AMath 585
Take home final
Due Tuesday, March 17, 2020

Due to Canvas by 11:00pm PDT on the due date, at the latest.

No late papers accepted, so aim to get it in earlier!

To submit, see `https://canvas.uw.edu/courses/1352870/assignments/5301318`

100 points possible.

Open book and notes, and you can use other resources too, except please don't discuss it with other students or anyone else.

If you need clarification on some problem, or you think there's an error or typo, please post it on the Canvas discussion board so everyone has access to the same information.

Problem 1.

In Homework 2 you solved a linear beam equation, valid for small deformations. If the deformations are larger, then the equation must be nonlinear. In some cases an equation of this form can be used:

$$\begin{aligned} &au''''(x) - b(u'(x))^2(u''(x))^2 = f(x) \quad \text{for } 0 \leq x \leq 1 \\ &u(0) = \alpha_0, \quad u'(0) = \alpha_1, \quad u(1) = \beta_0, \quad u'(1) = \beta_1. \end{aligned}$$

where $a,\ b,\ \alpha_0,\ \alpha_1,\ \beta_0,\ \beta_1$ are all specified constants.

(a) If we discretize with a uniform grid using $h = 1/(m+1)$, suggest a nonlinear system of m equations that could be solved for $[U_1,\ U_2,\ \ldots,\ U_m]$ (the interior grid values) to obtain an approximate solution that is second order accurate.

Make sure the boundary conditions are also handled appropriately. Use the "First approach" described in hw2_solutions.html since this is less messy and should still be second order accurate.

(b) If we wanted to solve this system using Newton's method, we would need the Jacobian matrix for the nonlinear system developed in (a). You don't need to compute the full matrix, but do compute the diagonal elements J_{ii} for $i = 0,\ 1,\ 2,\ \ldots,\ m$.

Solution:

Part (a):
To discretize this equation, I must first find the approximations to the 4th, 2nd, and 1st derivatives, so in general, where U_0 just represents the center point:

$$U_0^{[4]} = \frac{1}{h^4}(U_{-2} - 4U_{-1} + 6U_0 - 4U_1 + U_2)$$

$$U_0'' = \frac{1}{h^2}(U_{-1} - 2U_0 + U_1)$$

$$U_0' = \frac{1}{2h}(U_{-1} + U_1)$$

Plugging into the equation we get:

$$\frac{a}{h^4}(U_{-2} - 4U_{-1} + 6U_0 - 4U_1 + U_2) - \frac{b}{4h^6}(U_{-1} + U_1)^2(U_{-1} - 2U_0 + U_1)^2 = f(x_0)$$

We are using a discretization where x_0 and x_{m+1} are the boundary points and x_1 through x_m are the interior points. Consider the boundaries and using one-sided approximations, we get:

$$U'(x_0) = \frac{-3}{2h}U(x_0) + \frac{2}{h}U(x_1) - \frac{1}{2h}U(x_2)$$

$$U'(0) = \frac{-3}{2h}\alpha_0 + \frac{2}{h}U(x_1) - \frac{1}{2h}U(x_2) = \alpha 1$$
$$\frac{2}{h}U(x_1) - \frac{1}{2h}U(x_2) = \alpha 1 + \frac{3}{2h}\alpha_0$$

$$U'(x_{m+1}) = \frac{3}{2h}U(x_{m+1}) - \frac{2}{h}U(x_m) + \frac{1}{2h}U(x_{m-1})$$
$$U'(1) = \frac{3}{2h}\beta_0 - \frac{2}{h}U(x_m) + \frac{1}{2h}U(x_{m-1}) = \beta 1$$
$$U'(1) = -\frac{2}{h}U(x_m) + \frac{1}{2h}U(x_{m-1}) = \beta 1 - \frac{3}{2h}\beta_0$$

The case for $U(x_2)$:

$$\frac{a}{h^4}(U(x_0)-4U(x_1)+6U(x_2)-4U(x_3)+U(x_4))-\frac{b}{4h^6}(U(x_1)+U(x_3))^2(U(x_1)-2U(x_2)+U(x_3))^2 = f(x_2)$$

