Numerical Solutions to Differential Equations

Lecture Notes #20 — Finite Difference Methods

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015

Outline

1. A Different Approach: Finite Difference Methods
   - Motivation
   - Derivation
   - Second Order Linear ODEs
2. Accuracy of Solutions
   - Improvement Strategy: Richardson Extrapolation
   - Improvement Strategy: Better Finite Differences
3. Some Remaining Issues
   - Boundary Conditions
   - BCs... and Accuracy

Finite Difference Methods

- Shooting methods converge very rapidly **when they work**, but convergence cannot be guaranteed. They tend to be **unstable** (especially when shooting with many variables.) — Remember that Newton's method has a small basin of attraction (*i.e* it only converges for "good enough" initial guesses.)
- **Finite Difference Methods** have better (more predictable) stability characteristics. The downside is that they generally require **more computation** to obtain a specified accuracy.
- We replace the derivatives in the equation with difference approximations, and thus convert the ODE into a set of simultaneous algebraic equations.
- The set of algebraic equations is linear (non-linear) if the ODE is linear (nonlinear).
- Finite Difference Methods can be applied directly to higher order ODEs — no need to convert to a system of 1st order ODEs.

We can derive finite difference approximation to derivatives using two methods:

[1] By differentiating the **Lagrange Interpolating Polynomial** of appropriate order, at the desired grid-point(s). [Math 541]

[2] By Taylor expansions (and matching coefficients), *e.g.*

$$
\begin{aligned}
y_{n+1} = y(x_{n+1}) &= y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \ldots \\
y_{n-1} = y(x_{n-1}) &= y(x_n) - hy'(x_n) + \frac{h^2}{2}y''(x_n) - \frac{h^3}{6}y'''(x_n) + \ldots \\
\hline
y_{n+1} - y_{n-1} &= 2hy'(x_n) + \frac{h^3}{3}y'''(x_n) \\
\frac{\mathbf{y_{n+1} - y_{n-1}}}{\mathbf{2h}} &= \mathbf{y'(x_n)} + \mathcal{O}(\mathbf{h^2})
\end{aligned}
$$

Forward Differences, truncation error $\mathcal{O}(h)$

$$
\begin{aligned}
y_n' &\approx [y_{n+1} - y_n]/h \\
y_n'' &\approx [y_{n+2} - 2y_{n+1} + y_n]/h^2 \\
y_n''' &\approx [y_{n+3} - 3y_{n+2} + 3y_{n+1} - y_n]/h^3 \\
y_n'''' &\approx [y_{n+4} - 4y_{n+3} + 6y_{n+2} - 4y_{n+1} + y_n]/h^4
\end{aligned}
$$

Backward Differences, truncation error $\mathcal{O}(h)$

$$
\begin{aligned}
y_n' &\approx [y_n - y_{n-1}]/h \\
y_n'' &\approx [y_n - 2y_{n-1} + y_{n-2}]/h^2 \\
y_n''' &\approx [y_n - 3y_{n-1} + 3y_{n-2} - y_{n-3}]/h^3 \\
y_n'''' &\approx [y_n - 4y_{n-1} + 6y_{n-2} - 4y_{n-3} + y_{n-4}]/h^4
\end{aligned}
$$

Central Differences, truncation error $\mathcal{O}(h^2)$

$$
\begin{aligned}
y_n' &\approx [y_{n+1} - y_{n-1}]/2h \\
y_n'' &\approx [y_{n+1} - 2y_n + y_{n-1}]/h^2 \\
y_n''' &\approx [y_{n+2} - 2y_{n+1} + 2y_{n-1} - y_{n-2}]/2h^3 \\
y_n'''' &\approx [y_{n+2} - 4y_{n+1} + 6y_n - 4y_{n-1} + y_{n-2}]/h^4
\end{aligned}
$$

Forward Differences, truncation error $\mathcal{O}(h^2)$

$$
\begin{aligned}
y_n' &\approx [-y_{n+2} + 4y_{n+1} - 3y_n]/2h \\
y_n'' &\approx [-y_{n+3} + 4y_{n+2} - 5y_{n+1} + 2y_n]/h^2 \\
y_n''' &\approx [-3y_{n+4} + 14y_{n+3} - 24y_{n+2} + 18y_{n+1} - 5y_n]/2h^3 \\
y_n'''' &\approx [-2y_{n+5} + 11y_{n+4} - 24y_{n+3} + 26y_{n+2} - 14y_{n+1} + 3y_n]/h^4
\end{aligned}
$$

