

Exercise 1. (Inverse matrix and Green's functions)

1. Write out the 4×4 matrix A from (2.43) for the boundary value problem $u''(x) = f(x)$ with $u(0) = u(1) = 1$ and for $h = 1/3$.
2. Write out the 4×4 inverse matrix A^{-1} explicitly for this problem.
3. If $f(x) = x$, determine the discrete approximation to the solution of the boundary value problem on this grid and sketch this solution and the four Green's functions from which the solution is obtained. (You may use Matlab or other tools for plotting, or you may plot by hand.)

Solution 1. (a) Using equation (2.43) as a guide, we can write out the matrix A as

$$A = 9 \begin{pmatrix} \frac{1}{9} & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & \frac{1}{9} \end{pmatrix}$$

(b) We'll use a grid of points

$$x_0 = 0, x_1 = \frac{1}{3}, x_2 = \frac{2}{3}, x_3 = 1.$$

As shown in class the matrix $A^{-1} = B$ has elements

$$B_{i+1,j+1} = hG(x_i, x_j) = \begin{cases} \frac{x_i}{3}(x_j - 1) & i < j \\ \frac{x_j}{3}(x_i - 1) & i \geq j \end{cases},$$

with the exception of the first and last columns given by $G_0(x) = 1 - x$ and $G_1(x) = x$ respectively. Using Julia for a bit of assistance, we see the full matrix is

$$A^{-1} = B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{2}{3} & -\frac{2}{27} & -\frac{1}{27} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{27} & -\frac{2}{27} & \frac{2}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

(c) When $f(x) = x$, our solution U is determined by

$$\begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \end{pmatrix} = A^{-1} \begin{pmatrix} 1 \\ \frac{1}{3} \\ \frac{2}{3} \\ 1 \end{pmatrix}$$

We implement this method of solving the BVP in julia and plot the Green's functions and the approximations below.