Since we know $U(x_0) = U(0) = \alpha_0$, we can rewrite as:

$$\frac{a}{h^4}(-4U(x_1)+6U(x_2)-4U(x_3)+U(x_4))-\frac{b}{4h^6}(U(x_1)+U(x_3))^2(U(x_1)-2U(x_2)+U(x_3))^2 = f(x_2)-\frac{a}{h^4}\alpha_0$$

The case for U_{m-1}

$$\frac{a}{h^4}(U(x_{m-3}) - 4U(x_{m-2}) + 6U(x_{m-1}) - 4U(x_m) + U(x_{m+1}))$$
$$-\frac{b}{4h^6}(U(x_{m-2}) + U(x_m))^2(U(x_{m-2}) - 2U(x_{m-1}) + U(x_m))^2 = f(x_{m-1})$$

Since we know $U(x_{m+1}) = U(1) = \beta_0$, we can rewrite as:

$$\frac{a}{h^4}(U(x_{m-3}) - 4U(x_{m-2}) + 6U(x_{m-1}) - 4U(x_m))$$
$$-\frac{b}{4h^6}(U(x_{m-2}) + U(x_m))^2(U(x_{m-2}) - 2U(x_{m-1}) + U(x_m))^2 = f(x_{m-1}) - \frac{a}{h^4}\beta_0$$

In general, when we are at the k^{th} interior point, we have the equation:

$$\frac{a}{h^4}(U(x_{k-2}) - 4U(x_{k-1}) + 6U(x_k) - 4U(x_{k+1}) + U(x_{k+2}))$$
$$-\frac{b}{4h^6}(U(x_{k-1}) + U(x_{k+1}))^2(U(x_{k-1}) - 2U(x_k) + U(x_{k+1}))^2 = f(x_k)$$

Combining all these together, we have a system of m nonlinear equations: Note on notation: $U(x_1) = U_1$ and so on:

$$\frac{2}{h}U(x_1) - \frac{1}{2h}U(x_2) = \alpha 1 + \frac{3}{2h}\alpha_0$$
$$\frac{a}{h^4}(-4U(x_1) + 6U(x_2) - 4U(x_3) + U(x_4)) - \frac{b}{4h^6}(U(x_1) + U(x_3))^2(U(x_1) - 2U(x_2) + U(x_3))^2 = f(x_2) - \frac{a}{h^4}\alpha_0$$
$$\frac{a}{h^4}(U(x_{k-2})-4U(x_{k-1})+6U(x_k)-4U(x_{k+1})+U(x_{k+2}))-\frac{b}{4h^6}(U(x_{k-1})+U(x_{k+1}))^2(U(x_{k-1})-2U(x_k)+U(x_{k+1}))^2 = f(x_k)$$
$$\frac{a}{h^4}(U(x_{m-3})-4U(x_{m-2})+6U(x_{m-1})-4U(x_m))-\frac{b}{4h^6}(U(x_{m-2})+U(x_m))^2(U(x_{m-2})-2U(x_{m-1})+U(x_m))^2 = f(x_{m-1})-\frac{a}{h^4}\beta_0$$
$$-\frac{2}{h}U(x_m) + \frac{1}{2h}U(x_{m-1}) = \beta 1 - \frac{3}{2h}\beta_0$$

To confirm second order accuracy:

$$T_j = \frac{a}{h^4}(u(x_{j-2})-4u(x_{j-1})+6u(x_j)-4u(x_{j+1})+u(x_{j+2}))-b(\frac{u(x_{j-1}) + u(x_{j+1})}{2h})^2(\frac{u(x_{j-1}) - 2u(x_j) + u(x_{j+1})}{h^2})^2$$

From homework 1, we know that the chosen discretization is second order accurate: Additionally, the we know that the other two are second order accurate as well, so:

$$T_j = a(\frac{1}{6}h^2 u^{(6)}(x_j) + O(h^3)) - b(O(h^2))^2(O(h^2))^2$$

Thus since as m gets larger, h gets smaller, then this method is still $O(h^2)$ accurate.