Backward Differences, truncation error $\mathcal{O}(h^2)$

$$\begin{aligned}
y_n' &\approx [3y_n - 4y_{n-1} + y_{n-2}]/2h \\
y_n'' &\approx [2y_n - 5y_{n-1} + 4y_{n-2} - y_{n-3}]/h^2 \\
y_n''' &\approx [5y_n - 18y_{n-1} + 24y_{n-2} - 14y_{n-3} + 3y_{n-4}]/2h^3 \\
y_n'''' &\approx [3y_n - 14y_{n-1} + 26y_{n-2} - 24y_{n-3} + 11y_{n-4} - 2y_{n-5}]/h^4
\end{aligned}$$

Central Differences, truncation error $\mathcal{O}(h^4)$

$$\begin{aligned}
y_n' &\approx [-y_{n+2} + 8y_{n+1} - 8y_{n-1} + y_{n-2}]/12h \\
y_n'' &\approx [-y_{n+2} + 16y_{n+1} - 30y_n + 16y_{n-1} - y_{n-2}]/12h^2 \\
y_n''' &\approx [-y_{n+3} + 8y_{n+2} - 13y_{n+1} + 13y_{n-1} - 8y_{n-2} + y_{n-3}]/8h^3 \\
y_n'''' &\approx [-y_{n+3} + 12y_{n+2} - 39y_{n+1} + 56y_n - 39y_{n-1} + 12y_{n-2} \\
&\qquad - y_{n-3}]/6h^4
\end{aligned}$$

We consider the problem

$$\begin{array}{lr} y''(x) + p(x)y'(x) + q(x)y(x) = r(x), \quad x \in [a, b] & \text{(ODE)} \\ y(a) = y_a,\ y(b) = y_b & \text{(BCs)} \end{array}$$

If we use the second-order accurate finite difference approximations

$$y''(x_n) \approx \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2}, \quad y'(x_n) \approx \frac{y_{n+1} - y_{n-1}}{2h}$$

we get the following set of algebraic equations

$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + p_n \frac{y_{n+1} - y_{n-1}}{2h} + q_n y_n = r_n \quad \textbf{(ALG)}$$

where we have used the **notation**

$$\begin{array}{c} p_n = p(x_n),\ q_n = q(x_n),\ r_n = r(x_n),\ y_n = y(x_n) \\ x_n = a + nh,\ n = 0, 1, \ldots, N, \quad N = (b - a)/h \end{array}$$

Note that **(ALG)** only makes sense in the interior, *i.e.* for $n = 1, 2, \ldots, (N-1)$, and **not** at $n = 0$, and $n = N$:

$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + p_n \frac{y_{n+1} - y_{n-1}}{2h} + q_n y_n = r_n \quad \textbf{(ALG)}$$

the boundary conditions **(BCs)**

$$y_0 = y_a, \quad y_N = y_b, \qquad (\text{at } n = 0, \text{ and } n = N)$$

close the system — we have $(N-1)$ unknowns $\{y_1, y_2, \ldots, y_{N-1}\}$ and $(N-1)$ equations.

With a little bit of "massage" (ALG) becomes

$$\left[1 + \frac{h}{2} p_n\right] y_{n+1} + \left[h^2 q_n - 2\right] y_n + \left[1 - \frac{h}{2} p_n\right] y_{n-1} = h^2 r_n \quad \textbf{(ALG')}$$

Note that (ALG')

$$\left[1+\frac{h}{2}p_n\right]y_{n+1}+\left[h^2q_n-2\right]y_n+\left[1-\frac{h}{2}p_n\right]y_{n-1}=h^2r_n \quad \text{(ALG')}$$

contains y_0 when $n=1$, and y_N when $n=(N-1)$, *i.e.*

$$\left[1+\frac{h}{2}p_1\right]y_2+\left[h^2q_1-2\right]y_1+\left[1-\frac{h}{2}p_1\right]\mathbf{y_0}=h^2r_1$$
$$\left[1+\frac{h}{2}p_{N-1}\right]\mathbf{y_N}+\left[h^2q_{N-1}-2\right]y_{N-1}+\left[1-\frac{h}{2}p_{N-1}\right]y_{N-2}=h^2r_{N-1}$$

since these values are known (Boundary Conditions), we move them to the right-hand-side:

$$\left[1+\frac{h}{2}p_1\right]y_2+\left[h^2q_1-2\right]y_1=h^2r_1-\left[1-\frac{h}{2}p_1\right]\mathbf{y_a}$$
$$\left[h^2q_{N-1}-2\right]y_{N-1}+\left[1-\frac{h}{2}p_{N-1}\right]y_{N-2}=h^2r_{N-1}-\left[1+\frac{h}{2}p_{N-1}\right]\mathbf{y_b}$$

