k = 0, \quad \sum_{\substack{k=0 \\ k \equiv 0 \pmod{2}}}^N k a_k = 0, \quad (6)$$

$$\sum_{\substack{k=0 \\ k \equiv 1 \pmod{2}}}^N a_k = 0, \quad \sum_{\substack{k=0 \\ k \equiv 1 \pmod{2}}}^N k a_k = 0. \quad (7)$$

Obviously, the system (5)-(7) had  $N+1$  equations for  $N+1$  coefficients, therefore we could find a non-trivial solution,  $\phi_N(y) = a_N y^N + \dots + a_0$ , existing only for certain eigenvalues  $c$ .

But in this proposition, we consider another side that all of the coefficients in the equation (5) were coefficients of odd or even power of  $y$ , hence the system (5)-(7) separated into two sets with no coupling between coefficients  $a_k$  for odd and even  $k$ . Consequently, there existed a set of eigenfunctions with  $a_k = 0$  for  $k$  odd; corresponding to eigenfunction  $v(y)$  was symmetric, i.e.  $v(y) = v(-y)$ . Conversely, the eigenfunctions with  $a_k$  for  $k$  even were antisymmetric, i.e.  $v(y) = -v(-y)$ . We defined two sets  $A = \{c | \phi_N(y) \text{ was an odd function}\}$  and  $S = \{c | \phi_N(y) \text{ was an even function}\}$ . Assume that, there existed  $c_0 \in A \cap S$  and  $\phi_N(y)$ ,  $\bar{\phi}_N(y)$  are respectively odd and even eigenfunction, the

corresponding eigen value  $c_0$  then  $\phi_N(y) = \bar{\phi}_N(y) + \phi_N(y)$  was also eigenfunction of the equations (2)-(3). The proof was complete.

It immediately followed from proposition 3.1 that the only unstable eigenmode of plane Poiseuille flow was symmetric. Thus the following propositions allowed us to approximate eigenfunctions by odd polynomial and even polynomial functions. By relying on results of the Chebyshev method, we defined two basic functions, associated with Chebyshev-Gauss-Lobatto nodes  $y_j = \cos j \pi/N; j = 0, \dots, N$ , to interpolate odd and even polynomial polynomials in  $[-1; 1]$

$$h_k(y) = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{y-y_r}{y_k-y_r} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{y-y_r}{y_{N-k}-y_r}; \quad k = 0, \dots, N, \tag{8}$$

$$e_k \bar{h}_k(y) = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{y-y_r}{y_k-y_r} + \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{y-y_r}{y_{N-k}-y_r}; \quad k = 0, \dots, N, \tag{9}$$

Where

$$e_0 = e_1 = \dots = e_{[N/2]} = 1, \text{ if } N \text{ was odd.} \\ e_0 = e_1 = \dots = e_{N/2-1} = 1; e_{N/2} = 2, \text{ if } N \text{ was even.} \tag{10}$$

Proposition 3.2 Consider basic functions  $\bar{h}_k(y)$  and  $h_k(y)$  which was defined in (8) and (9). Then

(i)  $h_k(y)$  was the odd function and  $h_k(y_j) = \delta_{kj} - \delta_{(N-k)j}$ .

(ii)  $\bar{h}_k(y)$  was the even function and  $e_k \bar{h}_k(y_j) = \delta_{kj} + \delta_{(N-k)j}$ .

where  $\delta_{ij}$  stood for Kronecker delta symbol.

Proof. (i) Obviously, we could prove that  $h_k(y)$  was odd function easily. Indeed, because  $\mathbb{R}$  was the domain of  $h_k(y)$ , therefore  $\forall x_0 \in \mathbb{R}$  then  $-x_0 \in \mathbb{R}$  and

$$h_k(-x_0) = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{-x_0 - y_r}{y_k - y_r} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{-x_0 - y_r}{y_{N-k} - y_r} \\ = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{-x_0 - y_r}{-y_{N-k} - y_r} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{-x_0 - y_r}{-y_k - y_r} \\ = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{x_0 + y_r}{y_{N-k} + y_r} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{x_0 + y_r}{y_k + y_r}$$

$$= \prod_{\substack{r=0 \\ r \neq k}}^N \frac{x_0 - y_{N-r}}{y_{N-k} - y_{N-r}} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{x_0 - y_{N-r}}{y_k - y_{N-r}} \\ = \prod_{\substack{j=0 \\ j \neq N-k}}^N \frac{x_0 - y_j}{y_{N-k} - y_j} - \prod_{\substack{j=0 \\ j \neq k}}^N \frac{x_0 - y_j}{y_k - y_j} \\ = \bar{h}_k(x_0)$$