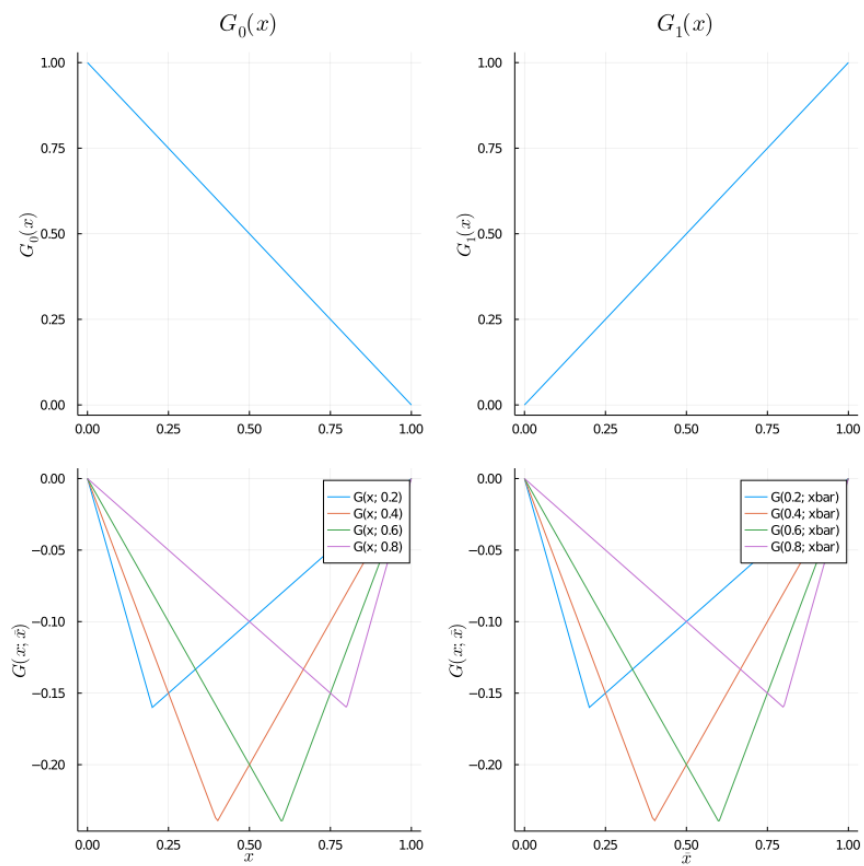


Figure 1: Plotting relevant Green's Functions

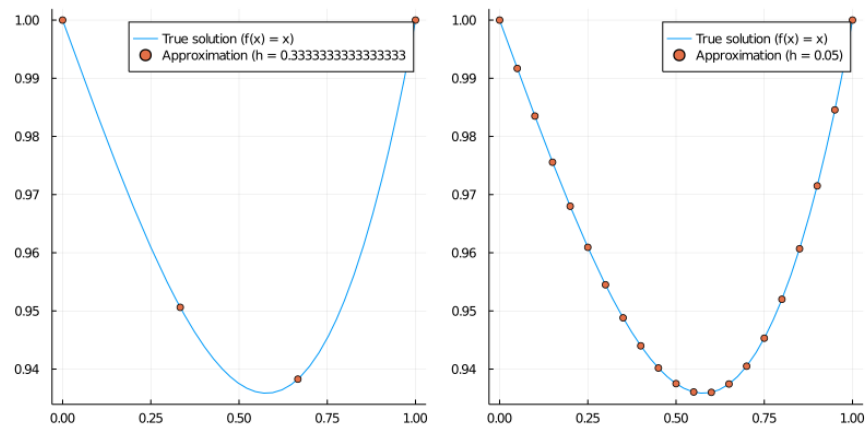


Figure 2: Plotting approximate solutions $\hat{u}(x)$ for $h = \frac{1}{4}$ and $h = \frac{1}{20}$.

Exercise 2. (Another way of analyzing the error using Green's functions) The *composite trapezoid rule* for integration approximates the integral from a to b of a function g by dividing the interval into segments of length h and approximating the integral over each segment by the integral of the linear function that matches g at the endpoints of the segment. (For $g > 0$, this is the area of the trapezoid with height $g(x_j)$ at the left endpoint x_j and height $g(x_{j+1})$ at the right endpoint x_{j+1} .) Letting $h = (b-a)/(m+1)$ and $x_j = a + jh$, $j = 0, 1, \dots, m, m+1$:

$$\int_a^b g(x) dx \approx h \sum_{j=0}^m \frac{g(x_j) + g(x_{j+1})}{2} = h \left[\frac{g(x_0)}{2} + \sum_{j=1}^m g(x_j) + \frac{g(x_{m+1})}{2} \right].$$

1. Assuming that g is sufficiently smooth, show that the error in the composite trapezoid rule approximation to the integral is $O(h^2)$. [Hint: Show that the error on each subinterval is $O(h^3)$.]
2. Recall that the true solution of the boundary value problem $u''(x) = f(x)$, $u(0) = u(1) = 0$ can be written as

$$u(x) = \int_0^1 f(\bar{x}) G(x; \bar{x}) d\bar{x}, \quad (1)$$

where $G(x; \bar{x})$ is the Green's function corresponding to \bar{x} . The finite difference approximation u_i to $u(x_i)$, using the centered finite difference scheme in (2.43), is

$$u_i = h \sum_{j=1}^m f(x_j) G(x_i; x_j), \quad i = 1, \dots, m. \quad (2)$$

Show that formula (2) is the trapezoid rule approximation to the integral in (1) when $x = x_i$, and conclude from this that the error in the finite difference approximation is $O(h^2)$ at each node x_i . [Recall: The Green's function $G(x; x_j)$ has a *discontinuous* derivative at $x = x_j$. Why does this not degrade the accuracy of the composite trapezoid rule?]

Solution 2. 1. Writing $g(x)$ in terms of its Taylor expansion

$$g(x) = g(x) + g'(x)h + O(h^2)$$

integrating this relationship over the interval $[x, x+h]$, we have

$$\int_x^{x+h} g(t) dt = hg(x) + \frac{h^2}{2} g'(x) + O(h^3).$$

Writing out the forward difference approximation for the first derivative which is $O(h)$, we see

$$\frac{h^2}{2} g'(x) = \frac{h}{2} (g(x+h) - g(x)) + O(h^3).$$

We then have that

$$\int_x^{x+h} g(t)dt = \frac{h}{2}(g(x) + g(x+h)) + O(h^3),$$

which shows that the error on a single subinterval of length h is $O(h^3)$. Summing this across all subintervals of which there are $(b-a)/h$, we have that the entire error is

$$\begin{aligned} \int_a^b g(x)dx &= \sum_{i=0}^{m+1} \int_{x_i}^{x_i+h} g(x)dx \\ &= \sum_{i=0}^{m+1} \left(\frac{h}{2}(g(x_i) + g(x_i+h)) + O(h^3) \right) \\ &= \frac{h}{2}g(x_0) + h \sum_{i=1}^m g(x_i) + \frac{h}{2}g(x_{m+1}) + O(h^2). \end{aligned}$$

Therefore, the error has order $O(h^2)$.

2. We will approximate the following integral with the composite trapezoid rule approximation

$$u(x_i) = \int_0^1 f(\bar{x})G(x_i; \bar{x}) d\bar{x}$$

Under the Trapezoid rule approximation, this becomes

$$u(x_i) \approx \frac{h}{2}f(x_0)G(x_i; x_0) + h \sum_{j=1}^m f(x_j)G(x_i; x_j) + \frac{h}{2}f(x_{m+1})G(x_i; x_{m+1}) \quad (3)$$

$$= h \sum_{j=1}^m f(x_j)G(x_i; x_j), \quad (4)$$

where we've used that $G(x_i; 0) = 0$ and $G(x_i; 1) = 0$. As this is simply the trapezoid approximation of the integral, we conclude that it has order $O(h^2)$. This holds despite the discontinuity in the derivative of $G(x; x_j)$ at $x = x_j$ due to the fact that the function is piecewise linear and its derivative is estimated using a forward difference i.e. at x_j the function is approximating the right hand derivative which does still exist and is of order $O(h)$. Though the derivative of G technically does not exist here, the error over the subinterval it contributes by using the forward difference is still of order $O(h^3)$ and contributed once and is inconsequential as $h \rightarrow 0$. Therefore, the total order of the algorithm is not degraded.