We have the following equations:

$$
\begin{aligned}
\left[h^2 q_1 - 2\right] y_1 + \left[1 + \frac{h}{2} p_1\right] y_2 &= h^2 r_1 - \left[1 - \tfrac{h}{2} p_1\right] \mathbf{y_a} \\
\left[1 - \frac{h}{2} p_n\right] y_{n-1} + \left[h^2 q_n - 2\right] y_n + \left[1 + \frac{h}{2} p_n\right] y_{n+1} &= h^2 r_n \quad n = 2, 3, \ldots, (N-2) \\
\left[1 - \frac{h}{2} p_{N-1}\right] y_{N-2} + \left[h^2 q_{N-1} - 2\right] y_{N-1} &= h^2 r_{N-1} - \left[1 + \tfrac{h}{2} p_{N-1}\right] \mathbf{y_b}
\end{aligned}
$$

This is a matrix equation, $A\tilde{\mathbf{y}} = \tilde{\mathbf{b}}$, where...

$$
\tilde{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix}, \quad
\tilde{\mathbf{b}} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N-2} \\ b_{N-1} \end{bmatrix} =
\begin{bmatrix} h^2 r_1 - \left[1 - \frac{h}{2} p_1\right] \mathbf{y_a} \\ h^2 r_2 \\ \vdots \\ h^2 r_{N-2} \\ h^2 r_{N-1} - \left[1 + \frac{h}{2} p_{N-1}\right] \mathbf{y_b} \end{bmatrix}
$$

We have the following equations:

$$\begin{aligned}
\left[h^2 q_1 - 2\right] y_1 + \left[1 + \frac{h}{2} p_1\right] y_2 &= h^2 r_1 - \left[1 - \tfrac{h}{2} p_1\right] \mathbf{y_a} \\
\left[1 - \frac{h}{2} p_n\right] y_{n-1} + \left[h^2 q_n - 2\right] y_n + \left[1 + \frac{h}{2} p_n\right] y_{n+1} &= h^2 r_n \quad n = 2, 3, \ldots, (N-2) \\
\left[1 - \frac{h}{2} p_{N-1}\right] y_{N-2} + \left[h^2 q_{N-1} - 2\right] y_{N-1} &= h^2 r_{N-1} - \left[1 + \tfrac{h}{2} p_{N-1}\right] \mathbf{y_b}
\end{aligned}$$

This is a matrix equation, $A\tilde{\mathbf{y}} = \tilde{\mathbf{b}}$, where...

$$A = \begin{bmatrix} d_1 & s_1^+ & & & \\ s_2^- & d_2 & s_2^+ & & \\ & \ddots & \ddots & \ddots & \\ & & s_{N-2}^- & d_{N-2} & s_{N-2}^+ \\ & & & s_{N-1}^- & d_{N-1} \end{bmatrix}, \quad \left\{ \begin{array}{ll} d_n = h^2 q_n - 2 & n = 1, 2, (N-1) \\ s_n^+ = 1 + \frac{h}{2} p_n & n = 1, 2, \ldots, (N-2) \\ s_n^- = 1 - \frac{h}{2} p_n & n = 2, 3, \ldots, (N-1) \end{array} \right.$$

Code: 2nd Order ODE/BVP Solver — Segment #1

```
% Solve 2nd Order ODE/BVPs. --- Octave code [www.octave.org]
%
% y''(x) + p(x)y'(x) + q(x)y(x) = r(x), a<=x<=b
% BC: y(a) = ya,  y(b) = yb
clear all

% Boundary Conditions
a = 1;  ya = 1;
b = 2;  yb = 2;

% Number of interior grid points
N = 64;

% Grid size
h = (b-a)/(N+1);

% The grid
x = ((a+h):h:(b-h))';
```

