Chaos, Solitons and Fractals 167 (2023) 113043

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

Machine learning techniques in magnetic levitation problems

Manuel Arrayás *, José L. Trueba, Carlos Uriarte

Área de Electromagnetismo, Universidad Rey Juan Carlos, Tulipán s/n, Móstoles, 28933, Madrid, Spain

ARTICLE INFO

Keywords:
Magnetic levitation
Machine learning
Stability regions

ABSTRACT

We present a method for calculating the stability region of a perfect diamagnet levitated in a magnetic field created by a circular current loop making use of the machine learning techniques. As an application we compute stability regions, points of stable equilibrium and stable oscillatory motions in two chip-based superconducting trap architectures used to levitate superconducting particles. Our procedure is an alternative to a full numerical scheme based on finite element methods which are expensive to implement for optimizing experimental parameters.

1. Introduction

Levitation [1] has been the guiding idea for the very recent design of a device to explore superfluidity regimes [2]. In order to levitate diamagnetic materials, it is crucial to find the stability region in a given magnetic field configuration. The problem reduces to identify the place where the energy of the system obtains a minimum. To achieve levitation, the total force on the object must vanish thus the gradient of the energy should be zero. However, that condition is not enough, we also need to impose that the Laplacian of the energy will be positive, so the equilibrium is stable. Solving the problem involves to get the magnetic fields and their derivatives, mostly in forms of gradients and Hessians. Here is where some techniques of machine learning are applied, in particular automatic differentiation (AD) which is at the heart of deep learning [3].

We will study the case of finding the stability region in the field created by circular current loops, which involves the use of elliptic functions. There are available ready-to-try packages for deep learning applications ranging from image and speech recognition, genome-based taxonomic classification, drug research and so on. Normally the standard functions used in these packages are limited to a set of basic functions, like the sigmoids [4]. When considering the magnetic field created by a circular loop we will be dealing with elliptic functions. We will code explicitly the AD rules for elliptic integrals of the first and second kind [5].

Once we have an appropriate method to deal with magnetic levitation techniques, we will apply it to find the stability regions of some recent chip-based superconducting magnetic traps [6,7], which may give us as a promising platform for performing quantum experiments with microparticles. The full numerical scheme based on finite element methods are expensive to implement in terms of time and resolution which is inconvenient to optimize experimental parameters. Our approach allows a fast way to explore and optimize the design of actual prototypes of magnetic traps.

2. The field of a circular current loop

Let us summarize the classical solution of finding the field created by a circular current loop of radius a [8]. We take our origin of coordinates at the center of the loop and xy the plane containing the loop. Thus the current density is given by

$$I\,d\mathbf{s} = -I\sin\phi\,\mathbf{i} + I\cos\phi\,\mathbf{j}, \tag{1}$$

the position $\mathbf{r}_i$ of the loop by

$$\mathbf{r}_i = a\cos\phi\,\mathbf{i} + a\sin\phi\,\mathbf{j}, \tag{2}$$

and, defining $R = |\mathbf{r} - \mathbf{r}_i|$, we have

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 I}{4\pi}\oint \frac{d\mathbf{s}}{R}, \tag{3}$$

for the vector potential in the Coulomb gauge. Taking cylindrical coordinates (r, ϕ, z), it can be proved [9] that

$$\begin{aligned} A_\phi &= \frac{\mu_0}{4\pi}\frac{4aI}{\sqrt{(r+a)^2+z^2}}\left(\frac{2-k^2}{k^2}K(k^2) - \frac{2}{k^2}E(k^2)\right) \\ &= \frac{\mu_0}{4\pi}4I\sqrt{\frac{a}{r}}\left[\left(\frac{1}{k}-\frac{k}{2}\right)K(k^2) - \frac{1}{k}E(k^2)\right], \end{aligned} \tag{4}$$

where $k^2 = 4ar/[(r+a)^2+z^2]$ and the elliptic integrals of the first and second type are defined as

$$K(m) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1-m\sin^2\theta}}, \quad E(m) = \int_0^{\frac{\pi}{2}} d\theta\,\sqrt{1-m\sin^2\theta}. \tag{5}$$

* Corresponding author.
E-mail address: manuel.arrayas@urjc.es (M. Arrayás).

https://doi.org/10.1016/j.chaos.2022.113043
Received 22 November 2022; Accepted 19 December 2022
Available online 23 December 2022

The magnetic field can then be calculated as

$$B_r = -\frac{\partial A_\phi}{\partial z}, \quad B_z = \frac{1}{r}\frac{\partial (rA_\phi)}{\partial r}, \tag{6}$$

which already involves derivatives of the elliptic integrals.