Exercise 3. (Green's function with Neumann boundary conditions)

1. Determine the Green's functions for the two-point boundary value problem $u''(x) = f(x)$ on $0 < x < 1$ with a Neumann boundary condition at $x = 0$ and a Dirichlet condition at $x = 1$, i.e, find the function $G(x, \bar{x})$ solving

$$u''(x) = \delta(x - \bar{x}), \quad u'(0) = 0, \quad u(1) = 0$$

and the functions $G_0(x)$ solving

$$u''(x) = 0, \quad u'(0) = 1, \quad u(1) = 0$$

and $G_1(x)$ solving

$$u''(x) = 0, \quad u'(0) = 0, \quad u(1) = 1.$$

2. Using this as guidance, find the general formulas for the elements of the inverse of the matrix in equation (2.54). Write out the 4×4 matrices A and A^{-1} for the case $h = 1/3$.

Solution 3. Starting with $G_0(x)$ and $G_1(x)$, the conditions given show that our solutions are

$$\begin{aligned} G_0(x) &= x - 1 \\ G_1(x) &= 1. \end{aligned}$$

Now to solve for $G(x, \bar{x})$, we know that

$$\begin{aligned} u'(1) - u'(0) &= \int_0^1 u''(x) dx \\ &= \int_0^1 \delta(x - \bar{x}) = 1. \end{aligned}$$

Since $u'(0) = 0$, this shows that $u'(1) = 1$. Therefore, we can write

$$G(x; \bar{x}) = \begin{cases} \bar{x} - 1 & \text{for } 0 \leq x < \bar{x} \\ x - 1 & \text{for } \bar{x} \leq x \leq 1 \end{cases}$$

which satisfies $G'(0; \bar{x}) = 0$, $G(1; \bar{x}) = 0$ and the second derivative condition we derived $G'(1, \bar{x}) = 1$ as well as the second derivative being 0 where it exists.

2. We can use these functions to determine the columns of the inverse matrix. In the case of the first and last columns of our inverse matrix A^{-1} , we write

$$\begin{aligned} (A^{-1})_{i1} &= G_0(x_i) \\ (A^{-1})_{i,m+1} &= G_1(x_i) \end{aligned}$$

with all other elements given by

$$(A^{-1})_{i+1,j+1} = hG(x_i, x_j).$$

Here, entries are shifted since we index the grid points starting with 0 and the matrix with 1. Writing out the matrix A explicitly, we see

$$A = 9 \begin{pmatrix} -\frac{1}{3} & \frac{1}{3} & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & & \frac{1}{9} \end{pmatrix}$$

Our Green's functions give the inverse matrix as

$$A^{-1} = \begin{pmatrix} -1 & -\frac{2}{9} & -\frac{1}{9} & 1 \\ -\frac{2}{3} & -\frac{2}{9} & -\frac{1}{9} & 1 \\ -\frac{1}{3} & -\frac{1}{9} & -\frac{1}{9} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Exercise 4. (Solvability condition for Neumann problem) Determine the null space of the matrix A^T , where A is given in equation (2.58), and verify that the condition (2.62) must hold for the linear system to have solutions.

Solution 4. We'll now find the nullspace of A^T which will satisfy

$$A^T U = \mathbf{0}.$$

This means that we have the following equations

$$\begin{aligned} -hU_0 + U_1 &= 0 \\ hU_0 - 2U_1 + U_2 &= 0 \\ U_1 - 2U_2 + U_3 &= 0 \\ &\vdots \\ U_{m-1} - 2U_m + hU_{m+1} &= 0 \\ U_m - hU_{m+1} &= 0. \end{aligned}$$

We then have

$$\begin{aligned} hU_0 &= U_1 \\ hU_0 - 2hU_0 + U_2 &= 0 \implies hU_0 = U_2 \\ hU_0 - 2hU_0 + U_3 &= 0 \implies hU_0 = U_3 \\ &\vdots \\ hU_0 - 2hU_0 + U_m &= 0 \implies hU_0 = U_m \\ U_0 &= U_{m+1} \end{aligned}$$

if U is in the null space of A^T . This means that the nullspace is spanned by the vector

$$U = (1, h, h, \dots, h, h, 1).$$

As we know, for A to have a solution, we must find vectors which are orthogonal to the nullspace of A^T . We then have that if a vector F of the form in 2.58 is orthogonal to the nullspace i.e. $U \cdot F = 0$, then

$$U \cdot F = \sigma_0 + \frac{h}{2}f(x_0) + \sum_{j=1}^m hf(x_j) + \frac{h}{2}f(x_{m+1}) - \sigma_1 = 0.$$

Moving σ_1 and σ_0 to the other side of the equation gives the desired condition.