It remained to check that  $h_k(y_j) = \delta_{kj} - \delta_{(N-k)j}$ . For all  $0 \leq k; j \leq N$ , we had

$$h_k(y_j) = \prod_{\substack{r=0 \\ r \neq k}}^N \frac{y_j - y_r}{y_k - y_r} - \prod_{\substack{r=0 \\ r \neq N-k}}^N \frac{y_j - y_r}{y_{N-k} - y_r} \\ = \delta_{kj} - \delta_{(N-k)j}.$$

(ii) The same as the proof of (i), we got (ii). The proof was complete.

The key feature of this method was that if we assumed that solution  $\phi(y)$  of (2)-(3) was even function then we could approximate  $\phi(y)$  by even polynomial  $\bar{\phi}(y)$  with only half nodes, i.e.  $y_j = \cos \pi j/N, j = 0, \dots, [N/2]$ . We got

$$\bar{\phi}_N(y) = \sum_{k=0}^{[N/2]} \phi_k \bar{l}_k(y),$$

where  $\bar{\phi}_k = \phi_k(y_k), k = 0, \dots, [N/2]$  and

$$\bar{l}_k(y) = \frac{1 - y^2}{1 - y_k^2} \bar{h}_k(y).$$

Conversely, suppose that  $\phi(y)$  was odd function then it was approximated by odd polynomial  $\phi_N(y)$ , which could be written as

$$\phi_N(y) = \sum_{k=0}^{[N/2]} \phi_k l_k(y),$$

where  $\phi_k = \phi_k(y_k), k = 0, \dots, [N/2]$  and

$$l_k(y) = \frac{1 - y^2}{1 - y_k^2} h_k(y).$$

Althought, we also needed that  $\phi''(y_j), \phi^{(4)}(y_j), j = 0, \dots, [N/2]$  in equation (2) should be approximated and expressed as expansions in  $\phi(y_1), \dots, \phi(y_{[N/2]})$  so that we could discrete equation (2) completely. The following proposition would help us to do that.

Proposition 3.3 The Lagrange polynomials associated to the Chebyshev-Guass-Lobatto points were

$$h_j(x) = \prod_{r=0, r \neq j}^N \left( \frac{x - x_r}{x_j - x_r} \right); \quad 0 \leq j \leq N,$$

where  $x_j = \cos \frac{\pi j}{N}; j = 0, \dots, N$ . Define

$d_{ij} = h'_j(x_i)$  then

$$d_{ij} = \frac{c_i(-1)^{i+j}}{c_j(x_i - x_j)}, \quad i \neq j.$$

$$d_{ii} = -\frac{x_i}{2(1 - x_i^2)}, \quad i = 1, 2, \dots, N - 1.$$

$$d_{00} = -d_{NN} = \frac{2N^2 + 1}{6},$$

where  $c_0 = c_N = 2; c_1 = c_2 = \dots = c_{N-1} = 1$ .

Proof. Since this theorem was very long, the reader could see this proof in [6] P.22.

Proposition 3.4 Let

$$u(y) = \sum_{j=0}^{[N/2]} u_j h_j(y) \quad (11)$$

where  $y_j = \cos \frac{\pi j}{N}; j = 0, \dots, [N/2]$ .

if  $\mathbf{u} = [u(y_0) \dots u(y_{[N/2]})]^T$  was the vector of function values, and  $\mathbf{u}^{(k)} = [u^{(k)}(y_0) \dots u^{(k)}(y_{[N/2]})]^T$  was the vector of approximate nodal  $k$ -th order derivatives, obtained by this idea, then

