Poisson project for InFoMM C++ training

Consider the 1D Poisson equation

$$\frac{d^2y}{dx^2} + f(x) = 0, \qquad 0 \leq x \leq 2\pi,$$

where $f(x) = 2\cos(x)\exp(-x)$, and where we are given the Dirichlet boundary conditions $y(0) = 0$, $y(2\pi) = 0$. The closed form solution to this BVP is $y(x) = sin(x)\exp(-x)$. Discretization of the partial second derivative operator using central differences and step length $h = 2\pi/(n-1)$ leads to the system of equations $Ay = b$ where

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & & & \vdots \\ 0 & -1 & 2 & -1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 0 \\ h^2 f(h) \\ h^2 f(2h) \\ \vdots \\ h^2 f((n-2)h) \\ 0 \end{bmatrix}.$$

Note: the first and last rows are different due to the boundary conditions.

1. Create a size n vector x containing evenly spaced values from 0 to 2π. Use any C++ linear algebra library you wish, but you might want to consider the Eigen library. Create a sparse matrix A and the corresponding source vector b and solve the equation $Ay = b$ to find y. Print out the error between your solution and the closed form solution and see how this varies with n.
2. Create STL vectors (`std::vector<>`) representing the diagonals of A and b. Use the Thomas Algorithm to obtain a solution to $Ay = b$. Use as many STL algorithms you can to implement your algorithm.
3. Using the Boost timer library, compare the speed of 1 and 2 and a Matlab solution for varying n.