COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA ELECTROMECÁNICA

TRABAJO FIN DE GRADO

HARVESTING GREEN ENERGY WITH MAGNETS

Autor: Luis Alberto Román Ruiz

Director: Christian Bahl

Madrid

June de 2019

AUTHORIZATION FOR DIGITALIZATION, STORAGE AND DISSEMINATION IN THE NETWORK OF END-OF-DEGREE PROJECTS, MASTER PROJECTS, DISSERTATIONS OR BACHILLERATO REPORTS

1. Declaration of authorship and accreditation thereof.

The author Mr. /Ms. LUIS ALBERTO ROMÁN RUIZ

HEREBY DECLARES that he/she owns the intellectual property rights regarding the piece of work: HARVESTING GREEN ENERGY WITH MAGNETS

that this is an original piece of work, and that he/she holds the status of author, in the sense granted by the Intellectual Property Law.

2. Subject matter and purpose of this assignment.

With the aim of disseminating the aforementioned piece of work as widely as possible using the University's Institutional Repository the author hereby **GRANTS** Comillas Pontifical University, on a royalty-free and non-exclusive basis, for the maximum legal term and with universal scope, the digitization, archiving, reproduction, distribution and public communication rights, including the right to make it electronically available, as described in the Intellectual Property Law. Transformation rights are assigned solely for the purposes described in a) of the following section.

3. Transfer and access terms

Without prejudice to the ownership of the work, which remains with its author, the transfer of rights covered by this license enables:

a) Transform it in order to adapt it to any technology suitable for sharing it online, as well as including metadata to register the piece of work and include "watermarks" or any other security or protection system.

b) Reproduce it in any digital medium in order to be included on an electronic database, including the right to reproduce and store the work on servers for the purposes of guaranteeing its security, maintaining it and preserving its format.

c) Communicate it, by default, by means of an institutional open archive, which has open and cost-free online access.

d) Any other way of access (restricted, embargoed, closed) shall be explicitly requested and requires that good cause be demonstrated.

e) Assign these pieces of work a Creative Commons license by default.

f) Assign these pieces of work a HANDLE (*persistent* URL). by default.

4. Copyright.

The author, as the owner of a piece of work, has the right to:

a) Have his/her name clearly identified by the University as the author

b) Communicate and publish the work in the version assigned and in other subsequent versions using any medium.

c) Request that the work be withdrawn from the repository for just cause.

d) Receive reliable communication of any claims third parties may make in relation to the work and, in particular, any claims relating to its intellectual property rights.

5. Duties of the author.

The author agrees to:

a) Guarantee that the commitment undertaken by means of this official document does not infringe any third party rights, regardless of whether they relate to industrial or intellectual property or any other type.

b) Guarantee that the content of the work does not infringe any third party honor, privacy or image rights.
c) Take responsibility for all claims and liability, including compensation for any damages, which may be brought against the University by third parties who believe that their rights and interests have been infringed by the assignment.
d) Take responsibility in the event that the institutions are found guilty of a rights infringement regarding the work subject to assignment.

6. Institutional Repository purposes and functioning.

The work shall be made available to the users so that they may use it in a fair and respectful way with regards to the copyright, according to the allowances given in the relevant legislation, and for study or research purposes, or any other legal use. With this aim in mind, the University undertakes the following duties and reserves the following powers:

a) The University shall inform the archive users of the permitted uses; however, it shall not guarantee or take any responsibility for any other subsequent ways the work may be used by users, which are non-compliant with the legislation in force. Any subsequent use, beyond private copying, shall require the source to be cited and authorship to be recognized, as well as the guarantee not to use it to gain commercial profit or carry out any derivative works.
b) The University shall not review the content of the works, which shall at all times fall under the exclusive responsibility of the author and it shall not be obligated to take part in lawsuits on behalf of the author in the event of any infringement of intellectual property rights deriving from storing and archiving the works. The author hereby waives any claim against the University due to any way the users may use the works that is not in keeping with the legislation in force.
c) The University shall adopt the necessary measures to safeguard the work in the future.
d) The University reserves the right to withdraw the work, after notifying the author, in sufficiently justified cases, or in the event of third party claims.

Madrid, on 21 of JUNE, 2019

HEREBY ACCEPTS

Signed.....Luis Román.....

Reasons for requesting the restricted, closed or embargoed access to the work in the Institution's Repository

I, hereby, declare that I am the only author of the project report with title:

HARVESTING GREEN ENERGY WITH MAGNETS

which has been submitted to ICAI School of Engineering of Comillas Pontifical University in the academic year 2018/19. This project is original, has not been submitted before for any other purpose and has not been copied from any other source either fully or partially. All information sources used have been rightly acknowledged.

Luis Román

Fdo.: LUIS ROMÁN RUIZ Date: 16/06/2019

I authorize the submission of this project

PROJECT SUPERVISOR

Fdo.: CHRISTIAN BAHL Date: 16/06/2019

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA ELECTROMECÁNICA

TRABAJO FIN DE GRADO

HARVESTING GREEN ENERGY WITH MAGNETS

Autor: Luis Alberto Román Ruiz

Director: Christian Bahl

Madrid

June de 2019

HARVESTING GREEN ENERGY WITH MAGNETS

Autor: **Román Ruiz, Luis Alberto**.

Director: Bahl, Christian.

Entidad Colaboradora: DTU - Danmarks Tekniske Universitet.

RESUMEN DEL PROYECTO

INTRODUCCIÓN

En toda la tecnología de hoy (industria, automóviles, smartphones, informática…) la mayor parte de las pérdidas son eminentemente en forma de calor, comprometiendo así su eficiencia. Se sabe aprovechar una fuente de gran temperatura para generar electricidad, como ocurre en las centrales de generación (térmica, nuclear…) pero aún no hay ninguna solución real que evite desaprovechar todas esas pérdidas de calor cuando la temperatura no alcanza el mínimo requerido por los ciclos usados en dichas centrales.

ESTADO DE LA CUESTIÓN

Tesla propuso en 1889 una máquina rotativa que generase electricidad aprovechando una fuente de calor al descubrir que la fuerza magnética que sufrían ciertos materiales magnéticos variaba con la temperatura del material. Desde entonces se han construido algunos generadores termomagnéticos usando este principio, aunque no eran máquinas rotativas sino aparatos similares a bancos de transformadores que se calentaban y enfriaban para convertir la energía térmica en magnética y luego eléctrica. Sin embargo, dichas máquinas no han conseguido una eficacia suficiente: son grandes, costosas y generan poco, haciéndolas poco viables.

MOTIVACIÓN

Este proyecto tiene como objetivo modelar y optimizar y un dispositivo de motor Curie, posibilitando el futuro prototipado de un modelo capaz de aprovechar estas pérdidas de calor generando energía a un ritmo rentable por medio de un generador termomagnético. Así, no solo se conseguirían unas eficacias globales mayores, disminuyendo el desaprovechamiento eléctrico, también se reducirían en gran medida las emisiones de CO_2 a la atmósfera; todo ello con una tecnología limpia no contaminante. Además, el ahorro energético podría ser vital para ciertas aplicaciones

donde se dispone de una fuente de energía limitada (dispositivos portátiles, submarinos, satélites…).

OBJETIVOS DEL PROYECTO

- Presentar una amplia descripción de cómo los distintos posibles materiales termomagnéticos para el futuro prototipo se comportan cuando el ciclo es sujeto a un rango de campo variable.
- Presentar una amplia descripción de cómo los distintos posibles materiales termomagnéticos para el futuro prototipo se comportan cuando el ciclo es sujeto a un rango de temperatura variable.
- Señalar las diferencias del desempeño de los materiales en distintas situaciones.
- Escoger el material más adecuado cuando se consideran ciertas variables.
- Optimizar variables como la temperatura de forma que se obtenga el mejor rendimiento para cada material.

METODOLOGÍA

Se comenzará analizando y describiendo el ciclo, de forma que se estudie el traspaso de energía termal y magnética. En base a las características del ciclo se analizará la idoneidad de parámetros como el calor específico del material a elegir, el incremento de temperatura del material y la intensidad del campo magnético a aplicar. Teniendo en cuenta las anteriores consideraciones se elegirá el material más adecuado. Se hará un estudio del comportamiento de los materiales bajo parámetros cambiantes para cada cambio de parámetro, señalando las diferencias en su desempeño. Esto se hará para todos los materiales.

Se hará uso de abundante bibliografía para el estudio del ciclo energético (universidad, internet, libros…), así como la bibliografía propia de las empresas manufactureras de los materiales termomagnéticos. Para el análisis del material y su modelado podría usarse un software de modelado propio de la universidad DTU. EL proyecto se basará en diversos estudios anteriores del estado del arte y supondrá cálculos escritos, con Excel y MATLAB u otros.

RESULTADOS

El estudio se centró primero en un caso nominal con un cambio de temperatura de 280 – 300 K y un campo aplicado de 0 – 1 T. El estudio mostró que cambiando las temperaturas de trabajo para estar centradas en torno a la temperatura Curie de cada material podía mejorar la producción de energía, como se muestra en la tabla de la figura 1.

