Problem Set 5

18.330 Intro to Numerical Analysis (MIT, Spring 2019)

Due: March 21. To be submitted *online* on Stellar.

Problem 1. Getting Richardson extrapolation wrong. (20 points)

As we saw in class, for any numerical method $F(\Delta)$ that depends on some stepsize Δ, and for which we know that the exact solution $F(0)$ is approached as $F(\Delta) = F(0) + C\,\Delta^p + \mathcal{O}(\Delta^{p+1})$ we can construct the *Richardson extrapolation*

$$F^{\text{Richardson}}(\Delta) = \frac{t^p F(\Delta/t) - F(\Delta)}{t^p - 1},$$

for some t. This extrapolation achieves an improved convergence order at the cost of more function evaluations. Crucially, this depends on knowing the convergence order p of $F(\Delta)$. But what if we get p wrong? Say we mistakenly believe that the convergence order of $F(\Delta)$ is $q \neq p$, will the order of the Richardson extrapolation be better, worse, or equal to that of F? Show your work!

Note: We are only interested in the power p, not the constant prefactor C of the methods.

Problem 2. Romberg integration. (20 points)

The error incurred by using the rectangular, trapezoid, and Simpson's rule decays as N^{-1}, N^{-2}, and N^{-4}, respectively.[1] Using this information, construct the Richardson-extrapolated versions of the three integration functions you wrote for PSET 1. Apply your functions to compute the integral

$$\int_0^{10} \frac{\sin(x^2)}{\sqrt{x^2+1}}\,dx.$$

For all three methods, plot the relative error against the number of function evaluations for both the original and the Richardson extrapolated versions. Note that the number of function evaluations is not necessarily equal to the number of subdivisions of the integration interval.

For reference, the first 20 digits of the exact value of this integral are `0.47519858913634741151`.

Problem 3. Steffensen's method. (20 points)

Recall Newton's method for computing the root $f(x^*) = 0$ of some differentiable function f,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

The fact that we need to know the derivative $f'(x)$ can be inconvenient. *Steffensen's method* is a variant of Newton's method with the iteration prescription

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}$$
$$g(x_n) = \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}.$$

[1] As you have noted correctly, our hand-waving analysis from class actually gives the *wrong* convergence order for Simpson's rule. We will correct this later.

Like the secant and the bisection method, this method requires only evaluations of $f(x)$, and not of the derivative. Here you will investigate whether it also shares a slow convergence rate with these other derivative-free methods.

(a) Interpret $g(x_n)$ as an approximation for $f'(x_n)$. What is the finite difference stencil used here? What is the stepsize?

(b) Mimic the convergence analysis from class to estimate the convergence rate of Steffensen's method. Specifically, if $\varepsilon_n = |x_n - x^*|$ with x^* the exact root, express ε_{n+1} as a function of ε_n, iterate, and then write ε_n as a function of the initial error ε_0. Compare with the convergence rate of Newton's method.

Problem 4. Newton's method in higher dimensions. (20 points)

Write a program that implements Newton's method in n dimensions to find solutions to the following systems of equations. You only need to find one solution for each system. Iterate until the L_2 norm of the solution vector $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ has converged to a relative tolerance of 10^{-8}.

(a)

$$\begin{aligned} 3x_1^2 - x_2^2 &= 0 \\ 3x_1 x_2^2 - x_1^3 - 1 &= 0. \end{aligned}$$

(b)

$$\begin{aligned} 6x_1 - 2\cos(x_2\, x_3) - 1 &= 0 \\ 9x_2 + \sqrt{x_1^2 + \sin x_3 + 1.06} + 0.9 &= 0 \\ 60x_3 + 3e^{-x_1 x_2} + 10\pi - 3 &= 0. \end{aligned}$$

Problem 5. The method of steepest descent. (20 points)

Intuitively, given a function $f(\mathbf{x})$, the gradient $\nabla f = (\partial f/\partial x_1, \partial f/\partial x_2, \dots, \partial f/\partial x_n)$ points in the direction of the greatest *increase* of f, so, starting from some $\mathbf{x}_k$ a reasonable prescription for a minimization algorithm would be to go in the direction of greatest *decrease*:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k).$$

This is the *method of steepest descent*. But how do we choose the stepsize α_k? Here you will solve this problem for quadratic functions

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q \mathbf{x} - \mathbf{b}^\top \mathbf{x},$$

where Q is a symmetric constant matrix, $\mathbf{b}$ is a constant vector, and $^\top$ denotes the transpose.

(a) Find the optimal stepsize α_k by exactly minimizing the *scalar* problem

$$\min_{\alpha_k} f(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)).$$

Note: This problem is called a *line search*.

(b) Consider the quadratic function given by

$$Q = \begin{pmatrix} 4 & -2 \\ -2 & 3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Starting from $\mathbf{x}_0 = (-4, -3)^\top$, iterate the method of steepest descent with (i) the optimal step size you found and (ii) $\alpha_k = 0.1$ until the relative change $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < 10^{-8}\,\|\mathbf{x}_{k+1}\|$. What is the value $f(\mathbf{x}^*)$ at the minimum?

Plot the iterates on top of a contour plot of $f(\mathbf{x})$ on the square $-6 < x < 6$, $-6 < y < 6$.

Then plot the relative errors $\|\mathbf{x}_k - \mathbf{x}^*\|/\|\mathbf{x}^*\|$ against the number of iterations. What is the rate of convergence?

(c) Solve the same problem as in part (b) by applying Newton's method to the equation $\nabla f(\mathbf{x}) = 0$. Try various initial conditions. What convergence behavior do you see? Can you explain this?