(i) If  $h_j(y) = \bar{h}_j(y)$  then there existed a matrix, say  $P^{(1)} = (p_{ij})_{0 \leq i, j \leq [N/2]}$  with  $e_j p_{ij} = e_j \bar{h}'_j(y_i) = d_{ij} + d_{i(N-j)}$  and  $e_j$  which was defined in (10), such that

$$\mathbf{u}' = P^{(1)} \mathbf{u} \quad (12)$$

(ii) If  $h_j(y) = \underline{h}_j(y)$  then there existed a matrix, say  $Q^{(1)} = (q_{ij})_{0 \leq i, j \leq [N/2]}$  with  $q_{ij} = \underline{h}'_j(y_i) = d_{ij} - d_{i(N-j)}$ , such that

$$\mathbf{u}' = Q^{(1)} \mathbf{u} \quad (13)$$

(iii) If  $h_j(y) = \bar{h}_j(y)$  then we had

$$\mathbf{u}^{(2n)} = (Q^{(1)} P^{(1)})^n \mathbf{u}, \quad \mathbf{u}^{(2n+1)} = P^{(1)} (Q^{(1)} P^{(1)})^n \mathbf{u}.$$

(iv) If  $h_j(y) = \underline{h}_j(y)$  then we had

$$\mathbf{u}^{(2n)} = (P^{(1)} Q^{(1)})^n \mathbf{u}, \quad \mathbf{u}^{(2n+1)} = Q^{(1)} (Q^{(1)} P^{(1)})^n \mathbf{u}.$$

Proof. It was straightforward to deduce the conclusions (i) and (ii) directly from proposition 3.3 and definition of  $\bar{h}_k(y)$  in (8),  $\underline{h}_k(y)$  in (9).

(iii) Let us prove the following assertion by using induction with respect to  $n$ .

$$\mathbf{u}^{(2n)} = (Q^{(1)} P^{(1)})^n \mathbf{u} \quad (14)$$

When  $n = 1$ , it was easy to see that

$$\mathbf{u}'' = Q^{(1)} P^{(1)} \mathbf{u}.$$

Indeed, since  $u(y)$  was even function,  $u'(y)$  should be odd function. Thus  $u'(y)$  could be approximated by the following polynomial in the interval  $[-1; 1]$

$$u'(y) = \sum_{j=0}^{[N/2]} u'_j h_j(y),$$

Applying the conclusion (ii) for  $u'(y)$  and using (12), we got  $\mathbf{u}'' = Q^{(1)} \mathbf{u}' = Q^{(1)} P^{(1)} \mathbf{u}$ .

Suppose that the conclusion in (14) was true for  $n = k$ , we found to show that (14) holded for  $n = k + 1$ . It follow from the induction hypothesis that  $\mathbf{u}^{(2k)} = (Q^{(1)} P^{(1)})^{(k)} \mathbf{u}$ , and since  $u^{(2k)}(y)$  was even function,  $u^{(2k+1)}(y)$  could be approximated by  $u^{(2k)}(y) = \sum_{j=0}^{[N/2]} u^{(2k)}(y_j) \bar{h}_j(y)$ ,

Therefore, applying the conclusion (i) for  $u^{(2k)}(y)$ , we had

$$\mathbf{u}^{2k+1} = P^{(1)} \mathbf{u}^{2k} = P^{(1)} (Q^{(1)} P^{(1)})^{(k)} \mathbf{u}.$$

Similarly, the odd polynomial  $u^{(2k+1)}(y)$  was approximated by  $u^{(2k+1)}(y) = \sum_{j=0}^{[N/2]} u^{(2k+1)}(y_j) \underline{h}_j(y)$ , and just applying the conclusion (ii) for  $u^{(2k+1)}(y)$ , we have  $\mathbf{u}^{2k+2} = Q^{(1)} \mathbf{u}^{2k+1} = (Q^{(1)} P^{(1)})^{(k+1)} \mathbf{u}$ . We completed the proof of the conclusion (14). Finally, to complete the proof of  $\mathbf{u}^{2k+2} = Q^{(1)} \mathbf{u}^{2k+1} = (Q^{(1)} P^{(1)})^{(k+1)} \mathbf{u}$  and (iv). We just repeated the arguments of the proof of (14).

