Praktisk programering - Homeworks

June 21, 2022

Simon S. Villadsen, 201804389

My student id ends with 89 so $89 \equiv 20 \pmod{23}$. The Akima spline interpolation is implemented in subsection 1.5.

1 Homeworks

1.1 Jacobi elimination

Methods
`timesJ`, `Jtimes`, `jacobi_cyclic` in `jacobi.cs`; `getrandomsymmetric` in `matrix.cs`; `test_jmd`, `hydrogen_jmd` in `Program.cs`; `plot_hydrogen_eigens` in `Form.cs`;

Really a given Jacobi matrix $J(p,q)$ can be defined as the identity matrix except for four coordinates $ij \in \{pq, qp, pp, qq\}$ [0]. Let us take a look on the defintion of the matrix product $P = AJ$

$$P_{ij} = \sum_{k=0}^{n} A_{ik} J_{kj} \tag{1}$$

So for a given coloumn j, P_{ij} only differ from $A_{i,j}$ if $j = p$ or $j = q$. We basically only need to calculate the product along these two colomuns which is done in $\mathcal{O}(n)$ time. The conclusion is similar when computing JA.

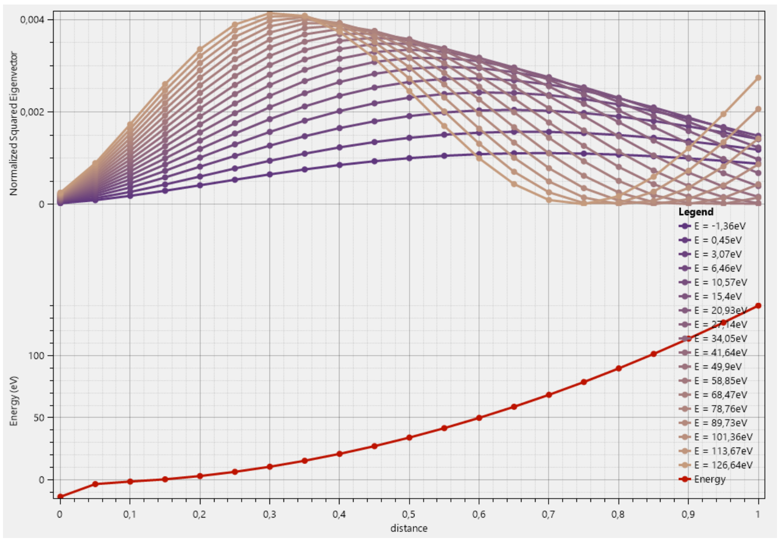

Figure 1: Hydrogen eigenvalue and -vector plot, with $dr = 0.05$ and $r_{\max} = 20$.

For the schödinger equation we solve the eigenvalue equation $\mathbf{H}f = \epsilon f$, by diagonalising $\mathbf{H} = VDV^T$. The eigenvalues lie along the diagonal of D and the eigenvectors are the coloumn vectors of V. We get the lowest and largest energy state values as where we set $dr = 0.1$ and $r_{\max} = 20$.

$$-0.49875621120889385 \text{ Hatree} * 27.211386245988 \frac{\text{eV}}{\text{Hatree}} = -13.5718 \text{ eV} \tag{2}$$
$$199,89946101554284 \text{ Hatree} * 27.211386245988 \frac{\text{eV}}{\text{Hatree}} = 5.43954 \text{ keV}. \tag{3}$$

1.2 Least Squared method

Methods
`QR_leastsq_fit` in `leastsq.cs`; `radioactivity_leastsqs` in `Form.cs`; `QR_decomposition`, `solve_uppertriangular_equation`, `leastsq_matrix` in `matrix.cs`;