3. The stability conditions for a perfect diamagnet

Having obtained the magnetic field in Section 2, we introduce the stability conditions for a perfect diamagnet [1]. When an isotropic linear magnetic material of mass m is introduced in a magnetic field, the induced magnetic moment reads

$$\mathbf{m}(\mathbf{r}) = \frac{\chi \mathbf{B}(\mathbf{r})}{\mu_0}, \tag{7}$$

where χ is the magnetic susceptibility and μ_0 is the permeability of vacuum. The change in the free energy f of the system kept at constant temperature reads

$$df = mg\, dz + \mathbf{m} \cdot d\mathbf{B}. \tag{8}$$

For a perfect diamagnet, we have $\chi = -1$ and the free energy reads

$$f(B) = f_0 + mgz - \frac{B^2}{2\mu_0}, \tag{9}$$

where f_0 is the free energy of the body at zero field and B^2 is the squared modulus of the magnetic field. At equilibrium, we must have a minimum, so the conditions to fulfill are

$$\begin{aligned} df &= 0, \\ d^2 f &< 0. \end{aligned} \tag{10}$$

The second condition provides the stability of the levitation. Using (9), it turns into

$$\nabla^2 B^2 > 0. \tag{11}$$

In consequence, to levitate a perfect diamagnet in a magnetic field, we need not only that the total force on the diamagnet is zero (condition $df = 0$ in (10)), but also that Eq. (11) holds.

4. Numerical computation of stability conditions using automatic differentiation

4.1. The method

Numerical differentiation based on finite difference approximations are easy to implement but can be inaccurate due to truncation errors. Normally for computing gradients and Hessian one can use an interpolation scheme, but can be highly inefficient and the scaling is very poor [3]. Manual differentiation and symbolic methods often result in rather long expressions as the complexity of derivatives computed by symbolic differentiation grows exponentially, [10] and afterwards we need to evaluate numerically the final expression.

Automatic differentiation (AD) allows the evaluation of derivatives at machine precision. If we arrange the inputs in a vector $\mathbf{x}$ and the outputs in a vector $\mathbf{y}$ we get the deep learning function $\mathbf{y} = F(\mathbf{x})$. Normally the deep learning function have the form $F(\mathbf{x}) = L_k(N_{k-1}(L_{k-1}(N_{k-1}(...N_1(L_1(\mathbf{x})))))$ where L and N are linear and non-linear functions. The linear functions L are affine functions $L_i(\mathbf{x}) = A_i\mathbf{x} + \mathbf{b}_i$ where the A_i and $\mathbf{b}_i$ are called the weights, and i runs across the so called neural layers. The problem in deep learning is to find the weights to minimize a total loss function over the sample data [11]. This is an optimization problem like finding the stability region for the levitation of our diamagnet.

In AD there are two variants. The forward mode differentiation, represented symbolically as $(\partial/\partial X)$ and the reverse mode or back propagation represented as $(\partial Z/\partial)$ [12]. AD is based on the basic fact that ultimately all numerical computations involve a finite set of elementary operations. For example to compute $E(k^2)$ defined in Section 2 the finite steps needed are

$$\begin{aligned} y_1 &= 4a, \\ y_2 &= y_1 * r, \\ y_3 &= r + a, \\ y_4 &= y_3^2, \\ y_5 &= \frac{y_2}{y_4}, \\ y_6 &= z^2, \\ y_7 &= y_5 + y_6, \\ y_8 &= E(y_7). \end{aligned} \tag{12}$$

Those steps constitute a Wengert list [13] which can be represented in a directed computational graph. When calculating derivatives, the graph is augmented with the extra calculations involved. The computational problem is how to travel the graph in an efficient way and that determines the two implementations of AD, the forward and the backward mode [3].

Forward mode can be implemented very efficiently using dual numbers [14]. When we calculate the derivatives, the number of terms in the Wengert list starts growing. For example, compare the lists to compute $E(x^2)$,

$$\begin{aligned} y_1 &= x^2, \\ y_2 &= E(y_1), \end{aligned} \tag{13}$$

and $dE(x^2)/dx$,

$$\begin{aligned} y_1 &= x^2, \\ y_2 &= E(y_1), \\ dy_1 &= x^1, \\ dy_2 &= 2 * dy_1, \\ dy_3 &= 2 * y_1, \\ dy_4 &= y_2/dy_3, \\ dy_5 &= dy_4 * dy_2, \\ dy_6 &= K(y_1), \\ dy_7 &= -dy_6, \\ dy_8 &= 2 * y_1, \\ dy_9 &= dy_7/dy_8, \\ dy_{10} &= dy_9 * dy_1, \\ dy_{11} &= dy_5 + dy_{10}. \end{aligned} \tag{14}$$

We could try to remove from the list the values that will not be needed in further steps, for example after evaluating dy_5 we could forget about dy_4. However, y_2 is used at step dy_4 so it needs to be kept during further steps.