Exercise 5. (Symmetric tridiagonal matrices)

1. Consider the **Second approach** described on p. 31 for dealing with a Neumann boundary condition. If we use this technique to approximate the solution to the boundary value problem $u''(x) = f(x)$, $0 \leq x \leq 1$, $u'(0) = \sigma$, $u(1) = \beta$, then the resulting linear system $A\mathbf{u} = \mathbf{f}$ has the following form:

$$\frac{1}{h^2} \begin{pmatrix} -h & h & & & \\ 1 & -2 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{m-1} \\ u_m \end{pmatrix} = \begin{pmatrix} \sigma + (h/2)f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{m-1}) \\ f(x_m) - \beta/h^2 \end{pmatrix}.$$

Show that the above matrix is similar to a symmetric tridiagonal matrix via a *diagonal* similarity transformation; that is, there is a diagonal matrix D such that DAD^{-1} is symmetric.

2. Consider the **Third approach** described on pp. 31-32 for dealing with a Neumann boundary condition. [Note: If you have an older edition of the text, there is a typo in the matrix (2.57) on p. 32. There should be a row above what is written there that has entries $\frac{3}{2}h$, $-2h$, and $\frac{1}{2}h$ in columns 1 through 3 and 0's elsewhere. I believe this was corrected in newer editions.] Show that if we use that first equation (given at the bottom of p. 31) to eliminate u_0 and we also eliminate u_{m+1} from the equations by setting it equal to β and modifying the right-hand side vector accordingly, then we obtain an m by m linear system $A\mathbf{u} = \mathbf{f}$, where A is similar to a symmetric tridiagonal matrix via a diagonal similarity transformation.

Solution 5. 1. Suppose we have a diagonal matrix D with diagonal $(\lambda_0, \dots, \lambda_{m+1})$. Then we have that

$$DAD^{-1} = \frac{1}{h^2} \begin{pmatrix} -h & h/\lambda_1 & & & \\ 1/\lambda_0 & -2 & 1/\lambda_2 & & \\ & 1/\lambda_1 & \ddots & \ddots & \\ & & \ddots & \ddots & 1/\lambda_{m+1} \\ & & & 1/\lambda_m & -2 \end{pmatrix}$$

From this, we see that selecting $\lambda_0 = 1/h$ and all other $\lambda_i = 1$ should produce a symmetric matrix.

2. Using equation at bottom of pg 31, we write

$$U_0 = \sigma h + \frac{4}{3}U_1 - \frac{1}{3}U_2,$$

which combined with the second derivative approximation for $U''(x_1) = f(x_2)$ allow us to remove the first equation and rewrite the second as

$$\frac{1}{h^2} \left(-\frac{2}{3}U_1 + \frac{2}{3}U_2 \right) = f(x_2) - \frac{\sigma}{h}$$

Also eliminating the U_{m+1} variable as described leaves us with the $m \times m$ matrix

$$A = \frac{1}{h^2} \begin{pmatrix} -\frac{2}{3} & \frac{2}{3} & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix}$$

Based on the knowledge of diagonal similarity transforms developed in part one, we see that using a diagonal matrix with diagonal $(\frac{3}{2}, 1, \dots, 1)$ gives the desired transform since the only ‘problem’ element is $A_{21} = 1$.

HW-2-Code-Figgins

January 22, 2021

```
[140]: using LinearAlgebra, Plots, LaTeXStrings
```

0.1 Exercise 1

```
[142]: G0(x) = 1 - x
      G1(x) = x

      function G(x, xbar)
        if x < xbar
          return x*(xbar - 1)
        elseif x >= xbar
          return xbar*(x - 1)
        end
      end
```