	ENERGY [J / m^3]				
	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
280 - 300 K (274 - 294 K for LaCaSrMnO)	4.12E+05	2.06E+05	5.17E+05	5.37E+05	2.33E+05
Centered around temperature	4.19E+05	5.79E+05	5.12E+05	5.45E+05	-

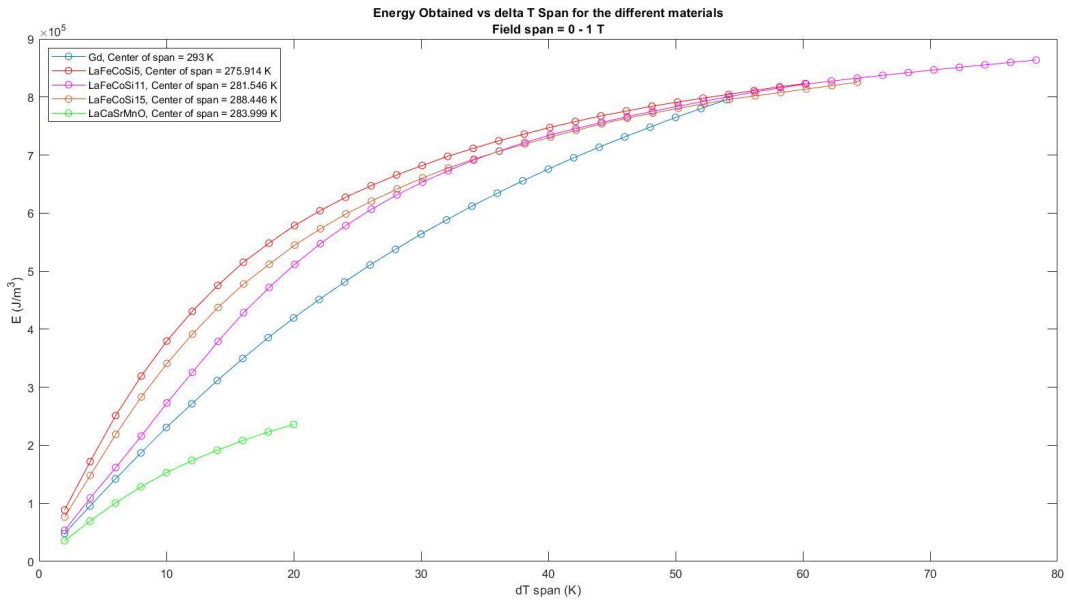
1 Centrar las temperaturas de trabajo en tono a la temperatura Curie puede mejorar el rendimiento.

Esto planteó si la temperatura Curie podría ser en todos los casos la temperatura óptima para centrar el rango de temperaturas. Finalmente, esto resultó no ser cierto, sino que la temperatura óptima no es constante sino que cambia con el campo aplicado como se puede observar en la tabla de la figura 2.La temperatura óptima decrece linealmente con el incremento del rango de temperaturas de trabajo.

	T center of maximum energy for different field spans (20 K span)				
Field	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
0 - 0.5 T	291	273.9	284.5	286.4	283
0 - 1 T	293	275.9	285.5	288.4	284
0 - 1.2 T	293	275.9	285.5	288.4	284
0.5 - 1 T	296	276.9	286.5	290.4	-

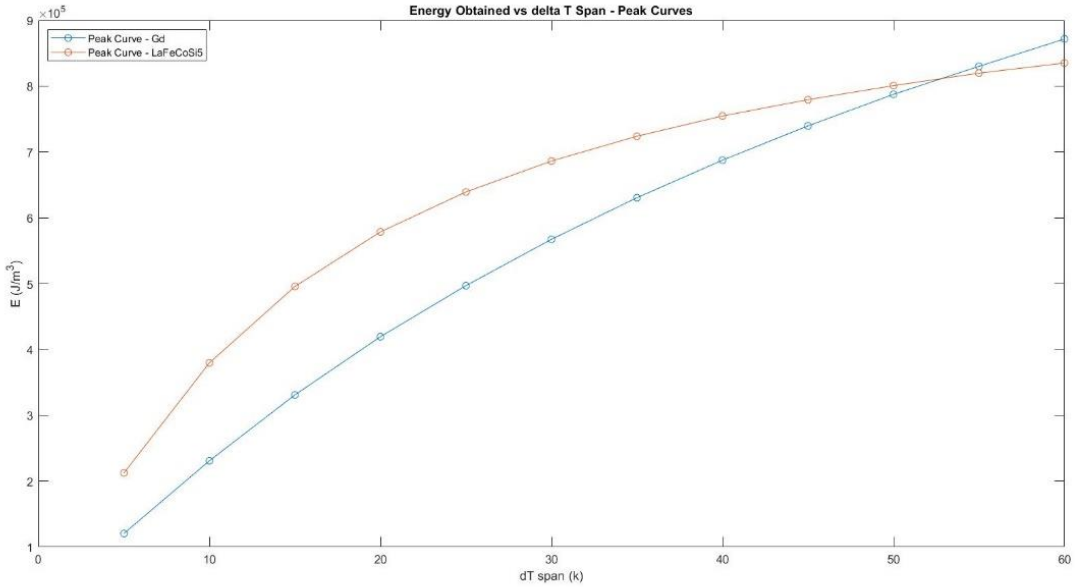
2 La temperatura óptima cambia con el cambio de campo aplicado.

También se pudo observar que la producción de energía y el rango de temperatura se relacionan de forma lineal, lo que es más apreciable para intervalos de temperatura cortos y largos como puede observarse en la figura 3.



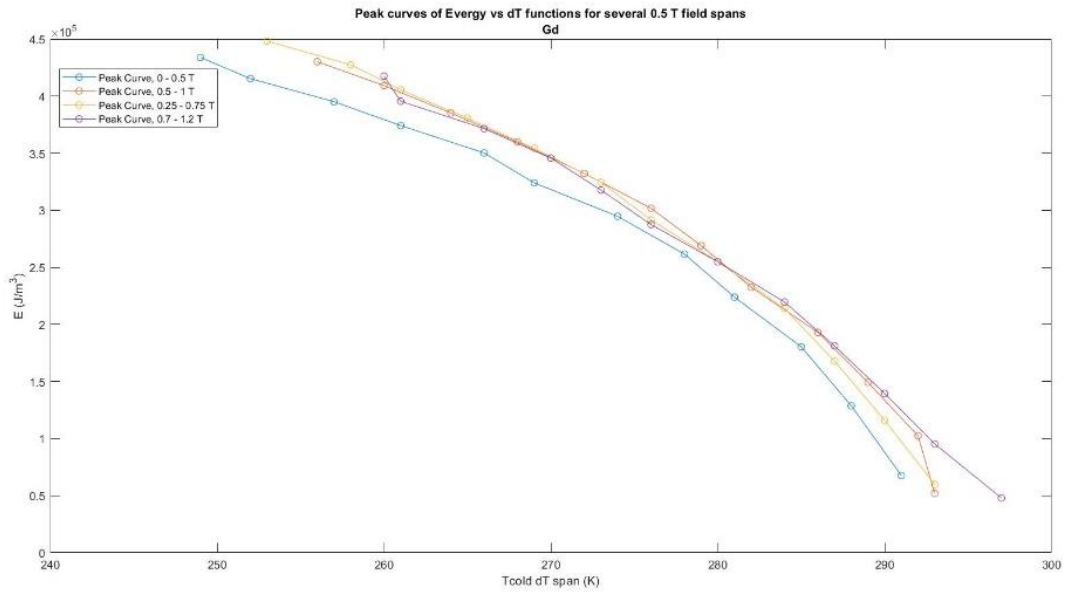
3 La producción de energía se relaciona de manera lineal para ciertos rangos de temperatura.

El estudio también reveló que el material LaFeCoSi5 otorga la mayor producción de energía excepto para intervalos de temperatura grandes, en cuyo caso superado por el Gd. Los rangos en los que uno produce más que el otro dependen del campo aplicado.



4 LaFeCoSi5 y Gd reportan la mayor energía, dependiendo del intervalo de temperatura aplicado.

Por otro lado el estudio reveló también que la producción de energía volumétrica es razonablemente la misma para cualquier intervalo de campo aplicado independientemente de dónde se sitúe este en la escala de campos, es decir, independientemente de los valores de campo en los que se sitúe. Esto se puede observar en la figura 5.



5 La producción de energía es la misma para un intervalo de campo dado independientemente de los valores en que se sitúe.

Considerando la Ley de Magnetización de Curie se presentó una fórmula para un ciclo termomagnético ideal en el que en cada fase del ciclo la temperatura o el campo son constantes:

$$E = \frac{c\,\mu}{3}(Ha^3 - Hb^3)\frac{-\Delta T}{Tc\,Th}$$

Sin embargo, esta fórmula no es siempre aplicable, tiene un error asociado a sus aproximaciones y necesita de datos experimentales para determinar los parámetros del material si estos son desconocidos, en cuyo caso los métodos de procesado y análisis computacional de los datos son preferibles. De esto se induce que no es posible obtener una fórmula para la producción de energía en un ciclo termomagnético de forma sencilla.

CONCLUSIONES

- El ciclo de un motor termomagnético tiene que desarrollarse a unas temperaturas de trabajo específicas para dar el mejor rendimiento. Estas temperaturas óptimas varían según el campo aplicado y el material considerado.