Approximating eigenfunction by even polynomial

We found polynomial  $\bar{\phi}(y)$  was even function which approximate the solution  $\phi(y)$  of form (2)-(3) such that

$$\bar{\phi}(y_j) = \bar{\phi}(y_{N-j}) = \phi(y_j); 1 \leq j \leq [N/2] \tag{15}$$

$$\bar{\phi}'(\pm 1) = \bar{\phi}(\pm 1) = 0 \tag{16}$$

where,  $y_j = \cos \frac{j\pi}{N}; 0, \dots, N$ . The solution of (15)-(16) was given by

$$\bar{\phi}(y) = \sum_{k=1}^{[N/2]} \phi_k \bar{l}_k(y),$$

where  $\phi_k = \bar{\phi}(y_k)$  and  $\bar{l}_k(y) = \frac{1-y^2}{1-y_k^2} \bar{h}_k(y)$ .

Indeed, we have

$$\begin{aligned} \bar{l}_k(y_k) &= \bar{l}_k(y_{N-k}) = 1, \\ \bar{l}_k(y_j) &= 0, y_j \neq y_k. \end{aligned}$$

This implies that the constraint (15) and the condition boundary  $\bar{\phi}(\pm 1) = 0$  are satisfied. Further,

$$\bar{l}'_k(y) = \frac{-2y}{1-y_k^2} \bar{h}_k(y) + \frac{1-y^2}{1-y_k^2} \bar{h}'_k(y); \quad k \neq 0.$$

this implies that  $\bar{\phi}(y)$  satisfy  $\bar{\phi}'(\pm 1) = 0$ .

Next, we use the following  $\bar{\phi}^{(4)}(y), \bar{\phi}''(y)$  to approximate  $\phi^{(4)}(y)$  and  $\phi''(y)$ , respectively

$$\bar{\phi}^{(4)} = \sum_{k=1}^{[N/2]} \left( (1-y^2) \bar{h}_k^{(4)} - 8y \bar{h}_k^{(3)} - 12 \bar{h}_k'' \right) \frac{\phi_k}{1-y_k^2},$$

$$\bar{\phi}'' = \sum_{k=1}^{[N/2]} \left( (1-y^2) \bar{h}_k'' - 4y \bar{h}_k' - 2 \bar{h}_k \right) \frac{\phi_k}{1-y_k^2}.$$

We can then substitute each of these derivative into (2) and we get the following relations

$$\begin{aligned} &\frac{i}{\alpha R} (\bar{\phi}^{(4)} - 2\alpha^2 \bar{\phi}'' + \alpha^4 \bar{\phi})(y_j) + \\ &(1-y_j^2)(\bar{\phi}'' - \alpha^2 \bar{\phi})(y_j) + 2\bar{\phi}(y_j) \\ &= c(\bar{\phi}'' - \alpha^2 \bar{\phi})(y_j); 1 \leq j \leq [N/2] \\ \Leftrightarrow &\left\{ [\bar{\mathbb{D}}_4 - 2\alpha^2 \bar{\mathbb{D}}_2 + \alpha^4 I] \frac{i}{\alpha R} + \right. \\ &\left. \text{Diag}(1-y_j^2)(\bar{\mathbb{D}}_2 - \alpha^2 I) + 2I \right\} U = c(\bar{\mathbb{D}}_2 - \alpha^2 I) U \\ \Leftrightarrow &A\bar{\Phi} = cB\bar{\Phi}. \end{aligned}$$

where

$$\begin{aligned} \bar{\mathbb{D}}_4 &\equiv (\text{Diag}(1-y_j^2) \mathbb{D}^4 - 8 \text{Diag}(y_j) \mathbb{D}^3 \\ &- 12 \mathbb{D}^2) \text{Diag} \left( \frac{1}{1-y_j^2} \right). \end{aligned}$$

$$\begin{aligned} \bar{\mathbb{D}}_2 &\equiv (\text{Diag}(1-y_j^2) \mathbb{D}^2 - 4 \text{Diag}(y_j) \mathbb{D} \\ &- 2I) \text{Diag} \left( \frac{1}{1-y_j^2} \right). \end{aligned}$$

Matrices  $\mathbb{D}_4, \mathbb{D}_3, \mathbb{D}_2$  were defined, respectively, by matrices  $(Q^{(1)}P^{(1)})^2, P^{(1)}Q^{(1)}P^{(1)}, Q^{(1)}P^{(1)}$  which were deleted its first column and first row, where matrices  $Q^{(1)}P^{(1)}$  were determined from the proposition 3.4.