Dual numbers provide evaluation of Wengert list in a very efficient way. A dual number is like a complex number, where the imaginary symbol i is replaced by another symbol ϵ being a nilpotent number of index 2, i.e $\epsilon^2 = 0$. So we can write any dual number as $a + b\epsilon$. If for any function $g(x)$ we take the truncated Taylor series expansion [15] and write it as $g + g'\epsilon$ this dual representation will propagate the usual rules of differentiation, for example the product rule

$$(g + g'\epsilon)(h + g'\epsilon) = gh + (gh' + hg')\epsilon.$$

Now let us extend any real function $g(x)$ to dual numbers by defining

$$g(a + b\epsilon) \equiv g(a) + g'(a)b\epsilon.$$

The chain rule will propagate automatically using the algebra of dual numbers when taking as the argument the dual number $x + \epsilon$,

$$g(h(x + \epsilon)) = g(h(x) + h'(x)\epsilon) = g(h(x)) + g'(h(x))h'(x)\epsilon.$$

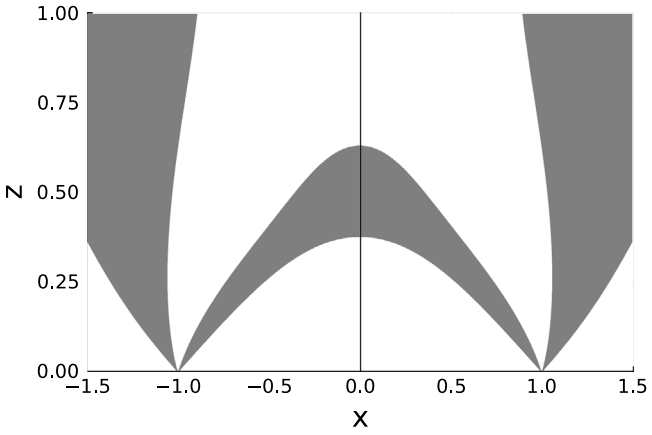

Fig. 1. The transversal plane cut xz. The shaded area is the stability region where $\nabla^2 B^2 > 0$.

Thus to implement the extension of the real functions to dual numbers we have to provide explicitly the derivative of the functions involved. In the case of elliptic integrals, from (5) it is straightforward to obtain

$$K'(m) = \frac{dK(m)}{dm} = \frac{E(m)}{2m(1-m)} - \frac{K(m)}{2m},$$
$$E'(m) = \frac{dE(m)}{dm} = \frac{E(m) - K(m)}{2m}. \quad (15)$$

For instance to find the magnetic field created by the circular loop [9] we compute the derivative of $E(k^2)$ following the dual number rules,

$$E((x+\epsilon)^2) = E(x^2 + 2x\epsilon) = E(x^2) + 2xE'(x^2)\epsilon,$$

and substituting (15)

$$\frac{dE(k^2)}{dk} = \frac{E(k^2) - K(k^2)}{k}.$$

In the same way the derivative of $K(k^2)$ is calculated,

$$\frac{dK(k^2)}{dk} = \frac{E(k^2)}{k(1-k^2)} - \frac{K(k^2)}{k}.$$

Those expressions allow us to find the explicit expression of the magnetic field using (6) and (4).

4.2. An example of implementation

For the implementation of the dual number algorithm of the AD forward mode, we have made use of the Julia language [16]. In particular we took the package *ForwardDiff.jl* [17]. That library depends on another library called *DiffRules.jl* [18]. This last library contains the file *rules.jl*, which implements the extension of the elementary functions to dual numbers. At present it lacks of the elliptic functions support, but it can be easily edited and the expressions (15) added.

We are ready to find the stability region of a perfect diamagnet in the magnetic field produced by a circular current loop. Our recipe reads as follows: taking a cross section, calculate the left hand side of (11) using the AD forward mode algorithm discussed in the previous section, and code in gray color the condition of being positive. The result is shown in Fig. 1, where the xz plane is taken. The loop is in the xy plane and cuts the xz plane at points $x = \pm 1$.

5. Levitation of nanospheres in some chip-based superconducting traps

Let us explore with our technique the problem of trapping micrometer-sized particles in chip-based superconducting devices. Chip-based superconducting trap architectures are capable of levitating micrometer-sized superconducting particles in the Meissner state [6].