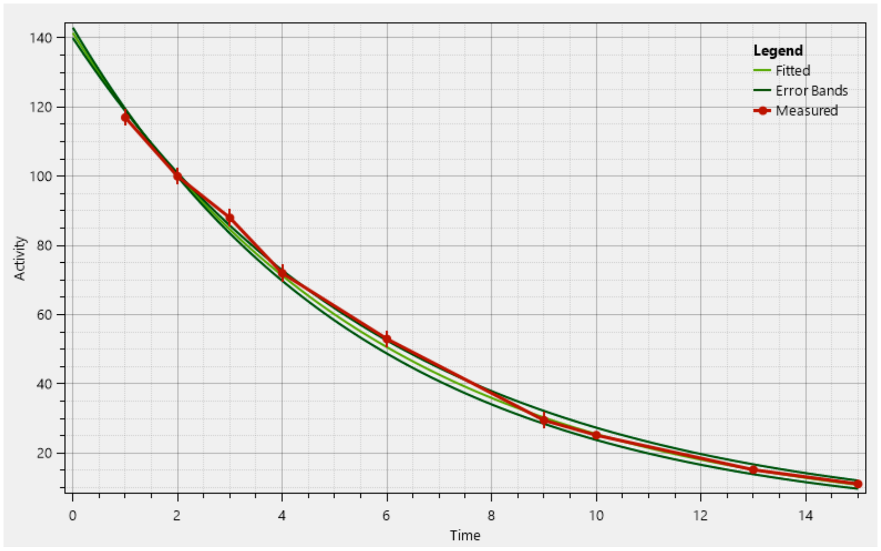

Figure 2: Least square QR-decomposition method for radioactivity data for ^{224}Ra. The fitted line is the function $a\exp(-\lambda t)$, where $a = 141,45$ and $\lambda = 0,1712$ day^{-1}. The atom's half life is $\tau = \frac{\ln 2}{\lambda} = 4.05$ days. The modern day half life value is 3.6 days, and the discrepancy may be due to limited measurement equipment, which fail to omit background radiation. The function was fitted with Givens rotation.

The Jcobi rotation matrix is the same as the Givens rotation matrix [Fedorov, p. 20]

$$G(p,q,\theta) = \delta_{qp}, \quad (p \neq q) \tag{4}$$
$$G(x,x,\theta) = \cos(\theta), \quad x \in \{p,q\} \tag{5}$$
$$G(p,q,\theta) = \sin(\theta), \quad G(q,p,\theta) = -\sin(\theta) \tag{6}$$

So we can resuse the `Jtimes` and `timesJ` methods from homework 1.1.

The decomposition $A = QR$ where Q is orthogonal and R is upper tringaular is not unique. Let A have dimension $m \times n$. Q can have dimension $m \times m$ and R hame $m \times n$ OR another solutiuon lets Q have dimension $m \times n$ and R have $n \times n$. It appears that our implementation of Givens gives one solution and Gram-Schmidt the other. For the Givens method, the dimension $\boldsymbol{b}$ vector need be extended in the system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$. Ex. to solve $QR\boldsymbol{x} = \boldsymbol{b} \Rightarrow R\boldsymbol{x} = Q^T\boldsymbol{b}$,, we notice how both $\boldsymbol{x}$ and $\boldsymbol{b}$ has dimension n if Q has dimension $m \times n$. For the other case we can `resize` Q and R to fit that dimension.

1.3 Runge Kutta Methods

An explicicit Runge-Kutta methods for a differential equation $f(x, \boldsymbol{y}) = \frac{d\boldsymbol{y}}{dx}$ are given by the update

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \sum_{i=0}^{s-1} b_i \boldsymbol{K}_i, \tag{7}$$

where

$$\boldsymbol{K}_i = hf(x_n + c_i h, \boldsymbol{y}_n + \sum_{j=0}^{i-1} a_{ij}\boldsymbol{K}_j). \tag{8}$$

Here s is the order of the Runge-Kutta method; The amount of intermediate points needed to estimate the secant. Now the $\boldsymbol{y}_n$ vector values is updated with

$$\boldsymbol{y}_{n+1} \mapsto \boldsymbol{y}_n + \sum_{i=1}^{s} b_i \boldsymbol{K}_i \tag{9}$$