The notation  $\text{Diag}(1-y_j^2)$  was a diagonal matrix with elements  $1-y_j^2, j = 1, 2, \dots, [N/2]$  along its diagonal.

The notation  $\text{Diag}(y)$  was a diagonal matrix with elements  $y_j, j = 1, 2, \dots, [N/2]$ , along its diagonal.

The notation  $\text{Diag} \left( \frac{1}{1-y_j^2} \right)$  was a diagonal matrix with elements  $y_j, j = 1, 2, \dots, [N/2]$  along its diagonal.

$I$  was the  $[N/2] \times [N/2]$  identity matrix.

$$\bar{\Phi} = [\phi_1 \quad \phi_2 \quad \dots \quad \phi_{[N/2]}]^T.$$

$$\begin{aligned} A &= [\bar{\mathbb{D}}_4 - 2\alpha^2 \bar{\mathbb{D}}_2 + \alpha^4 I] \frac{i}{\alpha R} + \\ &\text{Diag}(1-y_j^2)(\bar{\mathbb{D}}_2 - \alpha^2 I) + 2I. \end{aligned}$$

$$B = \bar{\mathbb{D}}_2 - \alpha^2 I.$$

Approximating eigenfunction by odd polynomial

In this case, we find the polynomial  $\bar{\phi}(y_j)$  was odd function which approximate the solution of (2)-(3), such that

$$\bar{\phi}(y_j) = -\bar{\phi}(y_{N-j}) = \phi(y_k); 1 \leq j \leq [N/2] \tag{17}$$

$$\bar{\phi}'(\pm 1) = \bar{\phi}(\pm 1) = 0 \tag{18}$$

where  $y_j = \cos \frac{j\pi}{N}; j = 0, 1, \dots, N$ . The solution of (17)-(18) was given by

$$\bar{\phi}(y) = \sum_{k=1}^{[N/2]} \phi_k l_k(y),$$

where  $\phi_k = \bar{\phi}(y_k)$  and  $l_k(y) = \frac{1-y^2}{1-y_k^2} h_k(y)$

The 4<sup>th</sup> order and 2<sup>nd</sup> order derivative of  $\underline{\phi}(y)$  were then calculated as follows

$$\underline{\phi}^{(4)} = \sum_{k=1}^{[N/2]} \left( (1-y^2)h_k^{(4)} - 8yh_k^{(3)} - 12h_k'' \right) \frac{\phi_k}{1-y_k^2},$$

$$\underline{\phi}'' = \sum_{k=1}^{[N/2]} \left( (1-y^2)h_k'' - 4yh_k' - 2h_k \right) \frac{\phi_k}{1-y_k^2},$$

We could then substitute each of these derivative into (2) and we got the following relations

$$\frac{i}{\alpha R} (\underline{\phi}^{(4)} - 2\alpha^2 \underline{\phi}'' + \alpha^4 \underline{\phi})(y_j) + (1-y_j^2)(\underline{\phi}'' - \alpha^2 \underline{\phi})(y_j) + 2\underline{\phi}(y_j) = c(\underline{\phi}'' - \alpha^2 \underline{\phi})(y_j); 1 \leq j \leq [N/2]$$

$$\Leftrightarrow \left\{ [\mathbb{D}_4 - 2\alpha^2 \mathbb{D}_2 + \alpha^4 I] \frac{i}{\alpha R} + \text{Diag}(1-y_j^2)(\mathbb{D}_2 - \alpha^2 I) + 2I \right\} U = c(\mathbb{D}_2 - \alpha^2 I) U$$

$$\Leftrightarrow A\Phi = cB\Phi.$$

where

$$\mathbb{D}_4 \equiv \left( \text{Diag}(1-y_j^2) \mathbb{D}^4 - 8 \text{Diag}(y_j) \mathbb{D}^3 - 12 \mathbb{D}^2 \right) \text{Diag} \left( \frac{1}{1-y_j^2} \right)$$

$$\mathbb{D}_2 \equiv \left( \text{Diag}(1-y_j^2) \mathbb{D}^2 - 4 \text{Diag}(y_j) \mathbb{D} - 2I \right) \text{Diag} \left( \frac{1}{1-y_j^2} \right).$$

Matrices  $\mathbb{D}_4, \mathbb{D}_3, \mathbb{D}_2$  were defined, respectively, by matrices  $(P^{(1)}Q^{(1)})^2, Q^{(1)}P^{(1)}Q^{(1)}, P^{(1)}Q^{(1)}$  which were deleted its first column and first row if  $N$  was odd and remove more last column and last row, where matrices  $P^{(1)}Q^{(1)}$  were determined from the proposition 3.4.

