Two methods of solving Poisson's equation

Anna Lina P. Sjur and Jan-Adrian H. Kallmyr

September 6, 2018

Abstract

State problem. Briefly describe method and data. Summarize main results.

1 Introduction

Physics is a field concerned with the behaviour of nature, and nature is everchanging. It is therefore no surprise that differential equations appear everywhere in physics. From global climate dynamics to statistical mechanics, what we find is that differential equations, often many and coupled, are required to explain or model the phenomena. For such large models, efficiency is important, as we would, for example, like to have timely weather forecasts. One way to make a model more efficient is by using an efficient algorithm for solving differential equations.

In this report, we compare two different numerical methods of solving linear second-order differential equations with the Dirichlet boundary conditions. To do this, we will solve the one-dimensional Poisson's equation:

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}r^2} = -4\pi\rho(r). \tag{1}$$

In the methods section, we develop an approximation for the 2nd derivative to the 2nd order. We will then solve eq. 1 numerically, using gaussian elimination and lower-upper decomposition. As for the former, we will further specialise it to solve eq. 1 more efficiently. Next, in the results section we present the comparison between our numerical solutions and the analytical solution, as well as the error. We then compare the efficiency of all three algorithms, and finally, in the discussion section we will consider the advantages and disadvantages of each algorithm, and discuss their uses.

2 Methods

We would like to solve eq. 1 numerically. Generalising the equation, we get

$$-\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = f(x), \tag{2}$$

where we have assumed that $\rho \propto \frac{1}{r}e^{-r}$ and let $r \to x$, $\phi \to u$. Summing the backward and forward Taylor expansions of $u(x)$ and discretising the equation for n integration points, we get:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \mathcal{O}\big(h^2\big), \tag{3}$$

where $h = \frac{1}{n+1}$. Using the Dirichlet boundary conditions, we

3 Results

4 Discussion

References

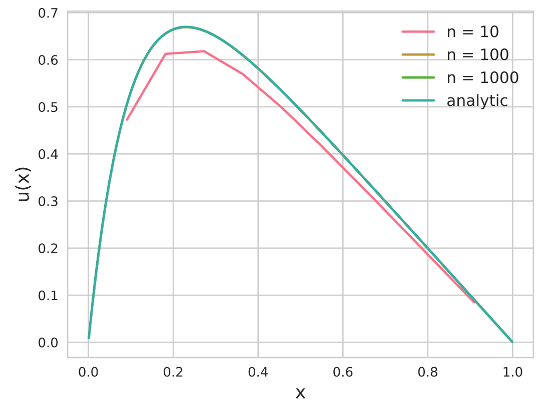

Figure 1: The numeric solution using different numbers of steps with the analytic solution. n=100 and n=1000 gives almost the same results as the analytic solution.

$\mathbf{n}$	$\mathbf{t_g/t_s}$	$\mathbf{t_{LU}/t_s}$
10	2.08	3.70
10^2	1.89	$1.00 \cdot 10^2$
10^3	1.48	$1.05 \cdot 10^4$
10^4	1.43	$1.18 \cdot 10^6$
10^5	1.39	-
10^6	1.41	-
10^7	1.39	-

Table 1: Ratio between CPU time for the general algorithm ($\mathbf{t_g}$), the special algorithm ($\mathbf{t_g}$) and the LU decomposition algorithm ($\mathbf{t_{LU}}$) for different matrix sizes ($\mathbf{n}$). The LU decomposition chrased for $\mathbf{n}$ greater than 10^4.

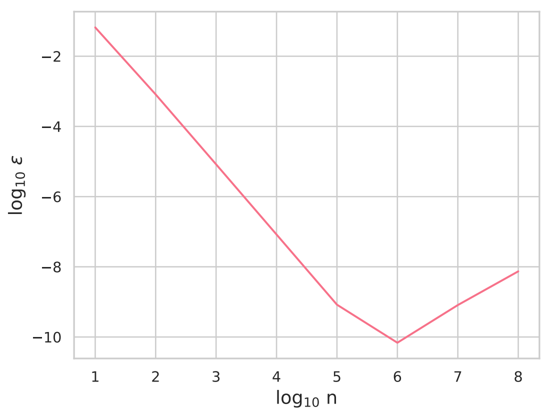

Figure 2: The maximum error in our specialised matrix solver as a function of the number of steps/matrix size in a logarithmic scale.