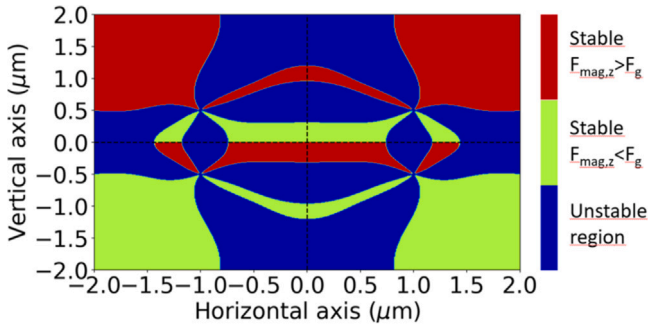

Fig. 2. Transversal plane cut xz. The areas colored in red and green are stable, with $\nabla^2 B^2 > 0$, while the areas colored in blue are unstable, with $\nabla^2 B^2 < 0$. Regions colored in red are stable points in which, moreover, the vertical component of the magnetic force is bigger than the gravity force. The horizontal component of the magnetic force should be zero in order to achieve levitation. As a consequence, the only point in which the sphere can be levitated at rest is the center of the configuration, and it can move in a vertical oscillatory motion around the center with an amplitude smaller than 0.5 μm in order to avoid blue regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

These architectures can be very useful for performing quantum experiments [7]. In order to achieve stable levitation, the magnetic force must balance the gravity force inside the stability area. That can be done by fine tuning the mass of the diamagnet and the current intensity circulating in two loops built in a chip substrate.

5.1. Anti-Helmholtz coil-trap

The first architecture studied in [6] is an anti-Helmholtz coil-trap. Two circular current loops are placed one above the other, both of them with axes along the z direction. The center of the loop in the bottom is placed at $(x, y, z) = (0, 0, -0.5)$ μm, and the center of the other loop is at $(x, y, z) = (0, 0, 0.5)$ μm. Both coils have 2 μm diameter and carry the same electric current $I = 30$ mA in opposite directions. The perfect diamagnet levitated is a sphere of 0.5 μm radius with a mass of 4.487×10^{-15} kg.

The stability regions are shown in Fig. 2, in which cylindrical symmetry is considered, so we plot the xz plane. Areas colored in red and green are stable regions, in which the total magnetic field produced by the coils satisfies the stability condition $\nabla^2 B^2 > 0$. Areas colored in blue are the unstable regions in which $\nabla^2 B^2 < 0$. Moreover, in order to achieve stable levitation, the magnetic force must balance the gravity force. We have plotted in red the regions in which the vertical component of the magnetic force on the sphere is equal or bigger than the gravity force and the region is stable. However, the horizontal component of the magnetic force has to be taken into account too, without gravity to balance it. As a consequence, it is only possible to find a point of stable equilibrium in the vertical axis $x = 0$. All these considerations lead to the following conclusions:

- The only point of stable equilibrium in this configuration is the center of the architecture, located at $(x, y, z) = (0, 0, 0)$.
- The sphere can move vertically in a oscillatory motion around the center of the configuration, but the amplitude of this motion has to be smaller than 0.5 μm in order to avoid unstable regions.

5.2. Concentrical coil-trap

The second architecture of superconducting trap [6] we consider is a double loop trap. In this case, two circular current loops are placed in the same plane, with centers at the same point $(x, y, z) = (0, 0, 0)$. The inner coil diameter is 12 μm and the outer coil diameter is 18 μm. Both coils carry the same electric current $I = 38$ mA in opposite directions.

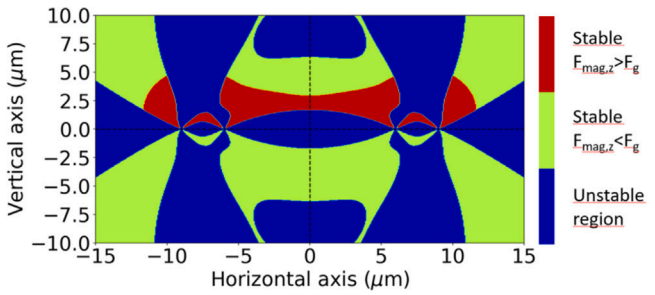

Fig. 3. Transversal plane cut xz. The areas colored in red and green are stable, with $\nabla^2 B^2 > 0$, while the areas colored in blue are unstable, with $\nabla^2 B^2 < 0$. Regions colored in red are stable points in which the vertical component of the magnetic force is bigger than the gravity force. The horizontal component of the magnetic force has to be zero too, so the only point in which the sphere can be levitated at rest is the point in the axis $x = 0$ which is in the intersection of the regions in red and green, close to $z = 3\,\mu$m. The sphere can make a vertical oscillatory motion of small amplitude around this point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The perfect diamagnet levitated is a sphere of 5 μm radius with a mass of 4.487×10^{-12} kg.