[142]: G (generic function with 2 methods)

```
[143]: function make_inverse(m)
      h = 1/(m-1)
      B = zeros(m,m)

      B[:,1] .= [G0(i*h) for i in 0:(m-1)]
      B[:, end] .= [G1(i*h) for i in 0:(m-1)]

      # Just Interior Rows
      for j in 1:(m-2)
        B[:,j+1] .= [h*G(i*h, j*h) for i in 0:(m-1)]
      end

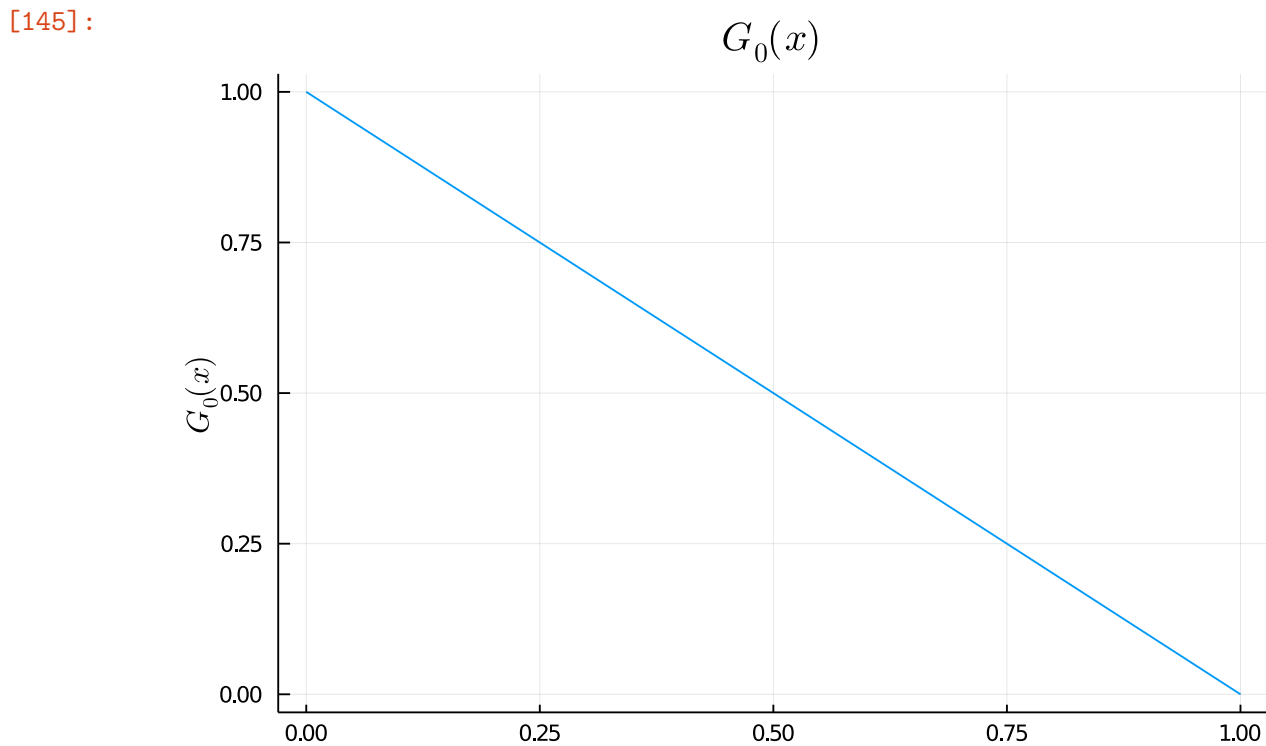
      return B
    end
```

[143]: make_inverse (generic function with 1 method)

```
[144]: B = make_inverse(4)
```

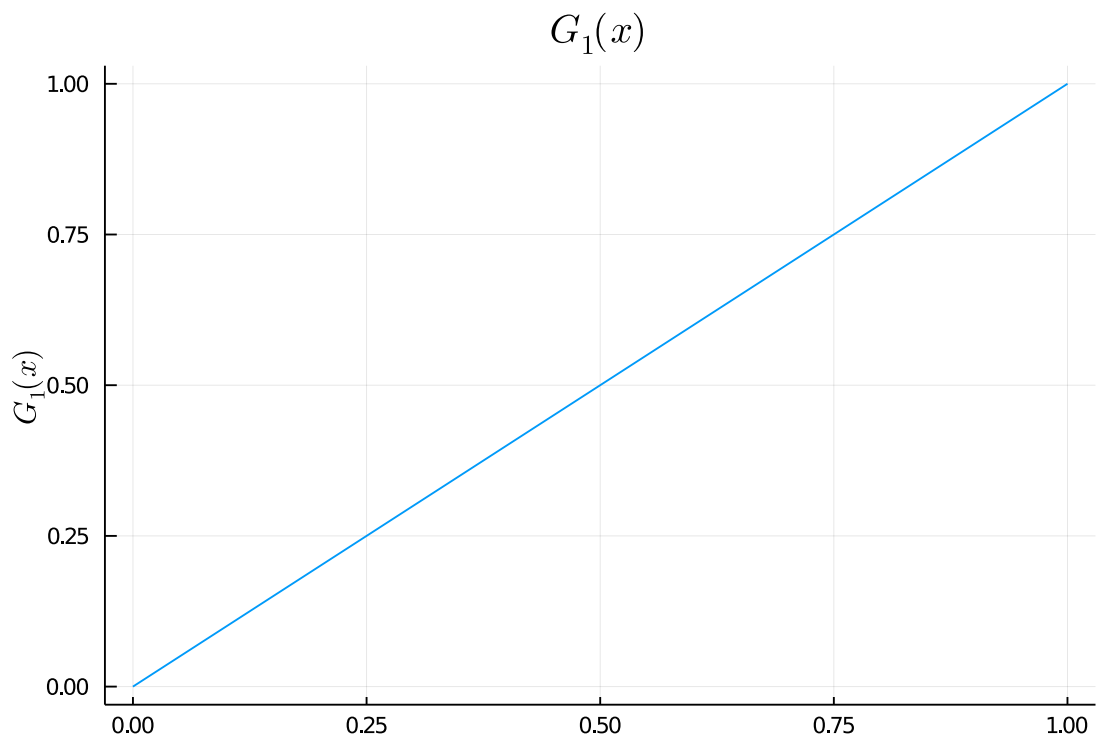
```
[144]: 4x4 Array{Float64,2}:
 1.0      -0.0      -0.0      0.0
 0.666667 -0.0740741 -0.037037  0.333333
 0.333333 -0.037037  -0.0740741 0.666667
 0.0       0.0       0.0       1.0
```

```
[145]: G0_plot = plot(G0, 0, 1,
    title = L"G_0(x)",
    ylab = L"G_0(x)",
    label = false)
```



```
[146]: G1_plot = plot(G1, 0, 1,
    title = L"G_1(x)",
    ylab = L"G_1(x)",
    label = false)
```