- No hay una relación clara entre la temperatura óptima y la temperatura Curie del material.

- La influencia del rango de temperaturas no es igual para todos los rangos de temperatura y es más importante para rangos menores o iguales a 20 K.

- La mayor producción de energía se consigue usando el material LaFeCoSi5 para rangos de campo pequeños e intermedios, y se alcanza para rangos grandes usando Gd.

- El rango de campo y temperatura aplicado se pueden compensar mutuamente de forma que se obtenga una gran producción de energía para rangos de temperatura o campo pequeños si el otro es lo suficientemente grande.

- La producción volumétrica de energía es la misma para un rango de campo aplicado dado independientemente de la localización de este en la escala de campos. Esto posibilita la construcción de un dispositivo termomagnético al no ser relevante los valores absolutos de campo aplicado, solo su diferencia neta.

- EL rendimiento de un ciclo en un motor Curie depende del campo aplicado, la diferencia de temperaturas a la que trabaja y el calor específico del material. Estas variables son dependientes las unas de las otras en el ciclo termomagnético, por lo que no es sencillo hallar una ecuación de producción de energía sin conocer todas estas variables.

HARVESTING GREEN ENERGY WITH MAGNETS

Author: **Román Ruiz, Luis Alberto**.

Director: Bahl, Christian.

Collaborating Entity: DTU - Danmarks Tekniske Universitet.

PROJECT SUMMARY

INTRODUCTION

Along all of today's technology (industry, automobiles, smartphones, computing…) the majority of losses are in the form of heat, harming their efficiency. Using a source of high temperature in order to generate electricity is already known, as in generation plants (thermal, nuclear…) but there is still not an actual solution that can avoid all these heat losses when the working temperature does not reach the minimum required in cycles used in this power plants.

ACTUAL STATE OF THE MATTER

In 1889 Tesla suggested a rotational machine which would generate electricity from a heat source when he considered that the magnetic force that certain magnetic materials suffer varies with the temperature of the material. Since then some thermomagnetic generators have been constructed, though not as rotative machines but transformer style devices that where heated and cooled to transform thermal energy into magnetic and then electric energy. Despite this, such machines have not achieved enough efficiency: they are large, costly and generate few, making them little viable.

MOTIVATION

This project aims to model and optimize a Curie motor device, enabling the implementation of a future prototype capable of using these heat losses generating energy at a profitable pace with the use of a thermomagnetic generator. Likewise, not only higher global efficiencies would be achieved, diminishing electric waste, also CO_2 emissions to the atmosphere would be greatly reduced; all within the use of a clean non-polluting technology. Furthermore, energy saving could be vital in certain applications where only a limited energy source is available (portable devices, submarines, satellites…).

PROJECT OBJECTIVES

- To give a broad view of how the different thermomagnetic materials behave when the cycle is subject to a variable field span.
- To give a broad view of how the different thermomagnetic materials behave when the cycle is subject to a variable temperature span.
- To highlight the difference between material performances under different conditions.
- To select the best fitting material when considering certain variables.
- To optimize variables such as temperature to give the best performance of each material.

METHODOLOGY

Firstly, the energy cycle will be analyzed and described studying the energy conversion from thermal to magnetic. Based on the cycle characteristics the convenience of parameters such as the specific heat of the material to choose, the change in temperature of material, and the value of magnetic field to be applied will be studied. With this consideration the best fitting thermomagnetic material will be chosen. A study of the behavior of materials under changing parameter will be made for each parameter change, showing the differences in performance. This will be done for all materials.

Plenty bibliography and documentation will be used for the study of the energetic cycle (from university, internet, books…), as well as documentation from the thermomagnetic materials manufacturers. For the analysis of the material and its modeling a DTU modeling software may be used. The project will be based on previous studies available form the literature and will involve handwritten calculus, calculations using Excel or MATLAB and others.

RESULTS

The study first centered on the nominal case of a field span from 0 – 1 T and a temperature span from 280 – 300 K. The study showed that changing the temperatures span to be centered on the Curie temperature could improve the energy output, as shown in the table of figure 6.

	ENERGY [J / m^3]				
	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
280 - 300 K (274 - 294 K for LaCaSrMnO)	4.12E+05	2.06E+05	5.17E+05	5.37E+05	2.33E+05
Centered around temperature	4.19E+05	5.79E+05	5.12E+05	5.45E+05	-

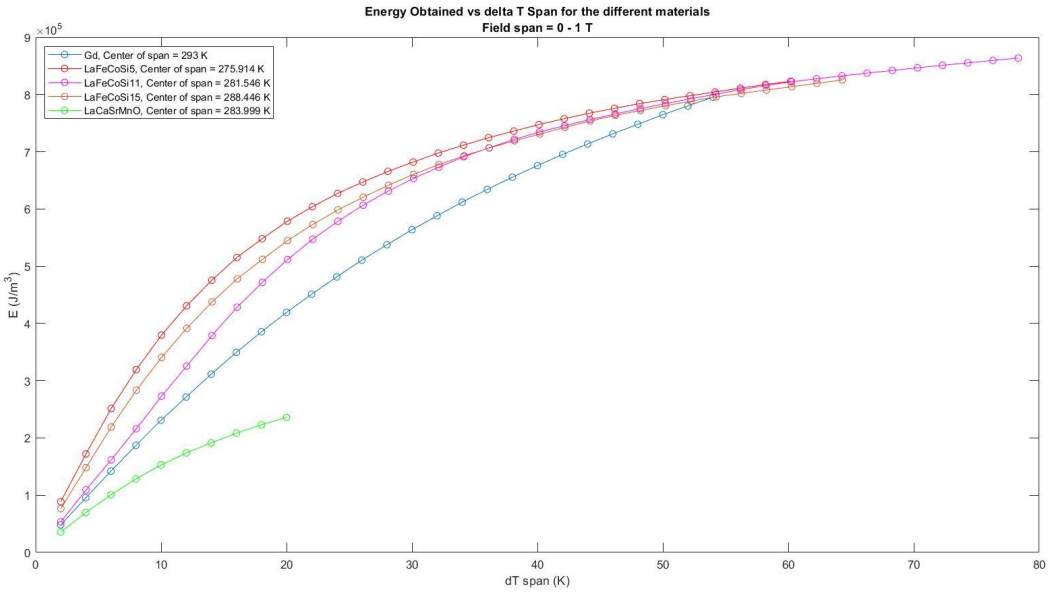
6 The energy output can be improved for certain materials if the Curie temperature is used as center of span.

This raised rose the discussion on whether the Curie temperature was the optimal temperature for the span to be centered on. However, this revealed not to be true, as the optimal temperature is not constant (Curie) but changes with the field applied as shown in the table of figure 7. The optimal center temperature decreases linearly for increasing spans.

	T center of maximum energy for different field spans (20 K span)				
Field	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
0 - 0.5 T	291	273.9	284.5	286.4	283
0 - 1 T	293	275.9	285.5	288.4	284
0 - 1.2 T	293	275.9	285.5	288.4	284
0.5 - 1 T	296	276.9	286.5	290.4	-

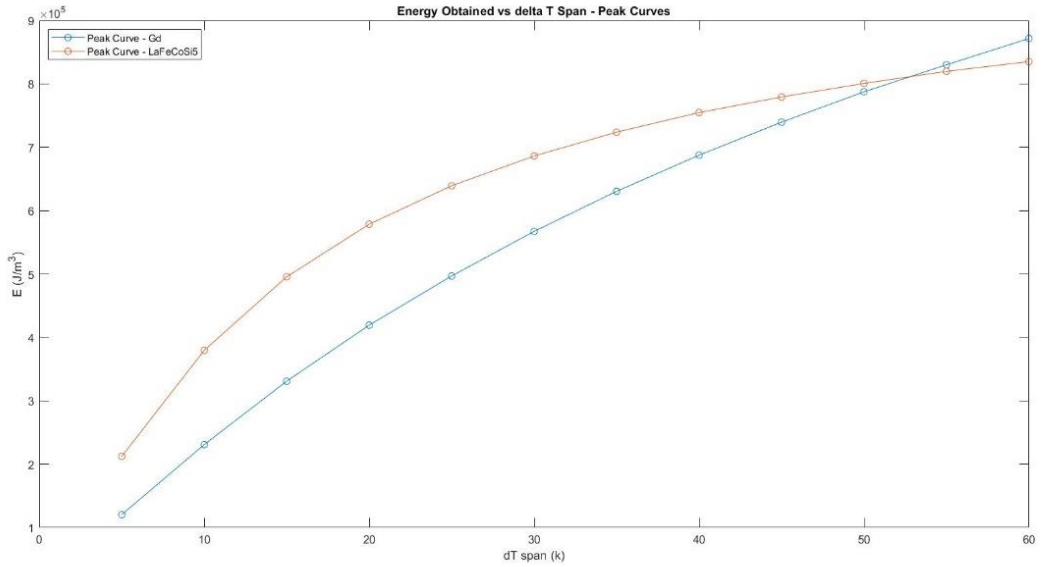
7 The optimal temperature changes with the change of field applied.