The notation  $\text{Diag}(1-y_j^2)$  was a diagonal matrix with elements  $1-y_j^2, j=1, \dots, [N/2]$  if  $N$  was odd and  $j=1, \dots, [N/2]-1$  if  $N$  was even.

The notation  $\text{Diag}(y_j)$  was a diagonal matrix with elements  $y_j, j=1, \dots, [N/2]$  if  $N$  was odd and  $j=1, \dots, [N/2]-1$  if  $N$  was even.

The notation  $\text{Diag} \left( \frac{1}{1-y_j^2} \right)$  was a diagonal matrix with elements  $y_j, j=1, \dots, [N/2]$  if  $N$  was odd and  $j=1, \dots, [N/2]-1$  if  $N$  was even.

$I$  was the unit matrix that its size was  $[N/2] \times [N/2]$  if  $N$  was odd and  $([N/2]-1) \times ([N/2]-1)$  if  $N$  was even.

$$\Phi = [\phi_1 \dots \phi_{[N/2]}]^T \text{ if } N \text{ was odd and}$$

$$\Phi = [\phi_1 \dots \phi_{[N/2]-1}]^T \text{ if } N \text{ was even.}$$

$$A = [\mathbb{D}_4 - 2\alpha^2 \mathbb{D}_2 + \alpha^4 I] \frac{i}{\alpha R} +$$

$$\text{Diag}(1-y_j^2)(\mathbb{D}_2 - \alpha^2 I) + 2I.$$

$$B = \mathbb{D}_2 - \alpha^2 I.$$

### 3. RESULTS AND DISCUSSION

In this section, these numerical results were executed on a personal computer, Dell Inspiron N5010 Core i3, CPU 2.40 GHz (4CPUs) RAM 4096MB and we denoted that  $c_1^N$  was the eigenvalue that had the largest imaginary part of all eigenvalues computed using the modified Chebyshev collocation method [3]. The modified Chebyshev collocation method was the Chebyshev collocation method which was modified by L.N so that its numerical condition was smaller than the original method. Trefethen so that its condition number was smaller than the original method, or the present method with  $N+1$  nodes. For  $R=10000, \alpha=1, N=49$ , we saw from Fig.2 that  $Im(c_1^{49}) = 0.0037396755i$ , where  $c_1^{49} = 0.23752648505 + 0.00373967557i$  by using the present method. This value was eight digits when it was compared with the exact eigenvalue

$c_{\text{exact}} = 0.23752648882 + 0.00373967062i$  [4]. Fig. 2 showed the distribution of the eigenvalues.

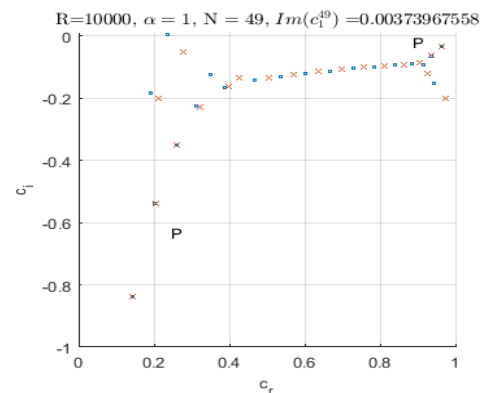


Fig. 2. The spectrum for plane Poiseuille flow when  $\alpha=1, R=10000$ . Open circle (o) = even eigenfunction, cross (x) = odd eigenfunction. The upper right branch and the lower left branch consist of "degenerate" pairs of even and odd eigenvalues