Code: 2nd Order ODE/BVP Solver Segment #2

```
function p = p(x)
  p = 2./x;
endfunction

function q = q(x)
  q = 2./(x.^2);
endfunction

function r = r(x)
  r = sin(log(x))./(x.^2);
endfunction

% Set up the linear system Ay=b

% the right-hand-side
rhs = h^2*r(x);
rhs(1) = rhs(1) - (1-h/2*p(x(1)))*ya;
rhs(N) = rhs(N) - (1+h/2*p(x(N)))*yb;
```

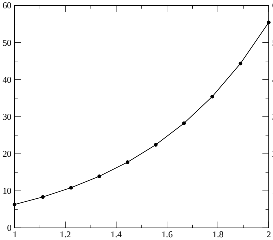
Code: 2nd Order ODE/BVP Solver — Segment #3

```
% the diagonal of the matrix A
d = h^2*q(x(1:N))-2;

% the superdiagonal of A
sp = 1 + h/2*p(x(1:(N-1)));

% the subdiagonal of A
sm = 1 - h/2*p(x(2:N));

% Assemble the matrix
A = diag(sm,-1) + diag(d,0) + diag(sp,1);

% Solve
y = A\rhs;

xs = [a; x; b];
ys = [ya; y; yb];
```

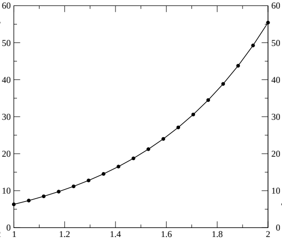
Example #1

The code solves the following BVP:

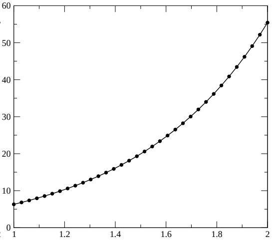
$$y''(x) - y'(x) + y(x) = 3e^{2x} - 2\sin(x)$$
$$y(1) = 6.308447, \quad y(2) = 55.430436$$



8 interior nodes



16 interior nodes



32 interior nodes

Accuracy of the Solutions

Since we used second-order accurate finite difference approximations to the derivatives, the numerical solution is second order accurate.

If (when) we need higher order accuracy, there are two ways to proceed:

[1] (Pointwise) Richardson Extrapolation.

[2] More accurate finite difference approximations.

Improving the Accuracy: Richardson Extrapolation

If we are using a symmetric second order accurate method, then at each grid point we have

$$y_i^{\text{numerical}}(h) = y_i^{\text{exact}} + \mathcal{C}h^2 + \mathcal{O}(h^4)$$

Due to *symmetry*, there are no h^{2k+1} terms in the error expansion.

We can combine two numerical solutions (at the same point)

$$\frac{4y_i^{\text{num}}(h/2) - y_i^{\text{num}}(h)}{3} = y_i^{\text{e}} + \frac{4\mathcal{C}(h/2)^2 - \mathcal{C}(h)^2}{3} + \mathcal{O}(h^4) = y_i^{\text{e}} + \mathcal{O}(h^4)$$

The error is now $\sim \mathcal{O}(h^4)$!

The procedure can be continued — see the review on the following three slides.

What it is: A method for generating high-accuracy results using low-order formulas (or results).

Applicable: When the approximation technique has an error term of predictable form, *e.g.*

$$M - N_j(h) = \sum_{k=j}^{\infty} E_k h^k,$$

where M is the unknown value we are trying to approximate, and $N_j(h)$ the approximation (which has an error $\mathcal{O}(h^j)$.)

Consider:

$$M - N_1(h) = \sum_{k=1}^{\infty} E_k h^k,$$

and

$$M - N_1(h/2) = \sum_{k=1}^{\infty} E_k \frac{h^k}{2^k}.$$

If we let $N_2(h) = 2N_1(h/2) - N_1(h)$, then

$$M - N_2(h) = \underbrace{2E_1 \frac{h}{2} - E_1 h}_{\mathbf{0}} + \sum_{k=2}^{n} E_k^{(2)} h^k,$$

where

$$E_k^{(2)} = E_k \left(\frac{1}{2^{k-1}} - 1 \right).$$

We can play the game again, and combine $N_2(h)$ with $N_2(h/2)$ to get a third-order accurate approximation, etc.