[146]:

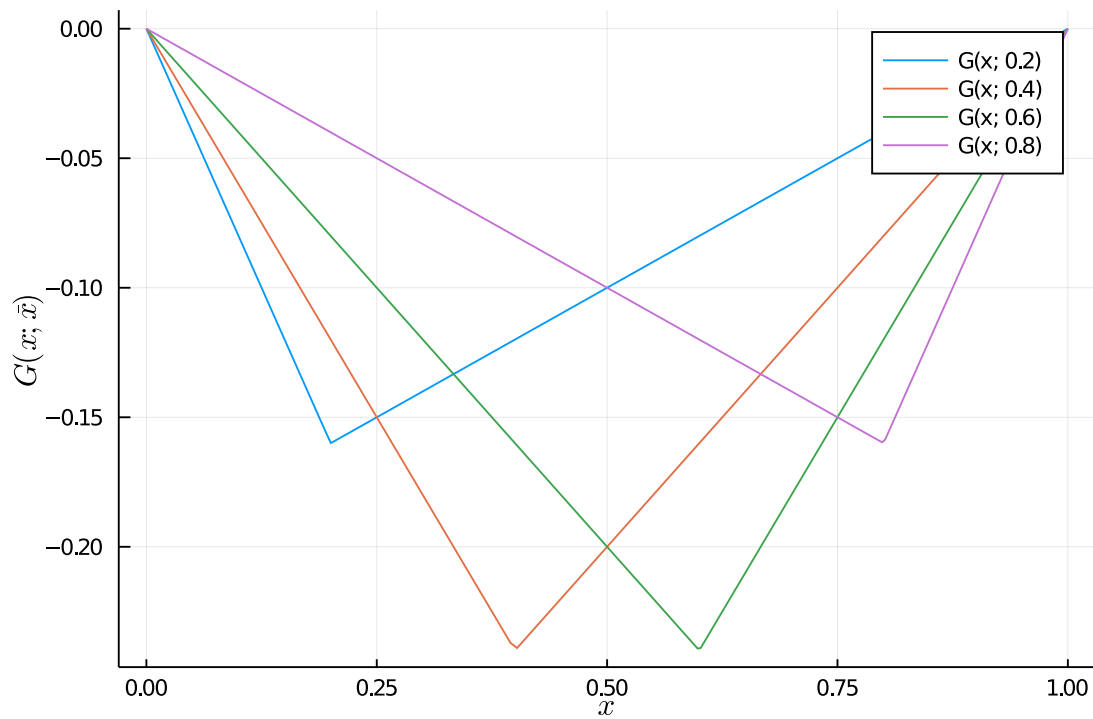


```
[156]: plot(x -> G(x, 0.2), 0, 1,
           label = "G(x; 0.2)",
           xlab = L"x",
           ylab = L"G(x; \bar{x})")

for xbar in 0.4:0.2:0.8
    plot!(x -> G(x, xbar), 0, 1,
           label = "G(x; $xbar)")
end

G_x_plot = plot!()
```

[156]:

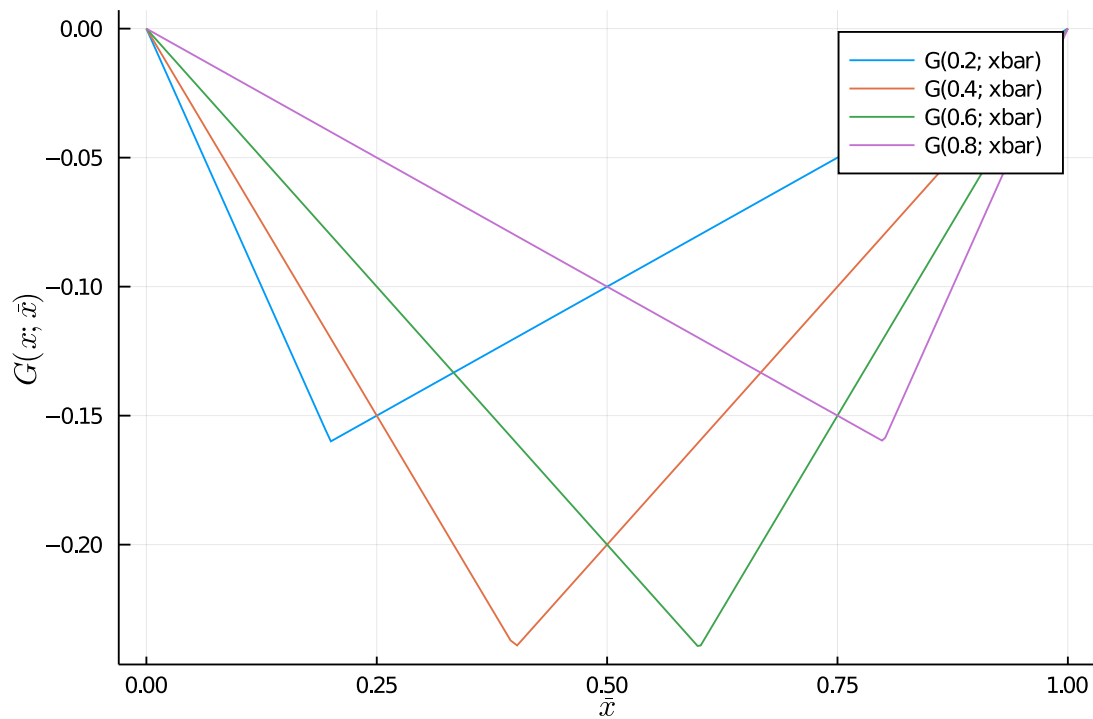


```
[157]: plot(xbar -> G(0.2, xbar), 0, 1,
           label = "G(0.2; xbar)",
           xlab = L"\bar{x}",
           ylab = L"G(x; \bar{x} )")

for x in 0.4:0.2:0.8
    plot!(xbar -> G(x, xbar), 0, 1,
           label = "G($x; xbar)")
end

G_xbar_plot = plot!()
```

[157]:



```
[177]: Greens_plot = plot(G0_plot, G1_plot, G_x_plot, G_xbar_plot, size = (800,800))
savefig("../hw/figs/hw-1-greens-plot.png")
```

```
[159]: function discretize_f(f, m)
    h = 1/(m-1)

    # Assume that endpoints are one
    F = ones(m)

    for i in 1:(m-2)
        F[i+1] = f(i*h)
    end
    return F
end

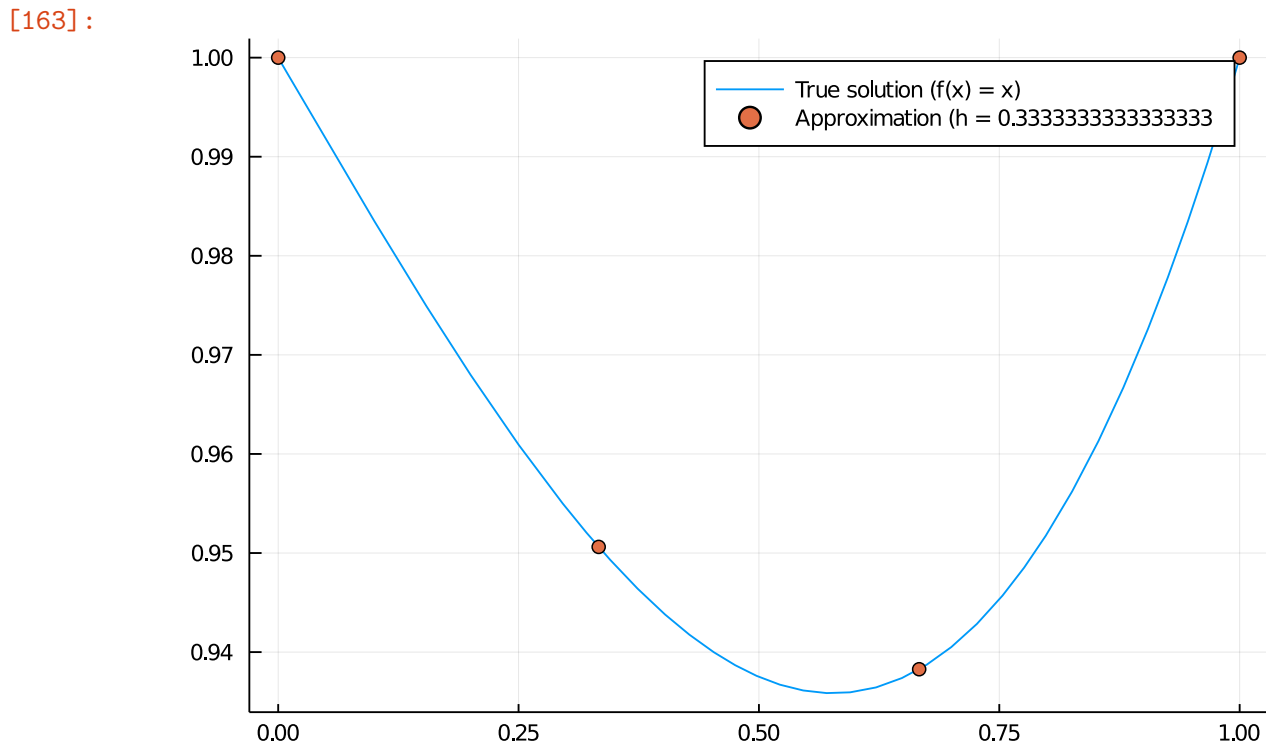
f(x) = x
discretize_f(f, 4)
```

```
[159]: 4-element Array{Float64,1}:
 1.0
 0.3333333333333333
 0.6666666666666666
 1.0
```

```
[163]: m = 4
B = make_inverse(m)
F = discretize_f(f, m)

plot(x -> x^3/6 - x/6 + 1, 0, 1,
     label = "True solution (f(x) = x)")

approx_plot_1 = scatter!([ i/(m-1) for i in 0:(m-1)], B*F,
                        label = "Approximation (h = $(1/(m-1)))")
```