Part (b): For our problem, the nonlinear system of equations $G_i(U)$ is:

$$G_i(U) = \frac{a}{h^4}(U_{i-2} - 4U_{i-1} + 6U_i - 4U_{i+1} + U_{i+2}) - \frac{b}{4h^6}(U_{i-1} + U_{i+1})^2(U_{i-1} - 2U_i + U_{i+1})^2 - f(x) = 0$$

Additionally,

$$G_0(U) = U_0 - \alpha_0 = 0$$
$$G_{m+1}(U) = U_{m+1} - \beta_0 = 0$$

The Jacobian matrix is:

$$J_{ij} = \frac{\partial G_i(U)}{\partial U_j}$$

We just want the J_{ii} elements. so:

$$J_{ii} = \frac{\partial G_i(U)}{\partial U_i} = \frac{6a}{h^4} + \frac{b}{h^6}(U_{i-1} + U_{i+1})^2(U_{i-1} - 2U_i + U_{i+1})$$
$$J_{00} = J_{mm} = 1$$

Problem 2.

Suppose A is a 2×2 singular matrix that is symmetric and has one postive eigenvalue, for example

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

is one such matrix. Then A is symmetric positive *semi-definite* ($u^T A u \geq 0$ for all nonzero u) but is not positive definite. If we define the functional

$$\phi(u) = \frac{1}{2}u^T A u - u^T f$$

as in Section 4.3 then levels sets of $\phi(u)$ are no longer ellipses.

(a) What is the geometry of these level sets in general?

(b) Let $z \in \mathbb{R}^2$ be a null vector of A and suppose $f \in \mathbb{R}^2$ is a vector satisfying $z^T f = 0$. Then the system $Au = f$ has infinitely many solutions. Show that the steepest descent method applied to $\phi(u)$ from any initial guess $u^{[0]}$ converges to *some* solution of the linear system in a single iteration. Hint: It might be easier to explain this using the result of (a) than by computing $u^{[1]}$ explicitly. That's fine as long as you give a convincing argument.

Solution: Part (a): The geometry of the level sets of $\phi(u)$ can be determined by the using the eigenstructure of A. In our case, since A is not SPD but only positive semi-definite, this means the eigenvalues are non-negative, or that at least one of the eigenvalues is 0.
Following the logic in the book for the ellipse case: Let us consider the case, as above, where m = 2 and let the points v_1 an v_2 have the property that the gradient $\nabla\phi(v_j)$ lies in the direction that connects v_j to the center u^* For some scalar α_j

$$Av_j - f = \alpha_j(v_j - u^*)$$

f = Au^* so:

$$A(v_j - u^*) = \alpha_j(v_j - u^*)$$

This means that α_j is an eigenvalue of A and $v_j - u^*$ is an eigenvalue. since m = 2, this means that:

$$A(v_1 - u^*) = \lambda_1(v_1 - u^*)$$

$$A(v_2 - u^*) = \lambda_2(v_2 - u^*)$$

Since we are assuming that one of the eigenvalues is zero, let $\lambda_2 = 0$. If we were assuming that the shape were ellipses, λ_2 would be the magnitude of one of the lengths of the semi-axis. Since this is now zero, the geometry of the level sets are now just lines.

Part(b): Using the geometry, if there are infinitely many solutions, then there are many lines. In 2D, these would just be parallel lines. Any initial guess, $u^{[0]}$ would be on one of these lines, and in the first iteration, following the level curve, you would converge right toward the solution.

Problem 3. Consider the *first order ODE*, with a single boundary condition,

$$\begin{aligned} &u'(x) = f(x), \quad 0 \leq x \leq 1, \\ &u(0) = \alpha. \end{aligned}$$

This boundary value problem has a unique solution $u(x) = \alpha + \int_0^x f(t)\,dt$. Using $u'(x_j) \approx (U_j - U_{j-1})/h$ on a uniform grid, we might attempt to approximate it by solving a system of the form

$$\frac{1}{h}\begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_m \end{bmatrix} = \begin{bmatrix} \alpha/h \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_m) \end{bmatrix}.$$

(a) Determine the exact solution to this linear system and show that U_j approximates $\alpha + \int_0^{x_j} f(t)\,dt$ with a "Riemann sum".