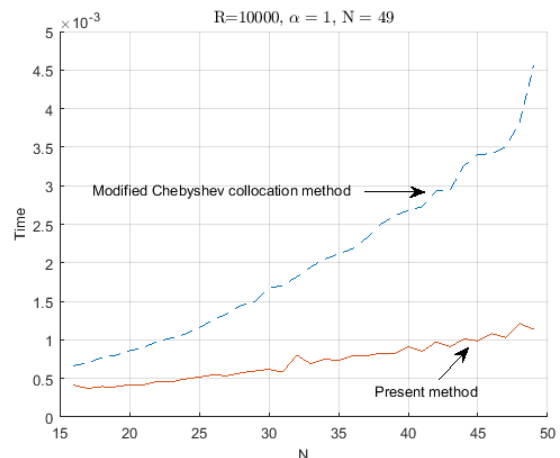
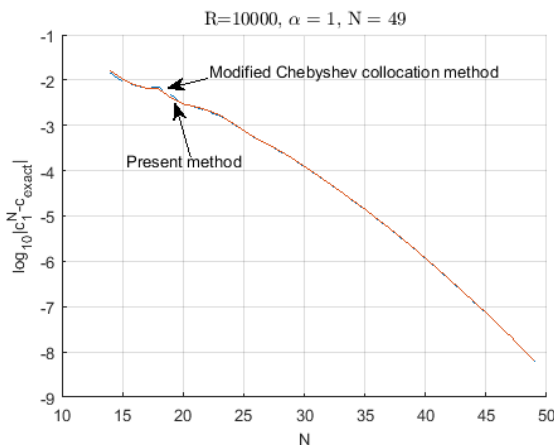
Next, we compared the accuracy of  $Im(c_1^N)$  and execution time between the present method and the Chebyshev collocation method, for  $R = 10000, \alpha = 1$ . Table 1 and Fig. 3 a) showed that although the accuracy of  $Im(c_1^N)$  in both methods was almost the same but we also saw from Table 1 and Fig. 3 B) that the execution time of the present method took less time than the other method with the same nodes. We could explain

this difference by recalling the discussion in Sec. Approximating eigenfunction by even polynomial and odd polynomial with if the same collocation points, then the size of matrices generated by the present method would only be half of the size of matrices generated by the other method, therefore it required considerably less computing time and storage.

**Table 1.** The eigenvalue  $c_1^N$  and executing time generated by the present method and the modified Chebyshev collocation

	The modified C.C method [3]			The present method		
	$c_1^N$	Time (s)	$\log_{10}  c_1^N - c_{exact} $	$c_1^N$	Time (s)	$\log_{10}  c_1^N - c_{exact} $
19	0.2 4233807106+0.0037 6565115i	0.0008	-2.3177	0.2 4156795715+0.003 98342010i	0.0003	-2.3926
24	0.23 842691002+0.003 02873472i	0.0010	-2.9403	0.23 843457669+0.003 01837942i	0.0004	-2.9356
29	0.237 66119611+0.003 60717941i	0.0014	-3.7236	0.237 66838150+0.003 61250703i	0.0005	-3.7200
34	0.2375 4548113+0.0037 2975124i	0.0020	-4.6690	0.2375 4611080+0.0037 2953814i	0.0007	-4.6559
39	0.23752 846688+0.003739 83066i	0.0026	-5.7023	0.23752 847431+0.003739 87797i	0.0008	-5.6997
44	0.237526 55005+0.003739 77835i	0.0032	-6.9068	0.237526 55270+0.003739 78084i	0.0010	-6.8948
49	0.23752648 526+0.00373967 555i	0.0045	-8.2161	0.23752648 505+0.00373967 557i	0.0011	-8.2058

( $c_{exact} = 0.23752648882 + 0.00373967062i$ , see [4], exact to eleven digits after the decimal point)



**Fig. 3.** A)  $\log_{10} |c_1^N - c_{exact}|$  as a function of  $N$ ; B) the computer time to achieve  $c_1^N$  as a function of  $N$  for Orr-Sommerfeld problem (2)-(3). The red solid line belonged to the present method and the blue dash line belonged to the modified Chebyshev collocation method

Fig. 3 showed obviously that the results obtained using both methods were very close, but the present method take less time than the other method.

#### 4. CONCLUSION

The present method, based on a combination of the Chebyshev collocation and the results of proposition 3.1, allowed us to solve the equations (2)-(3) by approximating the solution of this equations by even and odd polynomials, so it was

different from the modified Chebyshev collocation [3]. The numerical results showed that calculating the most unstable by the present method was more economical than the modified Chebyshev collocation about computer time and storage when the comparison could be done for the same accuracy, the same collocation points.