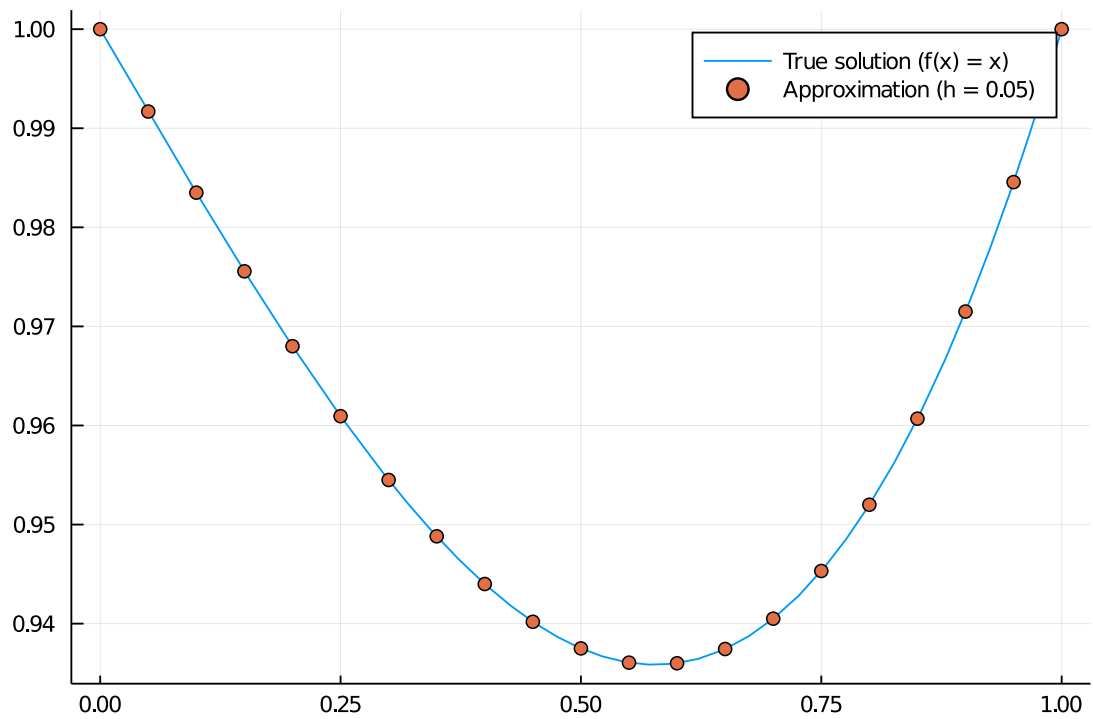


```
[164]: m = 21
B = make_inverse(m)
F = discretize_f(f, m)

plot(x -> x^3/6 - x/6 + 1, 0, 1,
     label = "True solution (f(x) = x)")

approx_plot_2 = scatter!([ i/(m-1) for i in 0:(m-1)], B*F,
                        label = "Approximation (h = $(1/(m-1)))")
```

[164]:



```
[178]: approx_plot = plot(approx_plot_1, approx_plot_2, layout = (1,2), size = (800,400))
savefig("../hw/figs/hw-1-approx-plot.png")
```