It was also seen that the energy harvest and the temperature span relate fairly linear under certain values, most noticeable for short and large spans as shown in figure



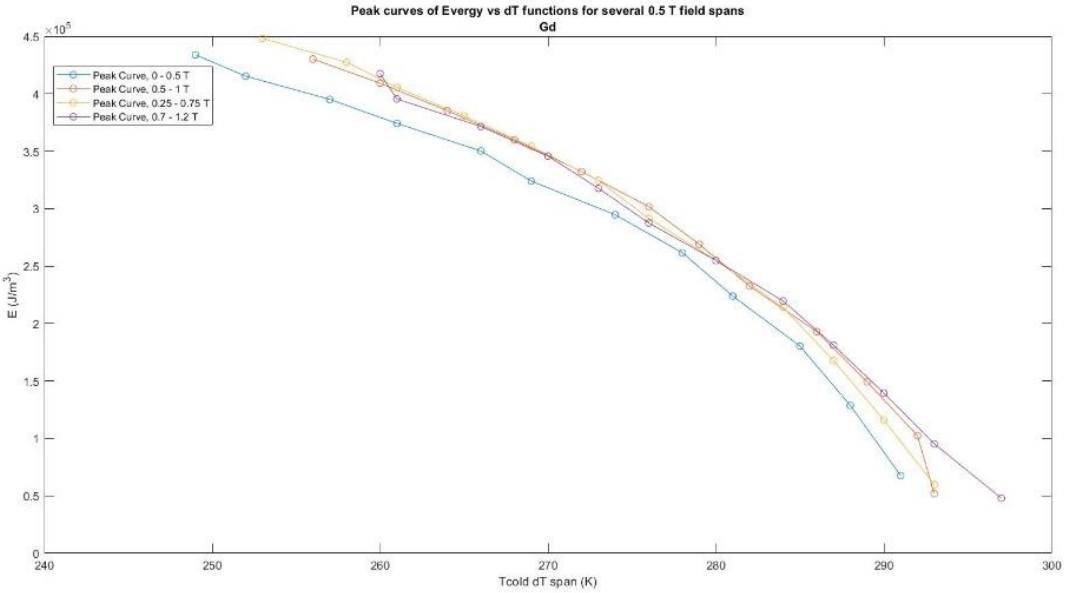
8 Energy output is linearly related for certain temperatures spans.

It was also seen that LaFeCoSi5 gives the highest energy outcome except for large temperature spans, when it is outperformed by Gd. The ranges at which each one is best depends on the field applied.



9 LaFeCoSi5 and Gd give the highest energy outcome, depending on the spans.

The study also revealed that the energy per volume output is fairly the same for a given range of field, regardless of where the range is located: for a given length of field spans the output is the same regardless on what values the span is set. This can be seen in the figure



10 Energy harvest is the same for a given field span no matter where the span is located.

Considering Curie's Law of Magnetization a formulation was presented for an ideal thermomagnetic cycle at which either the temperature or the field span is constant in phase:

$$E = \frac{c\,\mu}{3}(Ha^3 - Hb^3)\frac{-\Delta T}{Tc\,Th}$$

However, this formula is not always applicable, has an error associated for its approximation and needs experimental data to determine the parameters of the material if they are unknown; in this case computational processing of the data is preferable. Hence, it is not feasible to get a working formulation for energy harvesting of a thermomagnetic cycle.

CONCLUSSIONS

- The cycle of a thermomagnetic motor must be set at specific working temperatures to give the maximum performance. These optimal temperatures vary for the same span of temperature depending on the field applied and the material studied.

- There is no clear relation with the Curie temperature of the material and the optimal temperature.

- The impact of the temperature span length is not constant for all ranges of spans, and is more important for spans under 20 K.

- The highest energy output is achieved by LaFeCoSi5 material for short and intermediate field spans, and by Gd for large field spans.

- Field and temperature span can compensate each other so large energy harvest can be achieved even for small temperature spans if the field span is high enough.

- The energy per volume output is fairly the same for a given range of field, regardless of where the range is located. This makes construction of a working thermomagnetic cycle device easier to implement.

- The performance of the cycle in a Curie motor device is dependent on the field span, the temperature span and the specific heat capacity of the material. These variables are each related to each other in the thermomagnetic cycle, making it not feasible to get a formulation por energy harvesting without knowing all these variables.

INDEX

CHAPTER 1 – INTRODUCTION AND PROJECT APPROACH

In the industry and in any engineering application the majority of energy losses are in the form of heat. Thermal energy has been studied broadly when it comes to large energy production such as nuclear power or thermal energy production with coal, fossil fuel and others. However, many of these applications are not environmentally friendly and use liquids such as water or oil to work, or solely for cooling. The hot water or oil is often waste thrown away. In this cases, warm water may be thrown to rivers were they severely impact its environment. Though it is possible to use this waste hot water for regeneration in a Rankine cycle, for example, often the temperature of the wastewater is not enough for this usage. The result is that much thermal energy is being wasted with the water, without giving a use to that extra energy that was given to the water at a price, and that can be detrimental for environment. The same is true in many other types of cooling in industry.

Until now, not many applications have taken care about this: the use of water vapor at very high temperature has been well studied, but there are not any real applications for the usage of warm water at a relatively low temperature. This project aims to help in this by the study and optimization of a model device that is capable of exploiting this wasted thermal energy by the means of thermomagnetic energy harvesting.

As explained in *A Review on Low-Grade Thermal Energy Harvesting: Materials, Methods and Devices* by *Ravi Anant Kishore* and *Shashank Priya* [1], thermomagnetic energy harvesting relies on the effect of heat on magnetic properties. Thermomagnetic materials have good magnetic properties below a given temperature (Curie temperature) after which their magnetization decreases. Thermomagnetic devices use this effect to harvest energy using a cycle of high to low magnetization when heat is applied to the thermomagnetic materials.

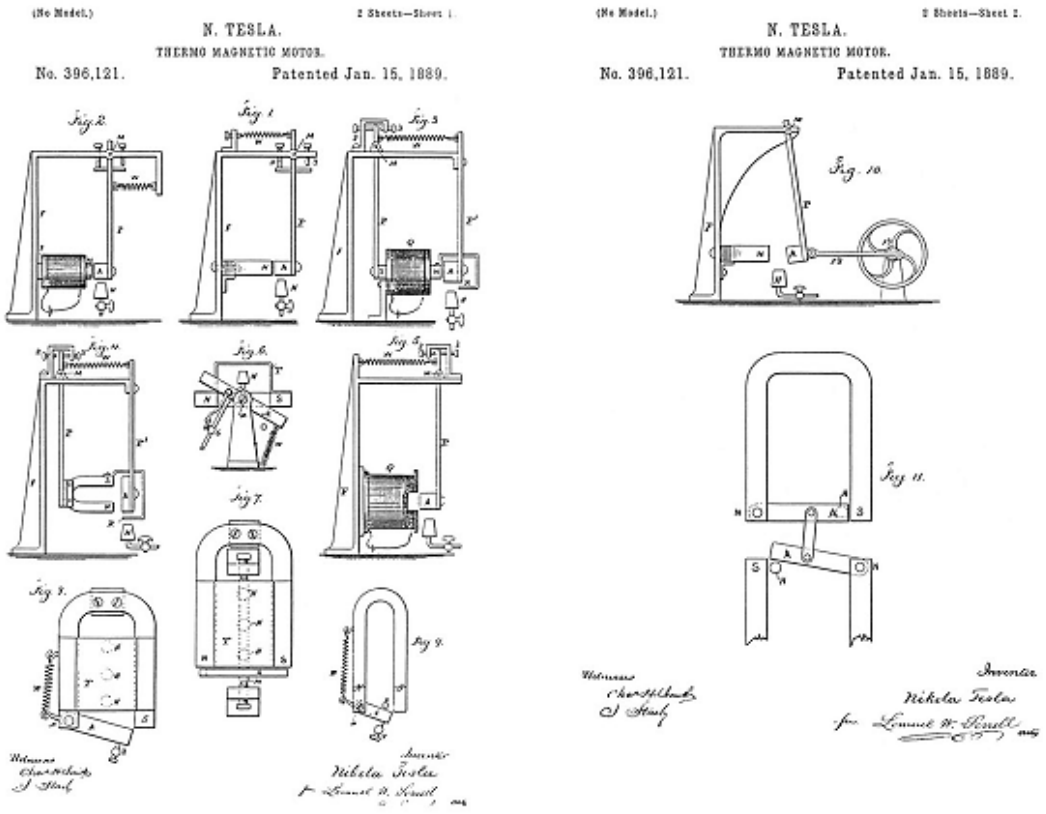
This project studies what are the best conditions for a thermomagnetic device to work. This involves the selection of material and parameters optimization such as temperature or field, so that the energy production from warm water is maximized. The motivation for this project is to provide an alterative source of renewable energy using the thermomagnetic properties of ferromagnets, and specifically to address the lack of alternatives for the thermal energy waste for temperatures around room temperature. If all the thermal energy wasted in industry could be exploited even to a minimum extent not only the industries would be more efficient and green, but it could have a very big impact on Earth's environment, helping to control climate change.

Likewise, not only higher global efficiencies would be achieved, diminishing electric waste, also CO2 emissions to the atmosphere would be greatly reduced; all within the use of a clean non-polluting technology. Furthermore, energy saving could be vital in certain applications where only a limited energy source is available (portable devices, submarines, satellites…).

