Bsplines Collocation

FK8029 - Computational Physics

Andreas Evensen

1 Introduction

In this report, we seek to solve Poission's equation, using callocation of basis-functions, namely B-splines. This, in contrast to finite element methods, allows for a non-uniform grid which can be useful in certain situations, e.g. points of discontinuities. Using collocation, we can solve the Poission equation, where the charge density ρ is known, and the potential V is to be found.

2 Theory & Method

Using finite element methods to solve certain problems works fine enough when you have a well-defined grid. However, when the grid is ill-defined, or a grid is not an option, we can use finite difference methods in order to solve our problem. The concept of Bsplines – basis splines, are that we can defined a set of basis functions and thus represent any function f on the chosen basis set. This allows for the function to be well-defined in a domain, in contrast to where in finite element methods, we function is only defined at certain points.

Suppose that we then have a set of points r_1 to r_n, of which are irregularly placed. We can then define the basis-function as follows:

$$B_i^1(x) = 1 \text{ if } r \in [r_i, r_i + 1) \text{ else } 0, \tag{1}$$

$$B_i^k(x) = \frac{r - r_i}{r_{i+k-1} - r_i} B_i^{k-1}(x) - \frac{r_{i+k} - x}{r_{i+k} - r_{i+1}} B_{i-1}^{k-1}(r). \tag{2}$$

Where we evaluate a given point r in the neighborhood of the current position, depending on the order of the spline k.

In this report our objective is to solve the Poission eq: $\nabla^2 V = \rho(x)$, where ρ is the charge density function. Rewriting our Poission equation, and utilizing the that we evaluate at the points x_i (knot-points), then we find the expression below:

$$\sum_{n=i-3}^{i-1} \left(c_n \partial_r^2 B_i^k(x_i) \right) = -x_i \frac{4\pi \rho(x_i)}{r\pi\epsilon_0} \tag{3}$$

This implies, that we will have a matrix equation of size $n - k$, where n is the number of knots, including ghost-points, and k is the spline order, in this case $k = 4$. This is referred to as collocation. Two boundary conditions exists, one on the left and one on the right. The left boundary conditions states that there is no charge at $x = 0$, which is implied by setting the first spline to be 0. The right boundary condition states that far away, the charge density looks like a point charge, which implies that the right-hand side of eq (3) is simply the charge.

We have three charge densities, one for a uniformly charged sphere, one for a spherical shell, and one for the ground state of hydrogen, all defined below:

$$\rho_{sphere}(r) = \frac{3Q}{4\pi R^3}, \tag{4}$$

$$\rho_{shell}(r) = \frac{3Q}{4\pi \left(R_o^3 - R_i^3 \right)}, \tag{5}$$

$$\rho_{1s}(r) = \frac{e}{\pi a_0^3} \exp\left(-\frac{2r}{a_0} \right). \tag{6}$$

For the purpose of this report, we define the following scaling units:

Table 1: Unit conversions

4π	a_0	e	a_0
1	1	1	1

Furthermore, we also define the charge Q to be equal to one.

3 Result & Discussion

I began implementing the definitions in `cpp` but ended up switch to `python` after some difficulities in definiting the linear equations defined in eq (3). We first defined the physical knot sequence $t = i \in [0, 4]; i \in Z^+$, using cubic splines, our Poission equation requires C^2-smooth functions, we obtain the following splines.

(a) BSplines python package

(b) BSplines own implementation
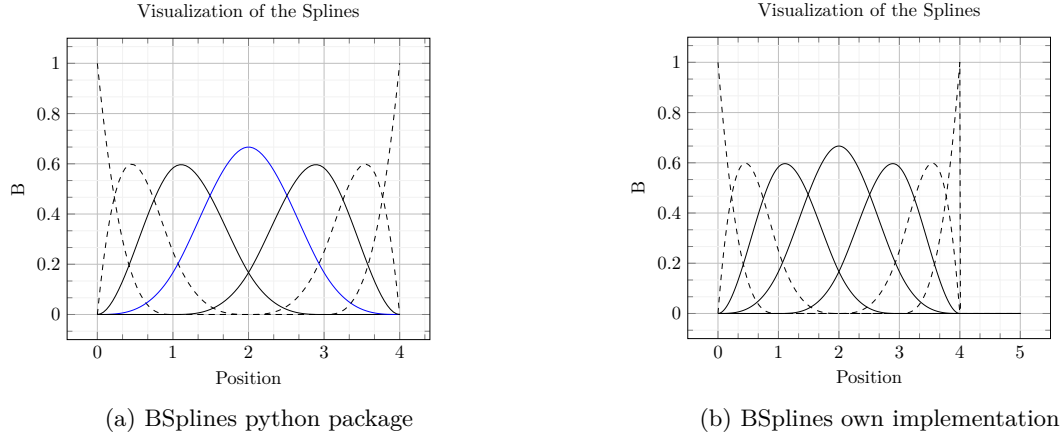
In fig 1b we have computed values to the right of the last physical knot point, which then is zero, as per definitions, and in fig 1a one shows the same knot-sequence, but where the domain ends at the last physical knot point. The area of each spline is equal to one, as per construction of the splines. Solving eq (3) where the charge density ρ is given by (4), where the radius R is defined to be 10-a.u, yields the following result.
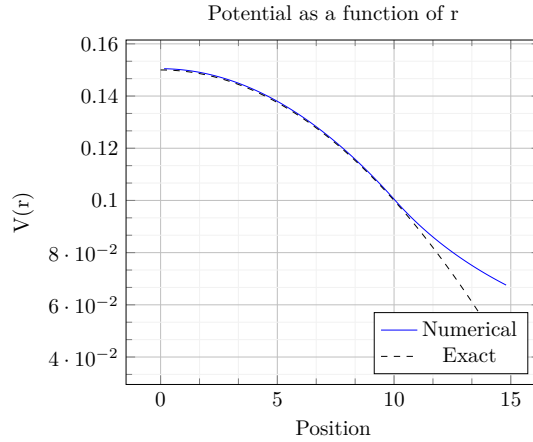