In theory, we can estimate $\vec{b}$ for given A and $\boldsymbol{c}$ by solving the system of linear equations

$$[\boldsymbol{K}_1, \boldsymbol{K}_2, ..., \boldsymbol{K}_s] \cdot \boldsymbol{b} = \boldsymbol{\kappa} \cdot \boldsymbol{b} = (\boldsymbol{y}_{n+1} - \boldsymbol{y}_n) \tag{10}$$

The nearest solution for b can be found by QR-Decomposition on $\boldsymbol{\kappa}$. Now define $l \in \mathbb{N}$. We propose a special set of cooefficients that is parametised by l, and gives the following Butcher's table [Fedorov, p. 65] at $l = 2$ and $s = 2$,

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1\ /2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \tag{11}$$

we could use a test function such as a gaussian

$$y(x) = e^{-x^2/2} \tag{12}$$

which gives the fortunate differential function

$$y' = f(x, y) = -xy \tag{13}$$

Here, both the variable x and y are included by their multiplicative product. Using this test method we can estimate $\boldsymbol{b}$, for $l = 2$:

$$\boldsymbol{b} \approx (0.1666782, 0.33334620, 0.33329734, 0.16667824)^T. \tag{14}$$

The estimation seems the best when κ is a square matrix and it is important to include the property

$$\sum_i b_i = 1, \tag{15}$$

when solving for the estimate vector.

We will also implement the smart RKF45 Butchers's table given in [Fedorov, p. 67]. This tableau is especially useful in calculating the error estimate, as it gives A and $\boldsymbol{c}$ values for both fourth and fifth order.

1.4 Neural network

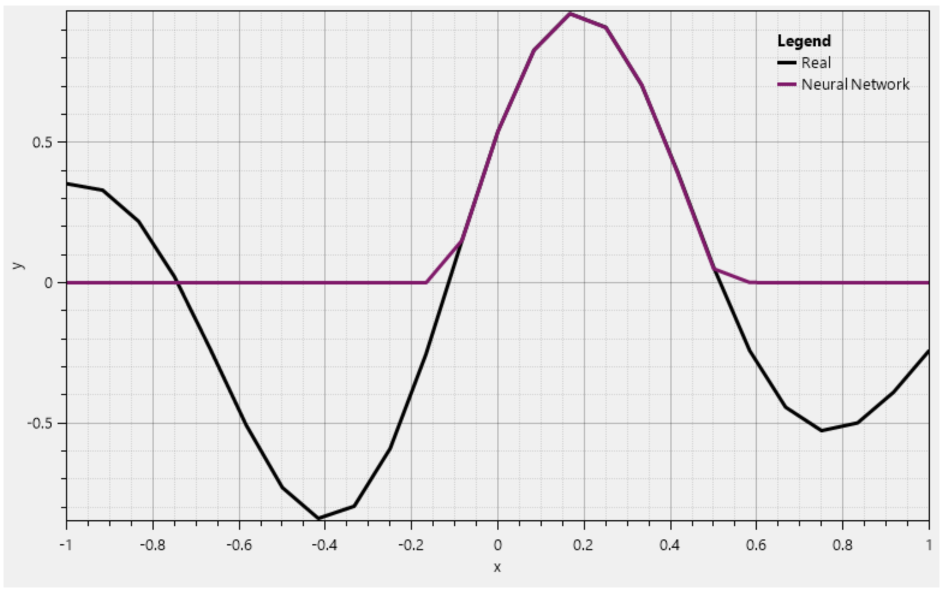

Figure 3: 25 points of the function $f(x) = \cos(5x - 1)\exp(-x^2)$ have been plotted and fitted by minimizing the cost function of a neural network with 25 hidden nodes and a gaussian activation function.