(b) Suppose we apply Gauss-Seidel, sweeping through the grid from left to right in the natural order (i.e. $j = 0,\ 1,\ \ldots,\ m$). Explain why one iteration is sufficient to converge to the exact solution of this linear system, for *any* choice of initial data $U^{[0]}$.

(c) Suppose we instead sweep from right to left in Gauss-Seidel (for $j = m,\ m-1,\ \ldots 0$). What is the iteration matrix G for this method on the system above? What are the eigenvalues of the G matrix? What are the matrices $G^2,\ G^3,\ \ldots$? (There is a simple pattern.)

(d) Suppose we start with initial guess $U^{[0]} = 0$ (the zero vector). Does this backward Gauss-Seidel method converge in a single step? In a finite number of steps? If so, how many?

(e) In Section 4.2.1 we saw that the spectral radius $\rho(G)$ tells us something about the rate of convergence of the method. Comment on what's going on in this example based on your answers to (c) and (d).

You are welcome to write a short code to try things out, but you are not required to implement this.

Solution:

Part (a) Determining exact solution to the linear system follows this pattern:

$$\begin{gathered} U_0 = \alpha \\ \frac{-U_0 + U_1}{h} = f(x_1) \\ \frac{-\alpha + U_1}{h} = f(x_1) \\ U_1 = hf(x_1) + \alpha \\ \frac{-U_1 + U_2}{h} = f(x_2) \\ \frac{-(hf(x_1) + \alpha) + U_2}{h} = f(x_2) \\ U_2 = h(f(x_1) + f(x_2)) + \alpha \end{gathered}$$

Following this pattern;

$$U_j = \alpha + \sum_{j=1}^{m} f(x_j)h$$

This approximates the solution to the integral as a "Riemann sum" where $h = \Delta x$

Part(b)

$$u^{[k+1]} = M^{-1}Nu^{[k]} + M^{-1}f$$

For forward Gauss- Seidel: A = M-N = D-L-U where

$$M = D - L$$
$$N = U$$

With this set up,

$$M = D - L = A$$
$$N = 0matrix$$

Thus:

$$M^{-1}N = 0matrix$$

The iteration becomes:

$$u^{[1]} = M^{-1}Nu^{[0]} + M^{-1}f$$
$$u^{[1]} = A^{-1}f$$

Thus, this is the exact solution to the linear system and the initial choice of $u^{[0]}$ does not impact it.

Part(c)

For this particularly: L is the sub-diagonal that is all ones and U is zeros and D is the diagonal which is all ones or the identity matrix. For backwards sweeping G-S,

$$M = \frac{1}{h}D - U = \frac{1}{h}I$$

$$N = \frac{1}{h}L$$

$$M^{-1} = hI$$

Thus:

$$G = M^{-1}N = IL = L$$

$$G = \begin{bmatrix} 0 & 0 & \dots & & & \\ 1 & 0 & \dots & & & \\ 0 & 1 & 0 & \dots & & \\ 0 & 0 & 1 & 0 & \dots & \\ & & & \ddots & & \\ & & & & 1 & 0 \end{bmatrix}$$

Since G is lower triangular, the eigenvalues are on the diagonal, which are all zeros.

$$G^2 = \begin{bmatrix} 0 & 0 & \dots & & & & \\ 0 & 0 & \dots & & & & \\ 1 & 0 & 0 & \dots & & & \\ 0 & 1 & 0 & 0 & \dots & & \\ & & \ddots & & & & \\ & & & 1 & 0 & 0 \end{bmatrix}, G^3 = \begin{bmatrix} 0 & 0 & \dots & & & & \\ 0 & 0 & \dots & & & & \\ 0 & 0 & 0 & \dots & & & \\ 1 & 0 & 0 & 0 & \dots & & \\ & \ddots & & & & & \\ & & 1 & 0 & 0 & 0 \end{bmatrix} \dots G^{m+1} = 0\ matrix$$

Part(d)

$$u^{[k+1]} = M^{-1}Nu^{[k]} + M^{-1}f$$
$$u^{[k+1]} = Lu^{[k]} + hf$$

If we start with the initial guess of the zero vector

$$u^{[1]} = hf$$

$$u^{[2]} = Lhf + hf = [L + I]hf$$
$$u^{[3]} = L[L + I]hf + hf = [L^2 + 2L]hf$$

Since G = L when taken to larger powers goes to zero, this will not converge.

