PJS-04-00
June 1 2001

# The Numerov Algorithm

P. J. Salzman
*Department of Physics*
*University of California*
*Davis, CA 95616*

## Abstract

This article discusses the use, derivation and application of the Numerov algorithm. I wrote it for my Physics 105AL students.

While the Numerov Algorithm is a higher order routine than 4th order Runge-Kutta, and is computationally easier for both the programmer and the computer, it's only applicable to a certain class of equations. The Numerov Algorithm is used to solve 2nd order ordinary differential equations that are missing a 1st order derivative term. However, this is one of the most important equations of physics; it happens to be the Helmholtz and Laplace equations in 1 dimension.

Furthermore, in many cases it's possible to transform a general 2nd order ODE into one missing its 1st order derivative, making the Numerov algorithm very widely applicable.

# 1 Background information

Most computational physicists will tell you that the fourth order Runge-Kutta algorithm is the best method to use for numerically solving ODE's. And for *general* ODE's, it is.

However, when the ODE doesn't contain a first order derivative, you can use the Numerov method which is an order more accurate than fourth order Runge-Kutta. But if that wasn't good enough, the Numerov algorithm requires fewer computations and is easier to program. And if *that* isn't good enough, it's possible to transform many general 2nd order equations into one without a 1st order derivative term (see Appendix A for details).

The Numerov Algorithm is designed to numerically solve ODE's of the form:[1]

$$\frac{d^2y}{dx^2} + k^2(x)\,y(x) \;=\; 0 \tag{1}$$

This is a very important ODE in physics. Perhaps the most important. Among other things, this is Schrödinger's time independent equation. For example, $k^2(x)$ for a non-infinite squre well is given by:

$$k^2 \equiv \frac{2m}{\hbar^2}\begin{cases} E + V_0 & |x| < a \\ E & |x| > a \end{cases} \tag{2}$$

In this example, the potential is "constant" and has no explict dependence on $x$, so $k^2(x)$ has no explicit $x$ dependence either, but in general, $k^2(x)$ can be any crazy function of $x$ you want.

# 2 The Algorithm

The equation we're trying to obtain a solution for is:

$$\frac{d^2y}{dx^2} + k^2(x)\,y(x) \;=\; 0 \tag{3}$$

Let the operator $1 + \frac{h^2}{12}\frac{d^2}{dx^2}$ operate on this equation. We'll call this the "modified ODE", but its solutions are the same as the solutions to eqn(3):

[1] The Numerov algorthm is more general than this. It can be extended to solve a more general ODE:

$$\frac{d^2y}{dx^2} + k^2(x)\,y(x) + s(x) = 0$$

but for the current Phys 105AL assignment, you don't need to use this more general equation.

$$\left[1 + \frac{h^2}{12}\frac{d^2}{dx^2}\right]\left(\frac{d^2y}{dx^2} + k^2(x)\,y(x)\right) \;=\; 0$$

$$\frac{h^2}{12}\frac{d^4y}{dx^4} + \frac{h^2}{12}\frac{d^2}{dx^2}\left[k^2(x)\,y(x)\right] + \frac{d^2y}{dx^2} + k^2(x)\,y(x) \;=\; 0 \qquad (4)$$

Computers don't know how to take derivatives, so the first thing we'll do is replace the 2nd and 4th order derivatives by something we can calculate on a computer. To get rid of the derivatives, write down the Taylor series for $y$ centred around $x + h$ where h is small:

$$y(x+h) \;=\; y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \frac{h^3}{3!}y'''(x) + \frac{h^4}{4!}y''''(x) + \cdots$$

And then expand $y(x)$ about $x - h$ (simply let $h \to -h$):

$$y(x-h) \;=\; y(x) - hy'(x) + \frac{h^2}{2!}y''(x) - \frac{h^3}{3!}y'''(x) + \frac{h^4}{4!}y''''(x) + \cdots$$

Add these two equations and solve for $y''(x)$. Note that all terms with odd powers of $h$ sum to zero.

$$y''(x) \;=\; \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} - \frac{h^2}{12}y''''(x) + \mathcal{O}(h^6)$$