Figure 2: Uniform distribution

The potential $V(r)$ decreases as distance from the center increases. This is expected, as given by

the exact solution, known from electromagnetism. The exact solution has the following form:

$$V(R) = \frac{Q}{8\pi\epsilon_0 R}\left(3 - \left(\frac{r}{R}\right)^2\right),$$

which exhibits the same behavior, as shown in the above figure. The solution differs slightly compared to the exact solution, after $R = 10$, which is a result of the boundary condition of the right side. To account for this, we could increase the number of physical knots, which would yield a better resolution after $R = 10$.

When computing the potential for a spherical shell, where the inner radius $R_i = 5$-a.u and the outer radius $R_o = 10$-a.u, we obtain the following result.
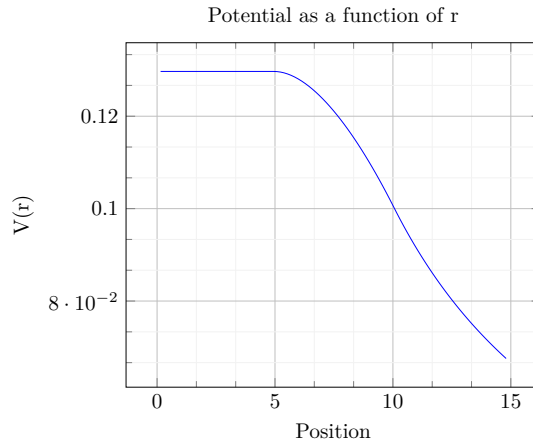

Figure 3: Shell distribution

The potential is constant up to the inner radii, which is expected. The potential V then drops as we move away from the void in the center. This behavior is expected, as the charge density is zero in the void, and the charge density is constant in the shell, as given by eq (5).

Lastly, the potential $V(r)$ was computed for the ground state of the hydrogen atom, i.e. state $1s$, where the charge density ρ is given by eq (6). The potential was computed for $r \in [0, 15]$-a.u, and the result is shown in fig 4.
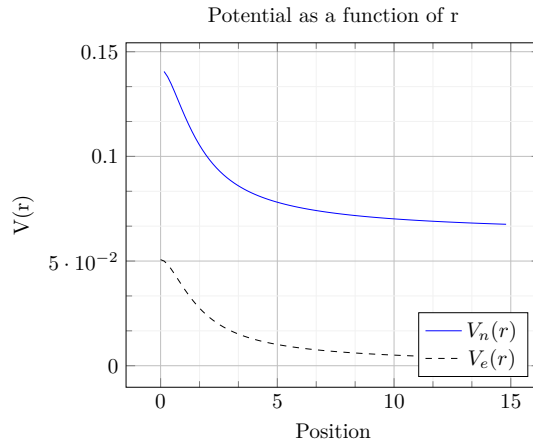

Figure 4: Potential energy of the ground state of hydrogen

3

The potential $V(r)$ is decreasing rapidly as we move away from the center, which is expected, as the charge density is decreasing exponentially as we move away from the center, as given by eq (6). Moreover, there is a discrepancy between the numerical, and exact solution. The exact solution seems to be shifted downwards, in comparison to the numerically obtained solution, even though their overall shape is in agreement. Most plausible reason for this, is the units used in the numerical solutions differs from those of the exact solution. Due to lack of time, I was not able to investigate this further.

4 Conclusion

The Poission equation was solved using the fundamental theory of basis functions, namely cubic b-splines in this report. This allows for non-uniform grid points, and can easily be extended to any general second order differential equation. This is in favor to other finite-difference methods, where the grid defines the function, instead of the function defining the grid.

The potential $V(r)$ was computed for three different charge densities, a uniformly charged sphere, a spherical shell, and the ground state of hydrogen, and all results were expected, and in rough agreement with the exact solutions. In contrast to previous tasks, it was problematic to implement the system of equations, mostly due to index counting. This was eventually solved, but a lot of time was wasted in order to make the system of equations working, in both a `cpp` and `python` environment.