Part(e) The rate of convergence is related to the spectral radius of G, which in this case, is 0 for both G matrices in parts a and b. However, since A is not symmetric positive definite, and G is non-normal, it does not tell us about convergence. In the case for part A, since we were doing forward G-S, it converged in one iteration since A = M and the G matrix was actually zero.

Problem 4.

Consider the Conjugate gradient algorithm on page 87 for some symmetric positive definite matrix $A \in \mathbb{R}^{m\times m}$. Suppose we happen to choose the initial guess u_0 in such a way that the initial residual $r_0 = f - Au_0$ is an eigenvector of A. Show that in this case the method converges to the true solution of the linear system in one iteration.

Solution:

Following the conjugate gradient algorithm: Let v be an eigenvector of A. This implies that $Av = \lambda v$

$$r_0 = f - Au_0$$

$$r_0 = v$$
$$p_0 = v$$
$$\omega_0 = Ap_0 = Av = \lambda v$$
$$\alpha_0 = \frac{r_0^T r_0}{p_0^T \omega_0} = \frac{v^T v}{vT\lambda v} = \frac{1}{\lambda}$$
$$u_1 = u_0 + \alpha_0 p_0 = u_0 + \frac{1}{\lambda}v$$

Now, if u_1 were the true solution, then $Au_1 - f = 0$

$$Au_1 - f$$
$$A(u_0 + \frac{1}{\lambda}v) - f$$
$$Au_0 + \frac{1}{\lambda}Av - f$$
$$Au_0 + \frac{1}{\lambda}\lambda v - f$$
$$Au_0 + v - f$$

Since $Au_0 = f - v$

$$f - v + v - f = 0$$

Thus:

$$Au_1 = f$$

and the conjugate gradient converges in one iteration.

Problem 5.

Consider the BVP

$$u''(x) = f(x), \qquad \text{for } 0 \le x \le 1$$

with boundary conditions

$$\gamma_0 u(0) + \gamma_1 u'(0) = \sigma, \qquad u(1) = \beta.$$

At $x = 1$ a standard Dirichlet BC is specified, but at $x = 0$ we now have a “mixed” or “Robin” boundary condition, assuming $\gamma_0,\ \gamma_1,\ \sigma$ are all specified constants, as is β. For some physical problems this is the correct type of boundary condition, e.g. in a heat conduction problem it corresponds to a case in which the heat flux at $x = 0$ is related to the temperature at this point.

(a) Set up a tridiagonal linear system $Au = f$ that could be solved to model this, with the following properties:

- $u = [u_0,\ u_1,\ \ldots,\ u_m]$ contains the unknown boundary value u_0 but not the known value $u_{m+1} = \beta$ (assuming as usual that $x_j = jh$ for $j = 0,\ 1,\ \ldots,\ m+1$ on a grid with $h = 1/(m+1)$).
- The method is second order accurate.

Follow the strategy of the second approach on page 31 to obtain the first equation in your linear system (i.e. introduce a ghost point u_{-1} and then eliminate it from the two equations that involve this unknown). Write out the matrix and right hand side of your system.

(b) Determine the local truncation error of your method, $\tau = [\tau_0,\ \tau_1,\ \ldots,\ \tau_m]$. We expect $\tau_j = C_j h^2 + o(h^2)$ so determine the constants C_j in terms of derivative(s) of the exact solution $u(x)$ (by doing Taylor series expansions, assuming $u(x)$ is sufficiently smooth).