1.5 Splines

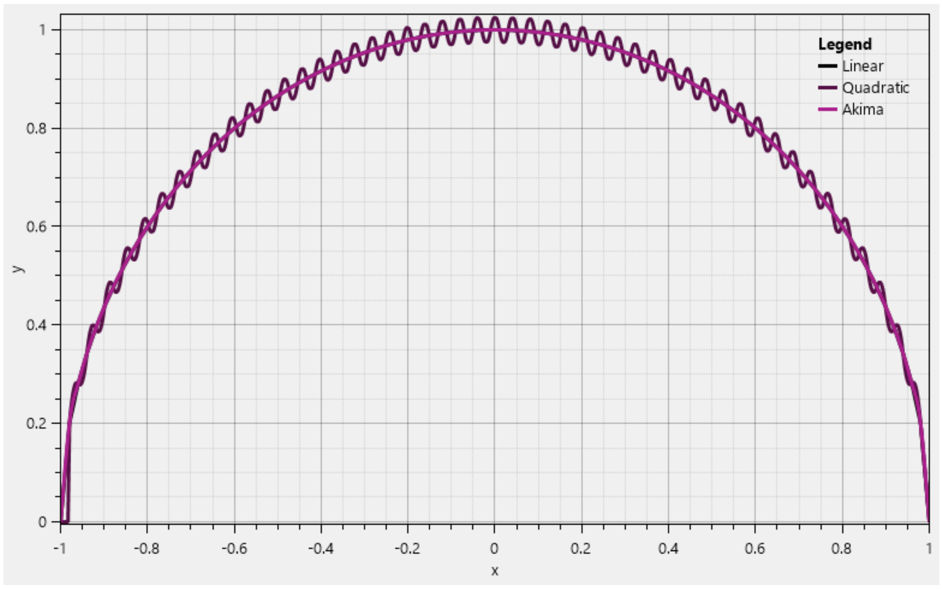

Figure 4: Spline interpolation of different types. $N = 100$ points on the unit circle are generated. The quadratic ocelates around each point. The doubled integral from -1 to 1 evaluates to 3.1382 (linear), 3.1355 (quadratic), 3.1395 (Akima)

1.6 Monte carlo

Estimating pi with normal pseudo-generation: 3.14178292. And with low discrepancy (Van der Corput and Halton sequences) the estimation is: 3.14159256. Both estimations are by 100.000.000 generated numbers.

1.7 Minimization and root-finding

Equation 6.13 in Fedorov gives a condition for the Jacobian matrix update difference $\Delta \boldsymbol{J}$:

$$(\boldsymbol{J} + \Delta \boldsymbol{J})\Delta x = \Delta \boldsymbol{f} \tag{16}$$

We choose the symmetric update given by setting $\Delta \boldsymbol{J} = \boldsymbol{u}\boldsymbol{u}^T$, and update the Jecobiann accordingly. We find the inverse with our previous matrix methods. Then $\Delta \boldsymbol{x} = -\lambda \boldsymbol{J}^{-1}\boldsymbol{f}(\boldsymbol{x})$, where $0 < \lambda \leq 1$ is found by linesearch.

Likewise, equation 7.9 in Fedorov gives a condition for the hessian matrix update difference $\delta \boldsymbol{H}$:

$$\nabla f(\boldsymbol{x} + \delta \boldsymbol{x}) - \nabla f(\boldsymbol{x}) = (\boldsymbol{H}(\boldsymbol{x}) + \delta \boldsymbol{H})\delta \boldsymbol{x} \tag{17}$$

We choose the symmetric update given by setting $\delta \boldsymbol{H} = \boldsymbol{u}\boldsymbol{u}^T$, and update the inverse Hessian accordingly. Then $\Delta \boldsymbol{x} = -\lambda \boldsymbol{H}^{-1}\nabla f(\boldsymbol{x})$, where $0 < \lambda \leq 1$ is found by linesearch.

2 References

1. `https://en.wikipedia.org/wiki/Givens_rotation`

2. `https://phys.au.dk/~fedorov/Numeric/now/Book/main.pdf`