BL FLOW SOLUTION: THE “SHOOTING METHOD”

Method description
The “shooting method” transforms a boundary-value problem into an initial-value problem which is easier to solve using standard numerical techniques.

Example:
Let’s consider the 2\textsuperscript{nd} order ordinary differential equation (ODE) and associated boundary conditions:

\[
\begin{aligned}
\text{ODE1} & \quad \begin{cases} 
  f'' + ff' = 0 \\
  f(0) = 0 \\
  f(L) = A 
\end{cases} \\
\end{aligned}
\]

In this example, \(f\) is known at the boundaries of the domain (i.e., \(x=0\) and \(x=L\)). This problem is referred to as boundary-value problem.

Standard ODE solvers cannot solve this equation directly. In fact, standard ODE solvers require the value of the function and its derivative(s) at a point (initial values) in order to estimate those values a step further at a neighboring point.

Therefore, in order to solve this equation numerically, it is necessary to transform the boundary-value problem into an initial-value problem.

Transformation from boundary- to initial-value problem

The same example can be reformulated as:

\[
\begin{aligned}
\text{ODE2} & \quad \begin{cases} 
  f'' + ff' = 0 \\
  f(0) = 0 \\
  f'(0) = \lambda 
\end{cases} \\
\end{aligned}
\]

where \(\lambda\) is such that the last condition \(f(L) = A\) is also satisfied.

As compared to ODE1, ODE2 consists of the same differential equation, but of a different set of conditions. Instead of fixing the value of the function at two different points (boundary conditions), the value of the function is fixed at the first point and a guess is made regarding the value of its derivative at that point. Because these new conditions are expressed at the same point, they are called initial conditions.

“Shooting method” steps

The shooting method consists of:

1. making a guess for \(\lambda\)
2. solving ODE2 using standard numerical algorithms (e.g., Runge-Kutta)
3. obtain the value \(f(L)\) predicted by the numerical scheme and comparing it to the targeted value \(f(L) = A\)
4- change the value of \( \lambda \) and redo steps 2, 3 and 4 until the predicted value \( f(L) \) matches the targeted value \( f(L) = A \).

The method can be illustrated graphically below.

In this example, the guess \( \lambda_1 \) underestimates the desired boundary value \( f(L) = A \).

The second guess \( \lambda_2 \) overestimates the desired boundary value \( f(L) = A \).

The third guess \( \lambda_3 \) yields a function \( f \) that satisfies the boundary condition \( f(L) = A \).

**Note**: \( \lambda_1 \) and \( \lambda_2 \) are random guesses. The next guess \( \lambda_3 \) can be obtained using a numerical scheme such as the secant method.

**Application to boundary layer equations**

As seen in class, using the Blasius transformation, the boundary layer equations can be written:

\[
2 f'''' + ff''' = 0
\]

With the following boundary conditions:

\[
\begin{align*}
  f'(0) &= 0 \\
  f(0) &= 0 \\
  \lim_{\eta \to \infty} f' &= 1
\end{align*}
\]

**Problem reformulation**

Let's consider the new functions \( y_1, y_2, y_3 \) defined as:

\[
\begin{align*}
y_1 &= f \\
y_2 &= f' \\
y_3 &= f''
\end{align*}
\]

We can express the derivatives \( y_1', y_2', y_3' \) as:
\[ y'_1 = f' = y_2 \]
\[ y'_2 = f'' = y_3 \]
\[ y'_3 = f''' = -\frac{1}{2} f'''' = -\frac{1}{2} y_1 y_3 \]

This is a system of three 1st order ODEs. The boundary conditions expressed in terms of \( y_1, y_2, y_3 \) are:

\[ f(0) = y_1(0) = 0 \]
\[ f'(0) = y_2(0) = 0 \]
\[ f'(\infty) = y_2(\infty) = 1 \]

Therefore, we're missing a boundary condition for \( y_3 \). We'll assume as a first guess that:

\[ f''(0) = y_3(0) = \lambda \]

Let's choose \( \lambda = 1 \) as a first guess. Using Matlab and this first guess, it is now possible to solve for the system of equations (b). The following plot shows the resulting function \( f'(\eta) \) using \( \lambda = 1 \) as a first guess:

It is clear on this graph that the last boundary condition (i.e., \( f'(\infty) = y_2(\infty) = 1 \)) is not satisfied. This demonstrates that the guess we made is not correct. Following an iterative process, it is possible to refine the value of the guess so that the boundary condition \( f'(\infty) = y_2(\infty) = 1 \) is
satisfied. The exact value of $\lambda$ satisfying this boundary condition is $\lambda = 0.33204$. The solution $f'(\eta)$ is plotted on the following graph: