Computational Physics: Poisson's Equation with B-Spline Collocation

Stockholm University, Spring Term 2024
Max Maschke

Apr 25 2024

Contents

1 Introduction

The aim of this project is to implement the collocation method using B-splines and to apply it in solving Poisson's equation for spherically symmetric electrostatic problems. It is part of the coursework for the computational physics class held at Stockholm University in 2024.

2 Poisson's Equation in Electrostatics

Maxwell's equations

$$\partial_\nu F^{\mu\nu} = -\mu_0 J^\mu, \quad \partial_\gamma F_{\mu\nu} + \partial_\mu F_{\nu\gamma} + \partial_\nu F_{\gamma\mu} = 0, \tag{1}$$

the solution to which is given by the electromagnetic field tensor $F^{\mu\nu}$ can be elegantly treated by introducing the four-potential $A^\mu = (V/c, \mathbf{A})$, defined by

$$F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu. \tag{2}$$

Here, V is the scalar potential and $\mathbf{A}$ is the vector potential. The four-potential in Lorentzian gauge obeys

$$\Box A^\mu = \mu_0 J^\mu \tag{3}$$

for a given distribution of sources $J^\mu = (\rho, \mathbf{j})$. The solutions are retarded or advanced potentials. For our purposes, we're interested in electrostatic sources, so the treatment simplifies greatly. The vector potential vanishes and the relevant equation for V is Poisson's equation

$$\partial_{\mathbf{x}}^2 V(\mathbf{x}) = -\frac{\rho(\mathbf{x})}{\varepsilon_0}, \tag{4}$$

solved by

$$V(\mathbf{x}) = \frac{1}{4\pi\varepsilon_0} \int \mathrm{d}\mathbf{x}' \, \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|}. \tag{5}$$

For more complicated charge densities, the integral may not have a simple exact representation. In these cases, we may be interested in solving the differential equation numerically instead.

If the charge density is spherically symmetric, i.e. $V(\mathbf{x}) = V(r)$, the potential will share this symmetry. Introducing $\varphi(r) = r \cdot V(r)$, (4) becomes

$$\partial_r^2 \varphi(r) = -r \frac{4\pi\rho(r)}{4\pi\varepsilon_0} \tag{6}$$

where we did not cancel the factor of 4π because we will let $4\pi\varepsilon_0 = 1$ later.

3 B-Splines and the Collocation Method

B-splines are a set of $N-k$ piecewise polynomial functions defined on an interval $[a, b]$ divided into N so-called knot points t_i, $i = 0, \ldots, N-1$ that must be ascending (but not necessarily strictly ascending). A B-spline of order k is a polynomial of degree $k+1$ on an interval $[t_i, t_{i+k})$

and 0 elsewhere. The definition is given by the recursion

$$B_{i,k=1}(x) = \begin{cases} 1, & x \in [t_i, t_{i+1}) \text{ and } i < N \\ 1, & x \in [t_i, t_{i+1}] \text{ and } i = N \\ 0, & \text{else} \end{cases} \tag{7}$$

$$B_{i,k} = \frac{x - t_i}{t_{i+k+1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x). \tag{8}$$

Here, if the denominator is ever zero, the corresponding spline is ensured to also be zero, and the formula is to be understood in the limit where the respective fraction goes to zero. This must be explicitly included in any implementation to prevent dividing by zero.

What's useful about the splines is that they can provide a local basis to express an approximation to a function. Depending on the order of the splines, they are guaranteed to satisfy certain differentiability conditions, e.g. for $k = 4$, the first two derivatives are continuous. What's even nicer is that these derivatives can themselves be expressed as a superposition of lower-order splines. We omit the formulae here for brevity as they can be found on the assignment sheet.

If we wish to find the solution $f(x)$ to a boundary value problem of the type

$$\partial_x^n f(x) + \sum_{i=0}^{n-1} q_i(x) \partial_x^i f(x) = g(x), \quad x \in [a, b] \tag{9}$$

with appropriate boundary conditions $f(a) = \alpha$, $f(b) = \beta$, we can find an approximate solution by making a B-spline superposition Ansatz for f. We must require $k \geq n + 2$ and choose at least $N - 2(k-1)$ discretisation knot points x_i, which must be appended by $k-1$ "ghost-points" at each boundary which we need to ensure the local basis properties of the splines. Then we suppose

$$f(x_i) = \sum_{n=i-k+1}^{i-1} c_n B_{n,k}(x_i), \quad i = k, \ldots, N - k \tag{10}$$

and insert this into (9). This yields $N - 2(k-1)$ linear equations for the $N - k$ unknowns c_n. The remaining equations are yielded by enforcing the boundary conditions. The c_n can be directly obtained using Gaussian elimination. This method is referred to as collocation.