In Fig. 3, we plot the stability regions of the double loop architecture in the xz plane, because cylindrical symmetry is assumed. Regions colored in red and green are stable and regions colored in blue are unstable. In order to achieve stable levitation, we have to impose that the vertical component of the magnetic force balance the gravity force. We have plotted in red the regions in which the vertical component of the magnetic force is equal or bigger than the gravity force and the region is stable. The horizontal component of the magnetic force has to be zero, so it is only possible to find a point of stable equilibrium in the vertical axis $x = 0$. The conclusions we find in this case are:

- The only point of stable equilibrium in this configuration is the point in the vertical axis $x = 0$ which is in the intersection of the green area and the red area.
- The sphere can move vertically in a oscillatory motion around that point, but the amplitude of this motion has to be small so the sphere avoid the unstable region.

6. Conclusions

In situations related to levitation of perfect diamagnets by magnetic fields, the calculation of the stability region is critical. The need for an efficient method to replace finite element methods in order to speed up the process of tuning experimental parameters motivated our study. We have made use of some tools encountered in deed learning. In particular the concept of dual numbers, which allows us to implement the forward mode of the automatic differentiation.

We have studied the problem of finding the region of stability of a perfect diamagnet in magnetic fields created by circular current loops. The algorithms develop are applied to the case in which two chip-based superconducting trap architectures are used to levitate micrometer-sized superconducting particles [6]. Our method allows us to obtain the stability regions of both structures, the points in which particles may rest and the kind of motions that the levitating objects can make in a stable way.

These examples help to understand how machine learning methods can be used to design complex levitation experiments. Machine learning methods are well suitable for such problems. They allow us to speed up the process of exploring the parameters set-up involved in the building of actual levitation systems [2].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was funded by Universidad Rey Juan Carlos, Spain, Programa Propio: Analysis, modelling and simulations of singular structures in continuum models, M2604.

References

[1] Berry MV, Geim AK. Of flying frogs and levitrons. Eur J Phys 1997;18:307–13.
[2] Arrayás M, Trueba JL, Uriarte C, Zmeev DE. Desing of a system for controlling a levitating sphere in superfluid ^{3}He at extremely low temperatures. Sci Rep 2021;11:20069.
[3] Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: a survey. J Mach Learn Res 2018;18:1–43.
[4] Higham CF, Higham DJ. Deep learning: an introduction for applied mathematicians. SIAM Rev 2019;69:860–91.
[5] Lawden DF. Elliptic functions and applications. In: Applied mathematical sciences. vol. 80, New York: Springer-Verlag; 1989.
[6] Gutierrez Latorre M, Hofer J, Rudolph M, Wieczorek W. Chip-based superconducting traps for levitation of micrometer-sized particles in the meissner state. Supercond Sci Technol 2020;33:105002.
[7] Gutierrez Latorre M, Paradkar A, Hambraeus D, Higgins G, Wieczorek W. A chip-based superconducting magnetic trap for levitating superconducting microparticles. IEEE Trans Appl Supercond 2022;32(4):1800305.
[8] Jackson JD. Classical electrodynamics. 3rd ed. John Wiley & Sons; 1998.
[9] Landau LD, Lifshitz EM, Pitaevskii LP. Electrodynamics of continuous media. 2nd ed. vol. 8, Butterworth-Heinemann; 1984.
[10] Corliss GF. Application of differentiation arithmetic. In: Perspectives in computing, vol. 80. Boston: Academic Press; 1988, p. 127–48.
[11] Strang G. Linear algebra and learning from data. 1st ed. Wellesley-Cambridge Press; 2019.
[12] Olah C. Calculus on comp. graphs: backpropagation. 2015, https://colah.github.io/posts/2015-08-Backprop/.
[13] Wengert RE. A simple automatic derivative evaluation program. Comm ACM 1964;80.
[14] Innes MJ. https://github.com/MikeInnes/diff-zoo/.
[15] Linnainmaa S. Taylor expansion of accumulated rounding error. BIT 1976;16:146–60.
[16] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev 2017;59:65–98.
[17] https://github.com/JuliaDiff/ForwardDiff.jl.
[18] https://github.com/JuliaDiff/DiffRules.jl.