Solution:

Part (a): Using the standard centered approximation for the second derivative:

$$\frac{1}{h^2}(U_{-1} - 2U_0 + U_1) = f(x_0)$$

At the left boundary, introducing a ghost point (U_{-1}) we get:

$$\gamma_0 U_0 + \frac{\gamma_1}{2h}(U_1 - U_{-1}) = \sigma$$

Using the method outlined on page 31, we can use these two equations to eliminate the ghost point: From the first equation:

$$\frac{1}{h^2}U_{-1} = f(x_0) + \frac{2}{h^2}U_0 - \frac{1}{h^2}U_1$$
$$U_{-1} = h^2 f(x_0) + 2U_0 - U_1$$
$$\gamma_0 U_0 + \frac{\gamma_1}{2h}U_1 - \frac{\gamma_1}{2h}U_{-1} = \sigma$$
$$\gamma_0 U_0 + \frac{\gamma_1}{2h}U_1 - \frac{\gamma_1}{2h}(h^2 f(x_0) + 2U_0 - U_1) = \sigma$$

$$(\gamma_0 - \frac{\gamma_1}{h})U_0 + \frac{\gamma_1}{h}U_1 = \sigma + \frac{\gamma_1 h}{2}f(x_0)$$

Since we know the boundary value at the right boundary, that does not need to be included, however, discretizing at the point x_m yields:

$$\frac{1}{h^2}(U_{m-1} - 2U_m + U_{m+1} = f(x_m)$$

$U_{m+1} = U(1) = \beta$

$$\frac{1}{h^2}U_{m-1} - \frac{2}{h^2}U_m = f(x_m) - \frac{\beta}{h^2}$$

Now, set up the tridiagonal system:

$$\begin{bmatrix} \gamma_0 - \frac{\gamma_1}{h} & \frac{\gamma_1}{h} & 0 & \cdots & & \\ \frac{1}{h^2} & \frac{-2}{h^2} & \frac{1}{h^2} & 0 & \cdots & \\ 0 & \frac{1}{h^2} & \frac{-2}{h^2} & \frac{1}{h^2} & 0 & \cdots \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{h^2} & \frac{-2}{h^2} & \frac{1}{h^2} \\ & & & & \frac{1}{h^2} & \frac{-2}{h^2} \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ U-2 \\ \vdots \\ U_{m-1} \\ U_m \end{bmatrix} = \begin{bmatrix} \sigma + \frac{\gamma_1 h}{2} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(m-1) \\ f(x_m) - \frac{\beta}{h^2} \end{bmatrix}$$

Part (b): Determining the LTE:

$$T_j = \frac{1}{h^2}[u(x_{j-1} - 2u(x_j) + u(x_{j+1}) - f(x_j)$$

Using Taylor Expansions:

$$
\begin{aligned}
u(x_{j-1} &= u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(x_j) - \frac{h^3}{6}u''(x_j) + \frac{h^4}{24}u''''(x_j) \\
-2u(x_j) &= -2u(x_j) \\
u(x_{j-1} &= u(x_j) + hu'(x_j) + \frac{h^2}{2}u''(x_j) + \frac{h^3}{6}u''(x_j) + \frac{h^4}{24}u''''(x_j) \\
T_j &= \frac{1}{h^2}[h^2u''(x_j) + \frac{h^4}{12}u''''(x_j) + O(h^4)] - f(x_j) \\
T_j &\sim \frac{h^2}{12}u''''(x_j) + O(h^3) \\
T_j &\sim \frac{h^2}{12}u''''(x_j) + o(h^2)
\end{aligned}
$$

Problem 6. (With corrected boundary conditions)

(a) Implement the method you derived in the previous problem (in Python or Matlab). It is ok to base this on code you have previously written for homework problems, and/or the class Jupyter notebooks.

(b) Test it on the problem

$$
\begin{aligned}
&u''(x) = 4, \qquad 0 \le x \le 1, \\
&2u(0) + 3u'(0) = 1, \qquad u(1) = 2,
\end{aligned}
$$

which has exact solution $u(x) = 2x^2 + x - 1$.

Explain why you expect the exact solution to your linear system to agree with exact solution of the ODE when evaluated at the grid points, and confirm that this is true for your implementation.

(c) Also test it on the problem

$$
\begin{aligned}
&u''(x) = -18x + 4, \qquad 0 \le x \le 1, \\
&2u(0) + 3u'(0) = 1, \qquad u(1) = -1,
\end{aligned}
$$

which has exact solution $u(x) = -3x^3 + 2x^2 + x - 1$. In this case check that your method is second order accurate by producing a log-log plot of the errors (similar to what was done in the notebook BVP1), using $m + 1 = 50,\ 100,\ 200,\ 400,\ 800$.