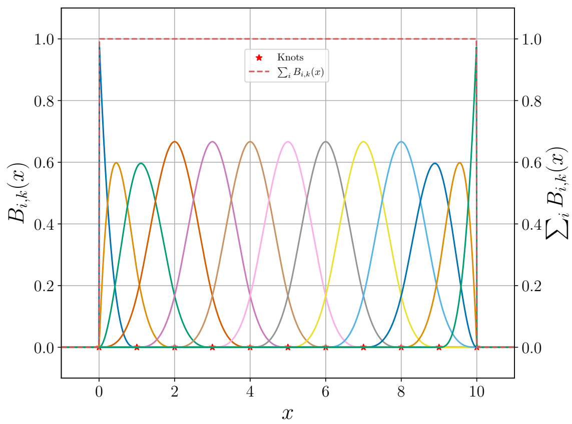
The obtained function has an exact representation everywhere and can be exactly integrated or differentiated. Another major advantage is that the knot points do not have to be linearly spaced, so any distribution that is appropriate for a given problem may be used to balance accuracy and computational resource usage.

4 Implementation and Numerics

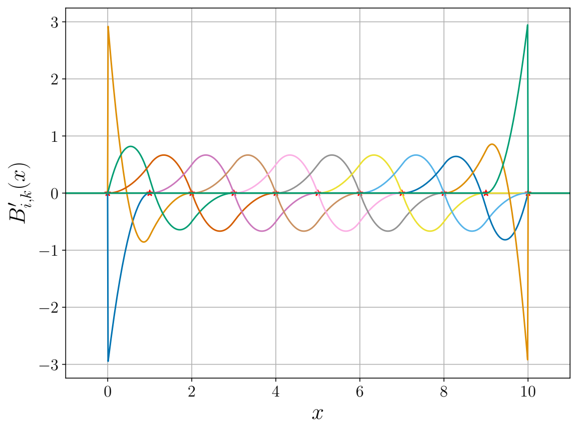
We implement two classes in `C++`: `b_splines.h` and `collocation.h`. These are templated such that they support being used with any desired flaoting-point type, e.g. `float` or `double`.

The code is enclosed with this report and can be found at [1].

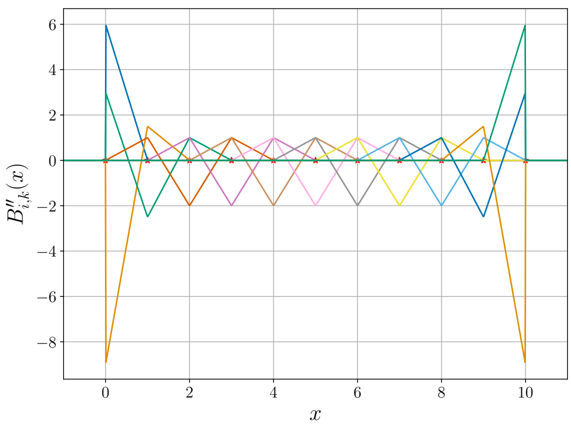
The `b_splines.h` class can either generate a spaced series of knot points or be called with an externally generated set of them. It automatically appends ghost points and implements the exact form of the first two derivatives, assuming k is large enough.



(a) $B_{i,k=4}$



(b) $\partial_x B_{i,k=4}$



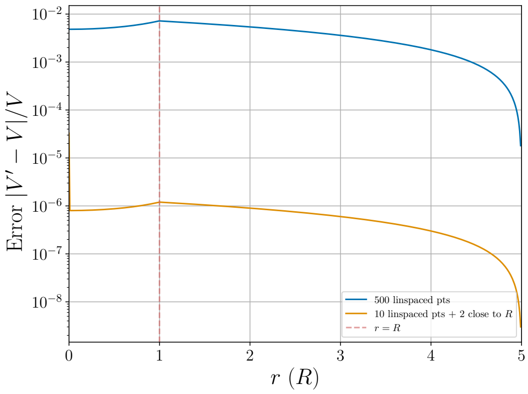
(c) $\partial_x^2 B_{i,k=4}$

Figure 1: $k = 4$ B-splines on $[0, 10]$ (1a) and their first (1b) and second (1c) derivatives with eleven evenly spaced knot points. The splines in 1a sum to unity on the entire interval. The first derivative is smooth, while the second derivative is continuous but not differentiable.