#### REFERENCES

[1]. S.A. Orszag, "Accurate solution of the Orr-Sommerfeld stability equation", *Journal of Fluid Mechanics*, vol. 50, pp. 689-703, 1971.

- [2]. J.T. Rivlin, The Chebyshev polynomials, A Wiley-interscience publication, Toronto, 1974.
- [3]. L.N. Trefethen, Spectral Methods in Matlab, SIAM, Philadelphia, PA, 2000.
- [4]. W. Huang, D.M. Sloan, “The pseudospectral method of solving differential eigenvalue problems”, *Journal of Computational Physics*, vol. 111, 399–409, 1994.
- [5]. J.J. Dongarra, B. Straughan, D.W. Walker, “Chebyshev tau - QZ algorithm methods for calculating spectra of hydrodynamic stability problem”, *Applied Numerical Mathematics*, vol. 22, pp. 399–434, 1996.
- [6]. C.I. Gheorghiu, Spectral method for differential problem, John Wiley & Sons, Inc., New York, 2007.
- [7]. D.L. Harrar II, M.R. Osborne, “Computing eigenvalues of ordinary differential equations”, *Anziam J.*, vol. 44(E), 2003.
- [8]. W. Huang, D.M. Sloan, “The pseudospectral method for third-order differential equations”, *SIAM J. Numer. Anal.*, vol. 29, pp. 1626–1647, 1992.
- [9]. Đ.Đ. Áng, T.A. Ngọc, N.T. Phong, Nhập môn cơ học, Nhà xuất bản Đại học Quốc Gia TP. Hồ Chí Minh, TP. Hồ Chí Minh, 2003.
- [10]. J.A.C. Weideman, L.N. Trefethen, “The eigenvalues of second order spectral differentiations matrices”, *SIAM J. Numer. Anal.*, vol. 25, pp. 1279–1298, 1988.

# Tính toán phương trình Orr-Sommerfeld cho dòng Poiseuille phẳng

Trịnh Anh Ngọc<sup>1</sup>, Trần Vương Lập Đông<sup>1,2</sup>

<sup>1</sup>Trường Đại học Khoa học Tự nhiên, ĐHHQG-HCM

<sup>2</sup>Trường THPT chuyên Hoàng Lê Kha

Tác giả liên hệ: tranvuonglapdong@gmail.com

Ngày nhận bản thảo 11-01-2018; ngày chấp nhận đăng 24-07-2018; ngày đăng 20-11-2018

**Tóm tắt**—Sự ổn định của dòng Poiseuille phẳng phụ thuộc vào các giá trị riêng và hàm riêng mà được tạo ra bằng việc giải phương trình Orr-Sommerfeld với các tham số đầu vào, bao gồm số sóng  $\alpha$  và số Reynold  $R$ . Trong nghiên cứu của bài báo này, phương trình Orr-Sommerfeld cho dòng Poiseuille phẳng có thể được giải số bằng việc cải tiến phương pháp Chebyshev collocation sao cho có thể xấp xỉ được nghiệm của phương trình Orr-Sommerfeld bằng các đa thức nội suy chẵn và lẻ dựa trên các kết quả của mệnh đề 3.1 mà đã được chứng minh một cách chi tiết trong phần 2. Những kết quả số đạt được bằng phương pháp này tiết kiệm hơn về thời gian và lưu trữ so với phương pháp Chebyshev collocation khi cho ra

trị riêng bất ổn định nhất với cùng độ chính xác. Cụ thể, phương pháp hiện tại cần 49 điểm nút và mất 0.0011s để tạo ra trị riêng  $c_1^{49} = 0.23752648505 + 0.00373967557i$  trong khi phương pháp Chebyshev collocation hiệu chỉnh cũng sử dụng 49 điểm nút nhưng cần 0.0045s để tạo ra trị riêng  $c_1^{49} = 0.23752648526 + 0.00373967555i$  với cùng độ chính xác là 8 chữ số thập phân sau dấu phẩy khi so sánh với  $c_{exact}^{49} = 0.23752648882 + 0.00373967062i$  xem [4], chính xác tới 11 chữ số thập phân sau dấu phẩy.

**Từ khóa**—phương trình Orr-Sommerfeld, phương pháp Chebyshev collocation, dòng Poiseuille phẳng, đa thức chẵn, đa thức lẻ