CHAPTER 2 – DESCRIPTION OF TECHNOLOGIES AND STATE OF ART

2.1 Early thermomagnetic devices:

A thermomagnetic device is a machine that transforms thermal energy to useful work. The work can be obtained in the form of mechanical work or electrical energy. Thermal energy affects the magnetic properties of ferromagnetic materials: when they reach a transition temperature called the Curie temperature their magnetization disappears and the materials transform into paramagnetic state. The alternative phase change of the ferromagnetic across the Curie temperature is used by the thermomagnetic device for energy generation. The most prominent early designs where presented by Nikole Tesla and Thomas Edison. Tesla patented numerous designs in 1889 under the US Patent 396121.

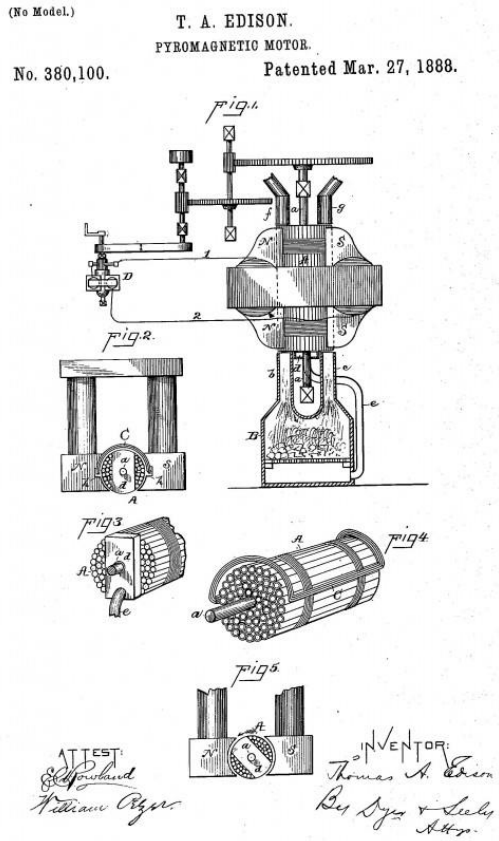


11 US Patent 396121 by Tesla

His thermomagnetic motor consisted of a permanent magnet or electromagnet generating the field with magnetic poles (N-S), an armature (A) that rotated about a hinge, a spring or weight force (W) and a Bunsen burner or heat source (H) [2].When

burned by the heat source, the armature would lose magnetization and thus magnetic attraction allowing the weight force to pull it away from the heat source. I this situation the armature will regain magnetization and the magnetic force will deliver it to its original position to start the cycle again. This resulted in a source of mechanical power that could be used in any manner.

Thomas Edison created a pyromagnetic motor in his US Patent 380100.


(No Model.)

T. A. EDISON.

PYROMAGNETIC MOTOR.

No. 380,100. Patented Mar. 27, 1888.

Fig. 1.

Fig. 2.

Fig. 3.

Fig. 4.

Fig. 5.

ATTEST:

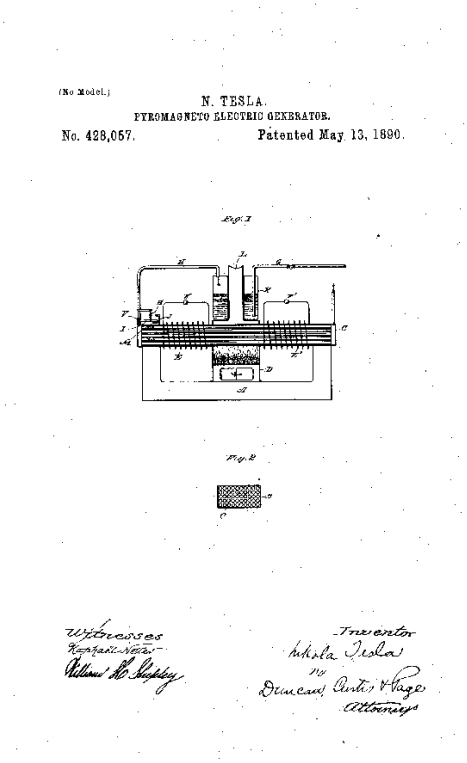
INVENTOR:

12 US Patent 380100 by Thomas Edison. Image from [8].

This device consisted on two magnetic poles from a permanent magnet or an electromagnet, an armature made with iron sheets interstices longitudinal tubes. With the use of a shaft the armature was mounted vertically and the tubes where heated by a furnace located beneath the armature, with two outlets covering the lower ends of the tubes. Hot air was blown form the furnace to some of the tubes, while cool air was blown in the others. The resulting temperature difference in the armature meant that its magnetization would also vary, which would give a force imbalance. This imbalance

will make the armature rotate. The motion would continue as the tubes where alternatively heated and cooled while rotating.

In 1890, Tesla invented a pyromagnetic-electric generator:


(No Model.)

N. TESLA.

PYROMAGNETO ELECTRIC GENERATOR.

No. 428,057. Patented May 13, 1890.

Fig. 1

Fig. 2

Witnesses

Inventor

Nikola Tesla

by

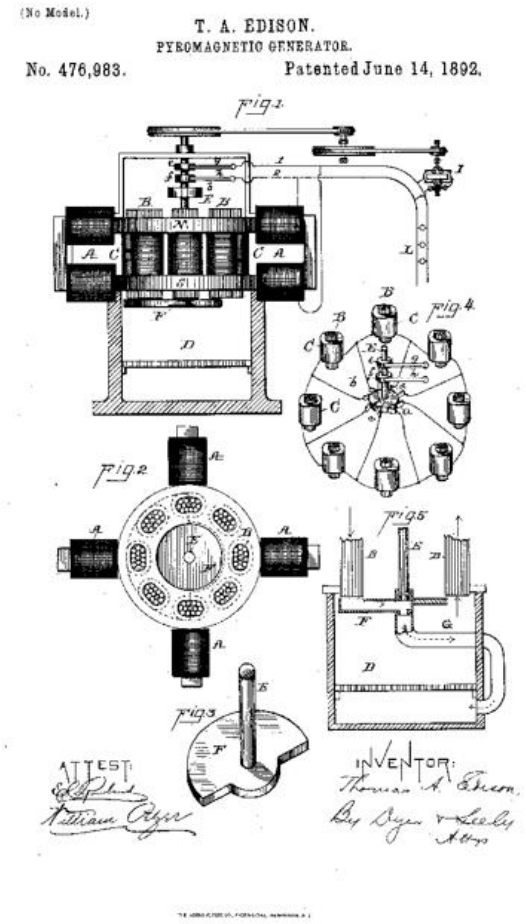
Duncan, Curtis & Page

Attorneys

13 Pyromagnetic-electric generator by Tesla (1890). Image from [2].

The device was designed to produce electric current without mechanical power creation. A C-shaped permanent magnet created a field in which the armature stayed. Around the armature two wires were coiled. The armature will be heated by a furnace and cooled by steam from a boiler. The alternation of heating and cooling the armature would change its magnetization, varying the flux seen by the coils, and thus inducing current in them.

In 1892, Edison also patented another type of pyromagnetic generator:



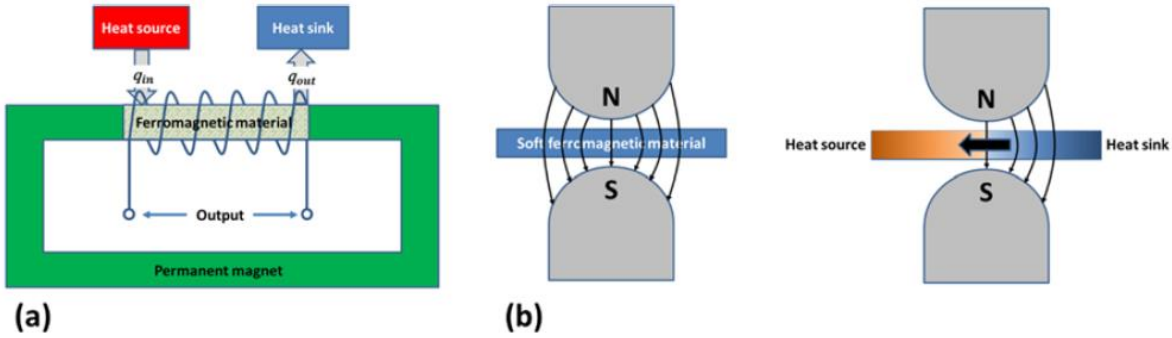
14 Pyromagnetic generator by Edison (1892).

Two iron rigs were put circumferentially containing thin iron bundles. These bundles were coiled with windings, a furnace layed beneath the construction and a rotating shield was placed between the furnace and the bundles, preventing the heat from entering the iron tubes contain in each bundle. This way, the open tubes would receive hot air from the furnace, while tubes behind the shield would cool down. The rotation of the shield made it so that the temperature around the bundle tubes was changing periodically. This changed magnetization in the tubes, generating current in the windings.