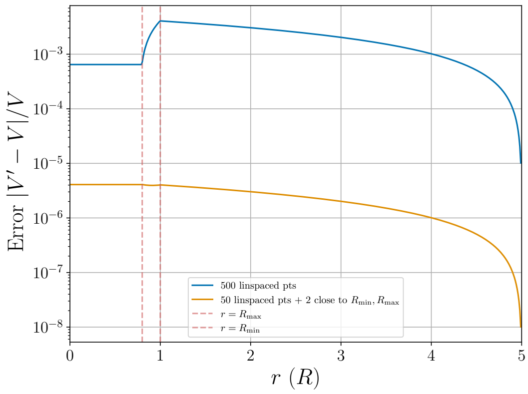
`collocation.h` implements the collocation method described above for $n = 2$ and $k = 4$. It must be called with references to functions that return the right-hand side and the q_i. It internally creates a `b_splines` object and sets up the coefficient matrix. The matrix system is solved by calling `solve()`, upon which the solution function can either be called directly or be saved at discrete points to a file.

Exemplary splines and their derivatives for $k = 4$ are shown in figure 1. It can be seen in 1a that the splines sum to unity everywhere as required and are non-zero only on the expected intervals. The first derivative is smooth, while the second derivative is continuous but not differentiable.

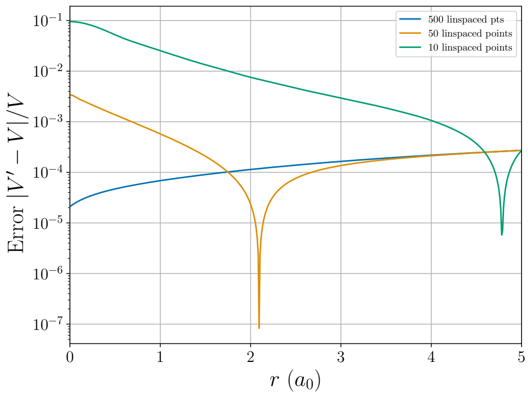
For the three charge densities suggested in the assignment, we present the relative deviation of the numerically obtained potential from the exact result in figure 2. It can be seen that accounting for the discontinuities in ρ is much more important for a good accuracy than using many knot points, see 2a, 2b as using 12 smartly placed knots vastly outperforms 500 naively



(a) Homogeneously charged sphere



(b) Homogeneously charged spherical shell



(c) Hydrogen ground state

Figure 2: Relative error of the numerically obtained potential for different charge densities and different numbers / distributions of knot points. Accounting for the discontinuities in ρ is much more important for a good accuracy than using many knot points, see 2a, 2b. The exponential charge distribution of hydrogen 2c poses a problem for linearly spaced knots and many are needed for good accuracy.

placed ones in the case of the solid sphere.

Furthermore, the exponential charge distribution of hydrogen 2c poses a problem for linearly spaced knots and many are needed for good accuracy. A better approach not explored here might be to use an exponential distribution of points for this charge density.

5 Results

We calculate the potential for the following charge densities and with the following knot point distributions:

1. Homogeneously charged sphere:

$$\rho(r) = \begin{cases} \frac{Q}{V}, & r \leq R \\ 0, & \text{else} \end{cases} \tag{11}$$

   10 evenly spaced knot points on $[0, 5R]$ plus two close to the discontinuity at $R \pm 1 \cdot 10^{-6} R$

2. Homogeneously charged spherical shell with $R_{\text{min}} = 4/5\, R_{\text{max}}$:

$$\rho(r) = \begin{cases} \frac{Q}{V}, & R_{\text{min}} \leq r \leq R_{\text{max}} \\ 0, & \text{else} \end{cases} \tag{12}$$

   50 evenly spaced knot points $[0, 5R_{\text{max}}]$ plus two close to the discontinuities each at $R_{\text{min/max}} \pm 1 \cdot 10^{-6} R_{\text{max}}$

3. Hydrogen ground state:

$$\rho(r) = \frac{e}{\pi a_0^3} \mathrm{e}^{-2r/a_0} \tag{13}$$

   500 evenly spaced knot points on $[0, 5a_0]$

4. $2s$-state:

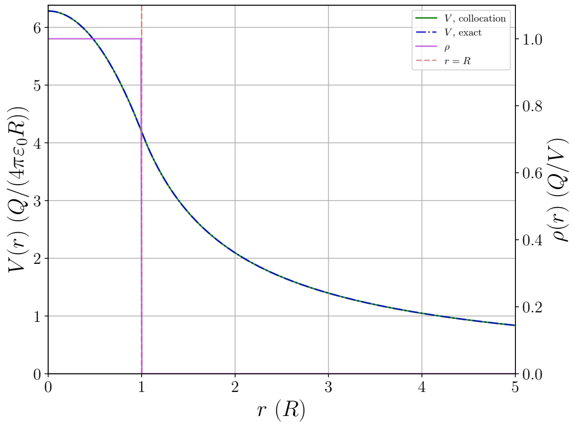
$$\rho(r) = \frac{e}{32\pi a_0^3} \left(2 - \frac{r}{a_0}\right) \mathrm{e}^{-r/a_0} \tag{14}$$

   1400 evenly spaced knot points on $[0, 14a_0]$

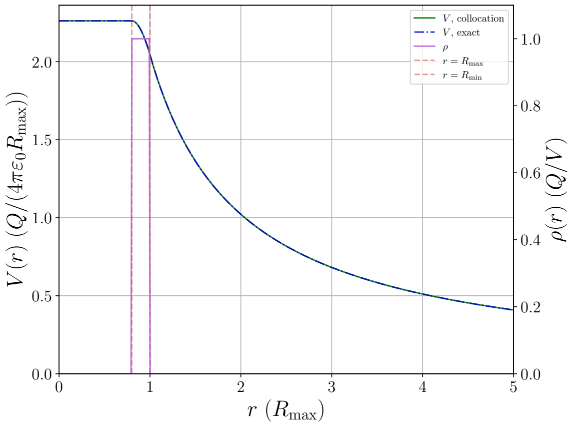
The appropriate boundary conditions for $\varphi(r)$ are $\varphi(0) = 0$ and $\lim_{r\to\infty} \varphi(r) = Q/(4\pi\varepsilon_0 r)$ where $Q = e$ for hydrogen. In our case, $\infty = 5R$.

The solutions to Poisson's equation obtained with the collocation method can be seen in 3. For the first three, the exact solutions are also shown, and we observe good to excellent agreement.

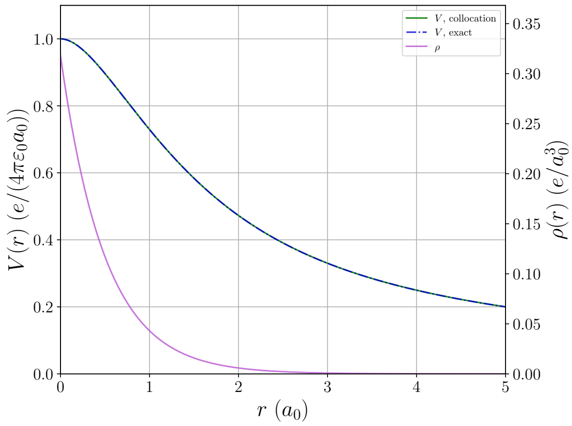
The potential for the $2s$-state of Hydrogen shows a bend at $r = 2a_0$ where the charge density vanished, which appears reasonable.



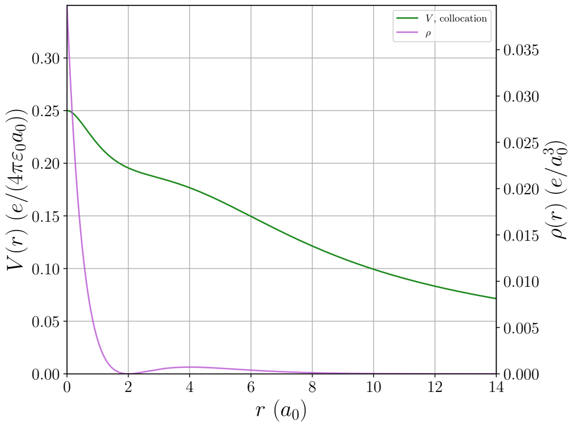
(a) Homogeneously charged sphere



(b) Homogeneously charged spherical shell



(c) Hydrogen $1s$-state



(d) Hydrogen $2s$-state

Figure 3: Numerically obtained electrostatic potentials for different charge densities.

6 Conclusion

We have successfully implemented B-splines and the collocation method in a high-level language and applied it to Poisson's equation, achieving good results in comparison to the exact solutions.

References

[1] Max Maschke. *compphys*. URL: https://github.com/max-mas/compphys (visited on 04/22/2024).