$$N_3(h) = \frac{4N_2(h/2) - N_2(h)}{3} = N_2(h/2) + \frac{N_2(h/2) - N_2(h)}{3}$$

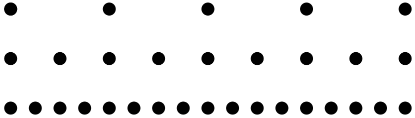
$$N_4(h) = N_3(h/2) + \frac{N_3(h/2) - N_3(h)}{7}$$

$$N_5(h) = N_4(h/2) + \frac{N_4(h/2) - N_4(h)}{2^4 - 1}$$

$$\boxed{N_{j+1}(h) = N_j(h/2) + \frac{N_j(h/2) - N_j(h)}{2^j - 1}}$$

Comment on the Richardson Extrapolation Technique

Note that we can only compute the Richardson extrapolation on the coarsest grid:



If we have the solutions for h, $h/2$, and $h/4$ we can extrapolate three times:

$$\begin{aligned}
\text{E1} &:= \text{combine } h, \text{ and } h/2, & \text{error} &\sim \mathcal{O}(h^4). \\
\text{E2} &:= \text{combine } h/2, \text{ and } h/4, & \text{error} &\sim \mathcal{O}(h^4). \\
\text{E3} &:= \text{combine E1, and E2}, & \text{error} &\sim \mathcal{O}(h^6).
\end{aligned}$$

However, the extrapolated solution is only available on the h-spaced grid.

Improving the Accuracy: Higher Order Finite Differences

If we use the 4th order accurate finite differences:

$$\begin{aligned} y'_n &\approx [-y_{n+2} + 8y_{n+1} - 8y_{n-1} + y_{n-2}]/12h \\ y''_n &\approx [-y_{n+2} + 16y_{n+1} - 30y_n + 16y_{n-1} - y_{n-2}]/12h^2 \end{aligned}$$

we can build a 4th order accurate scheme... but we run into some trouble.

Consider the point $n = 1$ (one grid-point from the left boundary point), and use the boundary condition $y_0 = y_a$:

$$\begin{aligned} y'_1 &\approx [-y_3 + 8y_2 - 8\mathbf{y_a} + \mathbf{y_{-1}}]/12h \\ y''_1 &\approx [-y_3 + 16y_2 - 30y_1 + 16\mathbf{y_a} - \mathbf{y_{-1}}]/12h^2 \end{aligned}$$

But, but, but... $\mathbf{y_{-1}}$ does not exist.

As we continue to solve ODEs, and especially PDEs we will see that dealing with boundary conditions is often the most challenging part of the problem.

In this case we can solve the problem by using a non-symmetric expression for the derivatives at $n = 1$ and $n = (N - 1)$... Generating those 4th order accurate stencils using either Taylor expansions, or Lagrange interpolating polynomials **is left as an exercise...**

Note that if we use non-symmetric stencils, the error expansion is going to contain all powers of h (h^4, h^5, h^6, h^7...)

We have a number of issues that require our attention:

[1] Other types of Boundary Conditions, including mixed (Robin-type: $\alpha u + \beta u' = \gamma$).

[2] Non-linear equations.

[3] Higher order equations.

[4] Solving the resulting linear system $A\tilde{\mathbf{y}} = \tilde{\mathbf{b}}$ in an efficient way. [Full details in Math 543]

Sometimes boundary conditions are stated in more complicated ways. Frequently it is stated as a linear combination of the function value, and its derivative, *i.e.*

$$c_1 y(a) + c_2 y'(a) = c_3$$

Note that this discussion covers the case $c_1 = 0$ (flux-only condition).

If we discretize the derivative using a forward difference we get

$$c_1 y(a) + c_2 \frac{y(a+h) - y(a)}{h} = c_3$$

or

$$\left[hc_1 - c_2\right] y_0 + c_2 y_1 = hc_3$$

If we solve

$$[hc_1 - c_2]\, y_0 + c_2 y_1 = hc_3$$

for y_0, we get

$$y_0 = \left[\frac{hc_3 - c_2 y_1}{hc_1 - c_2}\right].$$

If we use this value in the equation at node $n = 1$:

$$\begin{aligned}
\left[1 + \frac{h}{2}p_1\right] y_2 + \left[h^2 q_1 - 2\right] y_1 + \left[1 - \frac{h}{2}p_1\right] \mathbf{y_0} &= h^2 r_1 \\
\left[1 + \frac{h}{2}p_1\right] y_2 + \left[h^2 q_1 - 2\right] y_1 + \left[1 - \frac{h}{2}p_1\right] \left[\frac{\mathbf{hc_3 - c_2 y_1}}{\mathbf{hc_1 - c_2}}\right] &= h^2 r_1 \\
\left[1 + \frac{h}{2}p_1\right] y_2 + \left[h^2 q_1 - 2 - \left[\mathbf{1 - \frac{h}{2}p_1}\right]\left[\frac{\mathbf{c_2}}{\mathbf{hc_1 - c_2}}\right]\right] y_1 &= h^2 r_1 - \left[\mathbf{1 - \frac{h}{2}p_1}\right]\left[\frac{\mathbf{hc_3}}{\mathbf{hc_1 - c_2}}\right]
\end{aligned}$$

The rest of the linear system is unchanged.

Impact on Accuracy: The Curse of BCs, part II

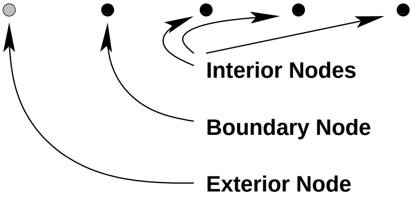
The forward difference we used

$$c_1 y(a) + c_2 \frac{y(a+h) - y(a)}{h} = c_3$$

is only **first-order accurate.**

Even though the rest of the equations in the system are based on second-order accurate approximations, the **overall order of accuracy is one.**

In order to overcome the lack of accuracy in the boundary condition, we **add** an external (fictitious) node to the grid (x_{-1}).



We can now express the boundary condition using the second-order accurate central difference:

$$c_1 y(a) + c_2 \frac{y(a+h) - y(a-h)}{2h} = c_3$$

or

$$c_1 y_0 + c_2 \frac{y_1 - y_{-1}}{2h} = c_3, \quad 2hc_1 y_0 + c_2 y_1 - c_2 y_{-1} = 2hc_3$$

We solve for y_{-1}:

$$y_{-1} = \left[\frac{2hc_1}{c_2}\right] y_0 + y_1 - \left[\frac{2hc_3}{c_2}\right]$$

If we use this value in the equation at node $\mathbf{n} = \mathbf{0}$:

$$\begin{aligned}
\left[1 + \frac{h}{2} p_0\right] y_1 + \left[h^2 q_0 - 2\right] y_0 + \left[1 - \frac{h}{2} p_0\right] \mathbf{y_{-1}} &= h^2 r_0 \\
\left[1 + \frac{h}{2} p_0\right] y_1 + \left[h^2 q_0 - 2\right] y_0 + \left[1 - \frac{h}{2} p_0\right] \left[\left[\frac{\mathbf{2hc_1}}{\mathbf{c_2}}\right] \mathbf{y_0} + \mathbf{y_1} - \left[\frac{\mathbf{2hc_3}}{\mathbf{c_2}}\right]\right] &= h^2 r_0 \\
\left[1 + \frac{h}{2} p_0 + \left[\mathbf{1} - \frac{\mathbf{h}}{\mathbf{2}} \mathbf{p_0}\right]\right] y_1 + \left[h^2 q_0 - 2 + \frac{\mathbf{2hc_1}}{\mathbf{c_2}} - \frac{\mathbf{h^2 p_0 c_1}}{\mathbf{c_2}}\right] y_0 &= \\
&= h^2 r_0 + \frac{\mathbf{2hc_3}}{\mathbf{c_2}} - \frac{\mathbf{h^2 c_3 p_0}}{\mathbf{c_2}}
\end{aligned}$$

This equation is **in addition to** the system on slides 11-12 — the additional unknown is y_0.

The changed system $\tilde{A}\tilde{\tilde{y}} = \tilde{\tilde{b}}$ looks like

The new top row corresponds to the new equation (at $n = 0$), the equation at $n = 1$ gains a sub-diagonal element $(s_1^- = 1 - \frac{h}{2}p_1)$, and the right-hand-side simplifies to $h^2 r_1$. The remainder of the new column is filled with zeros.

If we need higher degrees of accuracy, **or** higher order derivatives at the boundaries, we can use the same idea, but we have to add even more external / fictitious / “ghost” points.

Soon... Higher order equations, non-linear problems.