As stated in [1], these early thermomagnetic devices were had very complex designs and used iron as working ferromagnetic principle. Iron demands a high temperature around 700ºC to demagnetize, temperature that can harm the permanent magnet, causing thermal degradation in it. Also, in order to reach 700ºC high volumes of fuel will need to be used, questioning the economic feasibility of such devices. On top of this, thermodynamic efficiency of heat engines is proportional to $\Delta T/T$, where ΔT is usually small for ferromagnetic to paramagnetic transition, while the working temperature T is very high, giving very small efficiencies. It is therefore fair to say that the lack of good thermomagnetic materials with more desirable properties has slowed the development of thermomagnetic devices during history.

2.2 Working Principle:

Generating electricity with thermomagnetic materials can be achieved using two methods: with active thermomagnetic devices (thermomagnetic generators) or with passive thermomagnetic devices (thermomagnetic motors). Active thermomagnetic devices convert thermal energy to electrical energy directly. Passive devices go through and intermediate mechanical stage to obtain electrical energy. Though the original Tesla devices centered on passive devices, active converters have traditionally been given more attention in literature.



15 Figure from [1] showing the working mechanism of active (a) and passive (b) thermomagnetic energy conversion devices.

Active thermomagnetic generators (Figure 5a) include a permanent magnet, typically C-shaped, with a thermomagnetic shunt between poles. Around the shunt a winding is wrapped. The shunt is then heated and cooled alternatively. When heated, the magnetic dipole moments disorder and the shunt losses magnetization reducing the magnetic flux through the winding. When cooled, dipoles reorder, magnetization rises and the flux seen by the winding increases. This alternative variation of flux induces a voltage difference in the winding according to Faraday's law, that will give power when connected to an external resistor. Thus, power is generated.

Passive thermomagnetic devices, also known as thermomagnetic motors or Curie motors consist of a magnetic circuit and a movable armature of soft ferromagnetic material [1]. When all the armature is at the same temperature, the resulting magnetic forces from the extreme sides are equal and opposite, thus cancelling each other and the armature remains motionless. If one side is then heated above the Curie temperature, it will lose magnetization, weakening the magnetic force induced in this side and therefore creating an imbalance in forces that will move the armature. This way motion is created. A restoring mechanism is needed to make the armature go back to its original position.

Curie motors are then converting thermal energy into mechanical energy and thus need an electromechanical generator to produce electricity out of the mechanical energy.

2.3 Thermodynamics of thermomagnetic materials:

A thermomagnetic material, like Gadolinium, is magnetized when all magnetic spins from its dipole moments are ordered and oriented in the same direction, summing up to a resulting magnetic moment in this direction which is responsible for the material magnetization, just like in a regular magnet. In this situation the thermomagnetic material behaves as an actual magnet.

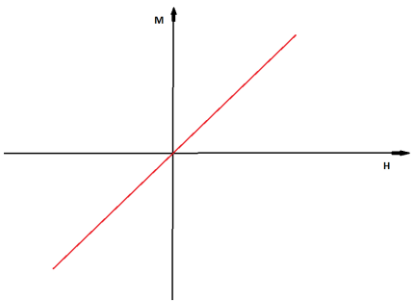
However, if the temperature of the material is increased, the extra energy from heat will make the molecules shake, gaining kinetic energy, and disordering the dipole moments. With the disordering of dipole moments comes the decrease in magnetization and the loss of magnetic energy. This applies to regular magnets and ferromagnetic materials. During this process, a ferromagnetic material may go through a phase change near the transition temperature. Across this transition temperature, also known as Curie temperature or Curie point, magnetization disappears and the material transforms into paramagnetic state [1]. Note that this Curie temperature is different form the ferroelectric Curie temperature.

In a similar way, if the material is cooled the magnetic dipole moments will order again and the material can fully magnetize once more. Using this magnetization variation along with a permanent field from a magnet, a magnetic cycle can be produced for a Curie motor to obtain energy.

About hysteresis:

When ferromagnetic materials are exposed to external field, they magnetized in accordance with the field applied. Inside the material, the magnetic dipoles have each a random orientation, making a chaotic bunch of dipoles that have total magnetic moment equal to zero. This is because on average, the vector sum of all magnetic dipole spins is zero, as they don´t have a preferred direction. When the material is subject to an external field, the spins order in the direction of the field. This way the spins contribute to sum field in the same direction and magnetization takes place. The higher the external field applied, the more spins are order and the higher the magnetization will be.

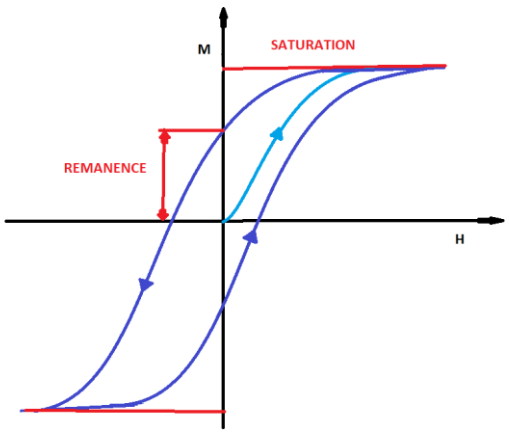
For an ideal endless ferromagnetic material, the magnetizations curve would be linear: an amount in external field applied H would result in a proportional increase in magnetization M. This proportion or ratio, the slope of the curve, would be constant for any range of values, thus resulting in a line function. This can be a very rough approximation of reality for small field values.



16 Ideal Magnetization for an ideal endless ferromagnetic material.

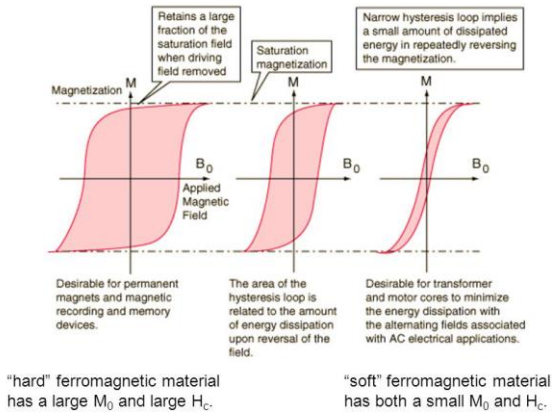
In real materials, as the external field is increased there comes a point where the magnetic material saturates: the gain in magnetization diminishes and finally there will be no more gain, setting the magnetization at a maximum value no matter how much more field is applied. Saturation occurs because as the applied field grows more and more magnetic spins are ordered inside the material until all of them but a few have been ordered. At this point, any excess of applied field will not see an increase of magnetization, as there are no more chaotic magnetic spins to order.

The same applies to an external field with opposite direction, where the magnetization reaches another limit. This gives the magnetization curve an "S" shape enclosed within the maximum and minimum magnetization. There is no limit in the external field that can be applied. However, some ferromagnetic materials have a "magnetic memory" after an external field is applied, so when the field is removed the material will still be magnetized to a certain degree. This is called remanence of the ferromagnetic and implies that a negative opposing field should be applied in order to get no magnetization. If the negative field is increased to the lower saturation limit, after it is removed there would be a remanence of negative magnetization that will require an extra positive field to get the magnetization to zero. This is called the hysteresis loop.



17 Hysteresis loop.

Hysteresis can be a problem as it results in energy losses. These losses are represented by the area inside the hysteresis loop in the magnetization graph. Soft ferromagnetic materials present a thin hysteresis loop and have then lower energy losses. This makes them suitable for use in motors and transformer cores. Hard ferromagnetic materials present a wide loop, suffering heavier energy losses but can retain a larger fraction of saturated field. Thus, they are desirable for permanent magnet and magnetic memory applications.

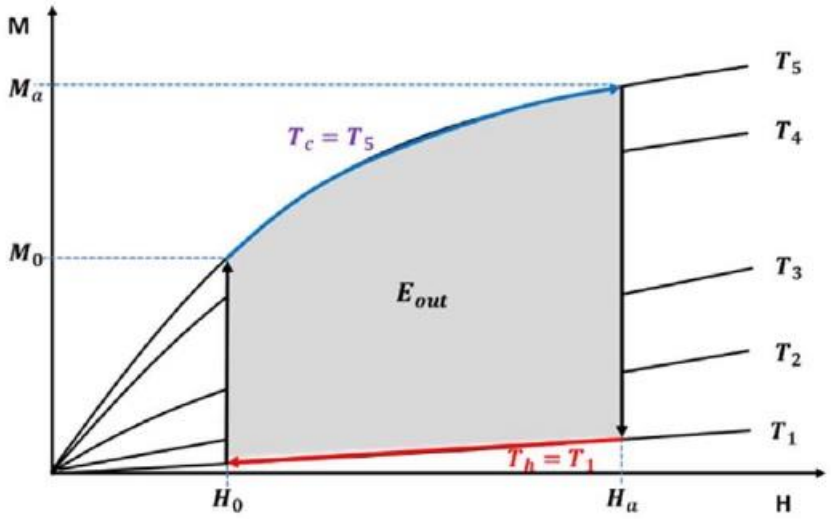


18 Hysteresis loop for different ferromagnetic materials. From [9].

In this project, the materials studied are soft ferromagnetic materials, as such, the hysteresis effect is minimum and the subsequent losses can be dismissed.