Plug this expression for $y''(x)$ into the modified ODE, eqn(4). The 4th order derivatives sum to zero:

$$\frac{h^2}{12}\frac{d^4y}{dx^4} + \frac{h^2}{12}\frac{d^2}{dx^2}\left[k^2y(x)\right] + \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} - \frac{h^2}{12}\frac{d^4y}{dx^4} + k^2(x)\,y(x) \;=\; 0$$

$$\frac{h^2}{12}\frac{d^2}{dx^2}\left[k^2y(x)\right] + \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + k^2(x)\,y(x) \;=\; 0 \qquad (5)$$

To handle the $k^2$ dependence on $x$, we'll approximate the 2nd derivative of $k^2(x)y(x)$ as[2]:

$$\frac{d^2[k^2(x)y(x)]}{dx^2} \;\approx\; \frac{\left[k^2(x+h)y(x+h) - k^2(x)y(x)\right] + \left[k^2(x-h)y(x-h) - k^2(x)y(x)\right]}{h^2}$$

[2]You can derive this using the Taylor series and derivative product rule.

Plugging this this expression into eqn(5) and rearranging gives the Numerov algorithm (if you're following along with the calculation, there's a little bit of algebra here):

$$y(x+h) \;=\; \frac{2\left[1-\frac{5}{12}h^2k^2(x)\right]y(x)-\left[1+\frac{h^2}{12}k^2(x-h)\right]y(x-h)}{1+\frac{h^2}{12}k^2(x+h)} \tag{6}$$

We can phrase this in terms of discrete indices, $x = jh$:

$$y_{j+1} \;=\; \frac{2\left[1-\frac{5}{12}h^2k_j^2\right]y_j-\left[1+\frac{h^2}{12}k_{j-1}^2\right]y_{j-1}}{1+\frac{h^2}{12}k_{j+1}^2} \tag{7}$$

Using this algorithm, we can compute $y_j$ from the values of $y_{j-1}$ and $y_{j-2}$. Of course, this means you either need to know the values of $y_0$ and $y_1$ and go forward through the lattice or know the values of $y_{N-1}$ and $y_{N-2}$ and go backwards through the lattice.

Having to know the first or last 2 values of $y(x)$ shouldn't be surprising. After all, this is a 2nd order ODE, and as you know, 2nd order equations need 2 boundary conditions. You obtain the boundary conditions by considering the physical system at hand.

## Appendix A  Equations With A First Order Derivative

Actually, it *is* possible to use the Numerov algorithm with 2nd order equations of the form:

$$\frac{d^2y}{dx^2} + p(x)\,\frac{dy}{dx} + q(x)\,y(x) + s(x) = 0 \tag{8}$$

but it requires a little work. Think back to your basic calculus class when you learned conic sections. Conic sections that were aligned with the axes were relations between $x^2$ and $y^2$. However, when an $x$ or $y$ term entered the equation, the effect was to "tilt" the conic section with respect to the axes. You learned that, by a suitable transformation, tilted conic sections could be put into a simplified form involving only $x^2$ and $y^2$. The interpretation of these magic transformations was that they tilted the axes by just the right angle to "un-tilt" the conic section.

Well, that's what we're going to do here. The difference is that we're not going to tilt the axes, but to transform the coordinate plane in a more complicated manner. But the result is not much different from what you did with conic sections.

In eqn(8), make the transformation $y'' = h(x)\,z(x)$, and group terms by derivatives of $z$:

$$\frac{d^2z}{dx^2}\,h + \frac{dz}{dx}\,\big(2h' + ph\big) + z(x)\,\big(h'' + ph' + qh\big) + s(x) \;=\; 0 \tag{9}$$

So clearly, we can eliminate the first derivative term, $z'(x)$, if we solve the equation:

$$\frac{dh}{dx} = -\frac{1}{2}p(x)\,h(x) \tag{10}$$

Thus, by solving a 1st order ODE[3] we can use the Numerov algorithm with equations that contain a first derivative term. When we divide the equation through by the coefficient of $z''(x)$, we're left with an equation of the form:

$$\frac{d^2z}{dx^2} + A(x)\,z(x) + B(x) = 0$$

which can then be numerically solved with the Numerov algorithm. We already know $h(x)$. Once we solve this equation, we'll know $z(x)$. To get the original desired function $y(x)$, we only need to compute $y(x) = h(x)\,z(x)$.

Why does this "trick" work? It works because our transformation has two degrees of freedom: $h(x)$ and $z(x)$. We can constrain $h(x)$, while leaving $z(x)$ unconstrained, and therefore, it remains a dependent variable.

[3] In principle, the equation is always solvable. However, depending on what $p(x)$ is, it may or may not be "worth it".