The thermomagnetic cycle:

The thermomagnetic experiments a change in magnetization according to its energy cycle: an increase in thermal energy results in the demagnetization of the material, upon cooling it magnetizes again and the cycle can start once more. In the study of the cycle it is interesting to consider the energy involved, thermal and magnetic. The chart below reflects both temperature and magnetization vs field:



19 Thermomagnetic cycle for a field span from H_0 to H_a and temperature span from T_c to T_h. The energy obtained from the cycle is the area inside the curves times the vacuum permeability constant μ_0.

The figure was extracted from the article *A review on design and performance of thermomagnetic devices* by Ravi Kishore and S.Jeba Priya [3]. T1 = Th is the hot temperature and T5 = Tc is the cold temperature at which the device would operate. The figure represents and ideal cycle where the shaded area between the cycle curves represents the maximum achievable energy, i.e. the maximum work potential of the generator. This would be calculated as:

wmax = Eout = μ0 ∮ HdM

As described in [3], this formula comes from the closed integration of the power per unit volume of ferromagnetic:

$$v = nA\frac{dB}{dt} \qquad i = \frac{HL}{n} \qquad B(H,T) = \mu_0[H+M(H,T)]$$

$$\Rightarrow \qquad P = \frac{vi}{AL} = H\frac{dB}{dt}$$

$$\Rightarrow \qquad E = \int P dt = \int H\,dB$$

Using the filed expression posted above, the energy equation can be rewritten as:

$$E = \mu_0 \int H\,dH + \mu_0 \int H\,dM$$

Which, for a cyclic process $\oint H\,dH = 0$, is:

$$E = \mu_0 \oint H\,dM$$

The maximum performance of the device can then be obtained as:

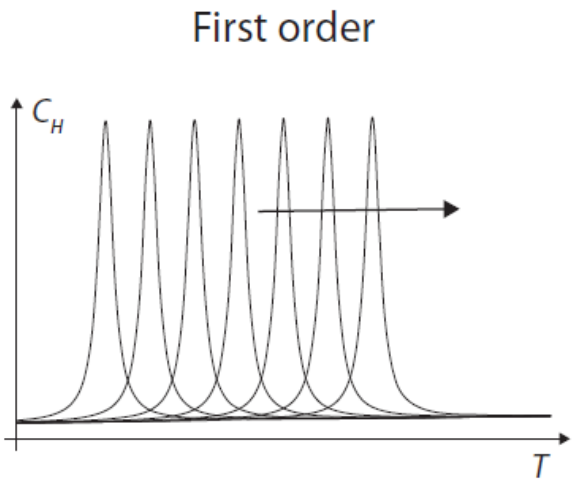
$$\eta_{max} = \frac{w_{max}}{q_{in}} = \frac{\mu_0 \oint H\,dM}{q_{in}}$$

Qin (heat input) can be obtained from the first law of thermodynamics: $\Delta U = Q - W$. The heat input is then $Q = \Delta U + W$. According to the *A review on design and performance of thermomagnetic devices* article [3], the heat input is then obtained by the internal change of energy associated with the change in temperature and the change in magnetic entropy:

$$q_{in} = \rho \int_{T_c}^{T_h} C_p(T)dT + T \int dS_m$$

In this way, the article assumes that the heat input is given only by the internal energy change, while the work is inexistent. Since the equation refers to heat input it is fair to say that the work considered is the input work – work done *on* the system, not *by* the system, and therefore is zero.

On another note, while in this formulation the specific heat is written as a function of the temperature only, in the article *Materials Challenges for High Performance Magnetocaloric Refrigeration Devices* [4] it is stated that it is also a function of the applied field H, as shown in the graphic for first order materials:



20 Specific heat capacity for first order materials.

The arrow indicates the way of increasing field. However, the field can also be understood as a function of temperature in thermomagnetic materials. For this reason it is often stated that the heat capacity of the materials considered C (T,H) can be understand as a function of temperature only C (T). In the study of this project the material datasets are both H and T dependent, so both parameters will be taken into account.

If the magnetic field applied is small the first term of the heat equation is much larger than the second term, hence the later can be ignored:

$$\eta(or\,\eta_{abs}) \cong \frac{\mu_0 \oint H\,dM}{\rho \int_{T_c}^{T_h} C_p(T)\,dT}$$

The relative efficiency with respect to Carnot will then be:

$$\eta_{rel} = \frac{\eta_{abs}}{\eta_{carnot}} = \frac{\eta_{abs}}{1-\frac{T_c}{T_h}}$$

These expressions can be used to evaluate the performance and energy obtained from thermomagnetic devices according to the material parameters.

In *A review on design and performance of thermomagnetic devices* [3] presents a cycle figure showing both an ideal and not ideal cycle, as well as Solomon thermomagnetic generator, an active thermomagnetic device. Solomon made an interesting model suggesting that the efficiency can be improved if the applied magnetic field is also cycled as is the temperature. The model presented in this project works with permanent magnets but the field could also be cycle due to the rotatory movement of the generator, increasing efficiency.

It is worth noting that the heat source from which the energy is extracted in this study is supposed to be warm wastewater from an industry. This means that no expenses are considered for heating the water: the heat income is therefore free, making efficiency not so relevant, and economic studies are not considered.

CHAPTER 3 – DESCRIPTION OF THE MODEL DEVELOPPED

3.1 Objectives and Specifications

The ultimate goal of this project is to model and optimize a device capable of using thermal energy in water to generate energy at a profitable pace with the use of a thermomagnetic generator. This would permit the implementation of a prototype designed based on the results obtained. As such, the objectives of the project were first set as:

- Specify the requirements for thermomagnetic material and the appropriate geometry for energy harvesting.
- Design possible implementation models.
- Prototype the model using 3D printing technology.
- Operate and test the device.

However, it was always clear that no other simple thermomagnetic prototype should be constructed, as it has already been. Instead, in the case of constructing a prototype the cycles had to be very well studied and the most desirable characteristics of thermomagnetic materials should be first found out so that its implementation would suppose a true advance in the state of the art, presenting a better device that would justify the prototype.

This way, the project focuses more on the modeling and study of the system, the optimization of parameters and material selection. As such, the nature of the project is different as it was first considered, being more of an investigation project that will ease the construction of a future device rather than a prototype project. The objectives achieved in this project are:

- To give a broad view of how the different thermomagnetic materials behave when the cycle is subject to a variable field span.
- To give a broad view of how the different thermomagnetic materials behave when the cycle is subject to a variable temperature span.
- To highlight the difference between material performances under different conditions.
- To select the best fitting material when considering certain variables.
- To optimize variables such as temperature to give the best performance of each material.

3.2 Data

In this study five different materials will be considered: Gd (Gadolinium) and different samples of LaFeCoSi5, LaFeCoSi11, LaFeCoSi15 and LaCaSrMnO. Gd will be used as an example when all material behaviors are similar, and differences will be highlighted when encountered. The general example is then for a sample of Gd at a field span of 0 – 1 T and a temperature span from 280 – 300 K. This ranges were selected because they are of much interest due to the easy implementation and because they are normal working values. The temperature span is also typical in range and is set near room temperature.

Gd is one of the most commonly used magnetocaloric materials (the magnetocaloric effect refers to the temperature change of a magnetic material when exposed to an external changing magnetic field). It has a Curie temperature near room temperature, is easy to acquire and has a good magnetocaloric performance. It is also easier to shape than the other materials considered. Gd however corrodes easily and is quite expensive. The commercial grade gadolinium considered in this study contains 99.5% of rare earth metal, 99.94% of which is gadolinium. The Gd sample has a density of 7900 kg/m^3 [5].

LaFeCoSi5, LaFeCoSi11 and LaFeCoSi15 all refer to different samples of $LaFe_{13-x-y}Co_xSi_y$. This material presents a significant adiabatic temperature change and allows its Curie temperature to be adjusted, explaining the study of three different samples. The first sample has a content of $x = 0.86$ and $y = 1.08$. The second sample is $x = 0.94$ and $y = 1.01$. The third is $x = 0.97$ and $y = 1.07$. The density of each sample is 6980, 7290 and 7160 kg/m^3 respectively [5].

LaCaSrMnO refers to $La_{0.67}Ca_{0.33-x}Sr_xMnO_3$, parameter x can range from 0 to 0.33 [6].

Data was provided in the form of tables of dataset for each material, where magnetization or specific heat can be found for each value of field and temperature. This tables were used by the codes to work on MATLAB. Special care was taken to convert units when needed (degrees Celsius to kelvin, teslas to amperes per meter…).

The datasets cover field values up to 1.5 T, but some data is not present for certain temperatures at high fields. In order to have a working code and avoid missing results and rare errors, the maximum field studied was 1.2 T, highest field value without any missing data. Also, the dataset for LaCaSrMnO is very limited when compared to the other material, so results at high fields or at certain temperature ranges were not possible.

3.3 Algorithms

Many codes were developed, some of them with various variations created as the study grew broader. The codes can be found in APPENDIX D. Here, a view of the main codes is provided to give a general understanding of what they do.

- The *E_vs_H* codes: these codes were the first implemented in MATLAB and are rather basic. The code calculates the energy of a cycle between two field values for a given temperature span, according to the formula $E = \mu \oint H \, dM$. This is done by calculating the integral as the difference between the two areas under the field curves and multiplying times μ constant. The code was created to calculate the integral step by step. One of these codes exists for all materials but LaCaSrMnO (which was considered afterwards).

- *Harvest_integration*_material*_E_vs_H* codes ('material' being Gd, LaFeCoSi5, LaFeCoSi11, LaFeCoSi15 or LaCaSrMnO): these codes have the same objective as the previous one calculating energy output (always per volume) as a function of external field applied, but have a better implementation that converts the datasets to an homogeneous equally spaced table, giving more consistent results and making use of the functions *trapz()* and *cumtrapz()* to calculate the areas (energy) automatically. *V2* version of codes provide some improvements when considering Curie temperature and improve plots.

- *Harvest_integration*_materia*l_E_vs_dT* codes: coding similar to the previous one, provides the energy output as a function of a growing temperature span; the temperature span is centered on the Curie temperature.
  Harvest_ALL_E_vs_dT_V2all_together gives a better implementation, allowing field selection and to plot all materials at the same time.

- *Harvest_integration*_material*_E_vs_20dT* codes: similar to the last codes, provides the energy output for a fix span of 20 K at different temperatures. This codes allow to find the optimal temperatures at which the span should work.

- *Harvest_integration_ALL_E_vs_20dT* code: an evolution of the previous, this enables the plotting of all material at the same time, as well as allowing variable temperature spans. Harvest_integration_ALL_E_vs_20dT_V2 is the second version, presenting an automatic field searcher, that converts field and permits to plot for different field values.

- *Harvest_integration_ALL_peaks* code: gives the energy output as a function of temperature span and cold temperature of the span (locating the span). Joins the maximum to form a peak curve and highlight tendencies.

- *Harvest_integration_ALL_field_varied_peak_curves* code: this code can give the peak curves containing the optimums for any field range for any material for

every temperature span. The peak curves are represented as energy output vs the cold temperature of the temperature span.

- *Harvest_integration_ALL_Tcenter_vd_Tspan_all_fields* code: gives the center temperature as a function of the span, for each material at a desire field span.

- *Harvest_integration_ALL_fields_E_vs_dT* code: provides the maximum energy as a function of temperature span for all materials at a desired field span.

- *Harvest_Cp_mean_method_ALL* code: gives the heat input vs temperature of the water for desired field span and temperature span.

3.4 Numeric Implementation

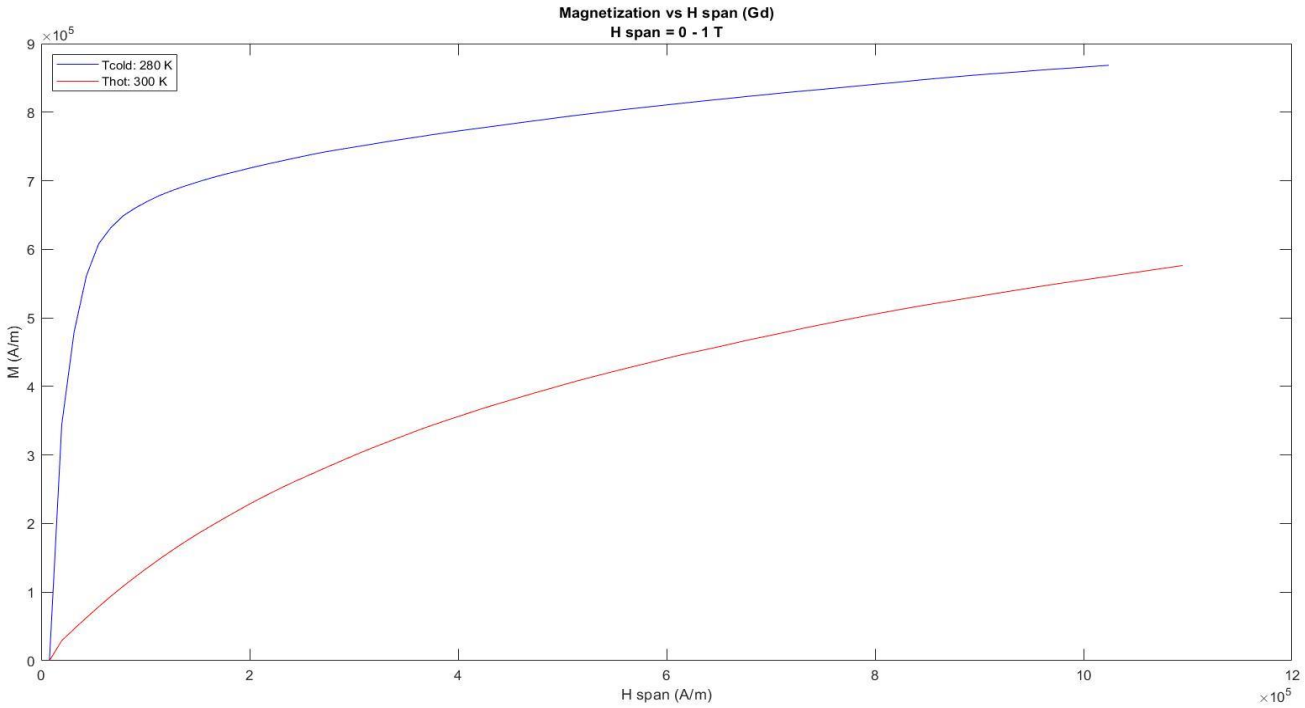
The numerical implementation of the study was first started as handmade calculus. As the operations became broader and complex, Excel was used, which also gave the opportunity of plotting graphs. Excel was then replaced by MATLAB, which offered more powerful tools for plotting and calculating under various conditions for changing parameters. MATLAB prove to be a big challenge as it was an almost unknown software but ended up providing the best results.

CHAPTER 4 – ANALISIS OF RESULTS

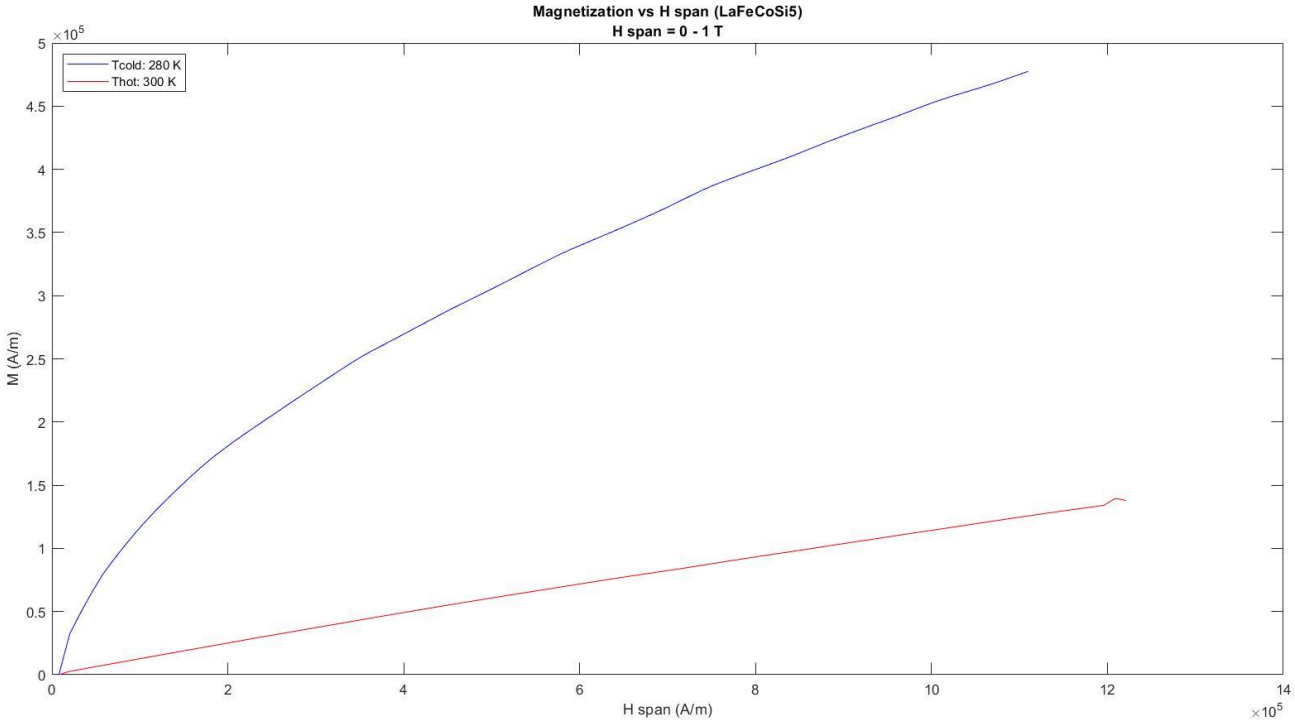
Results – set parameters:

The scripts were ran to evaluate a study cycle of 20 K temperature span ranging from 280 K to 300 K, for a field span from 0 to 1 T. This same case of study is applied to all five thermomagnetic materials considered. The goal of this is to obtain a general view of how each material behaves under the same circumstances. Then, the various parameters will be changed to give a broader view of their behavior which will be of use when choosing the best fitting material for the application desired.

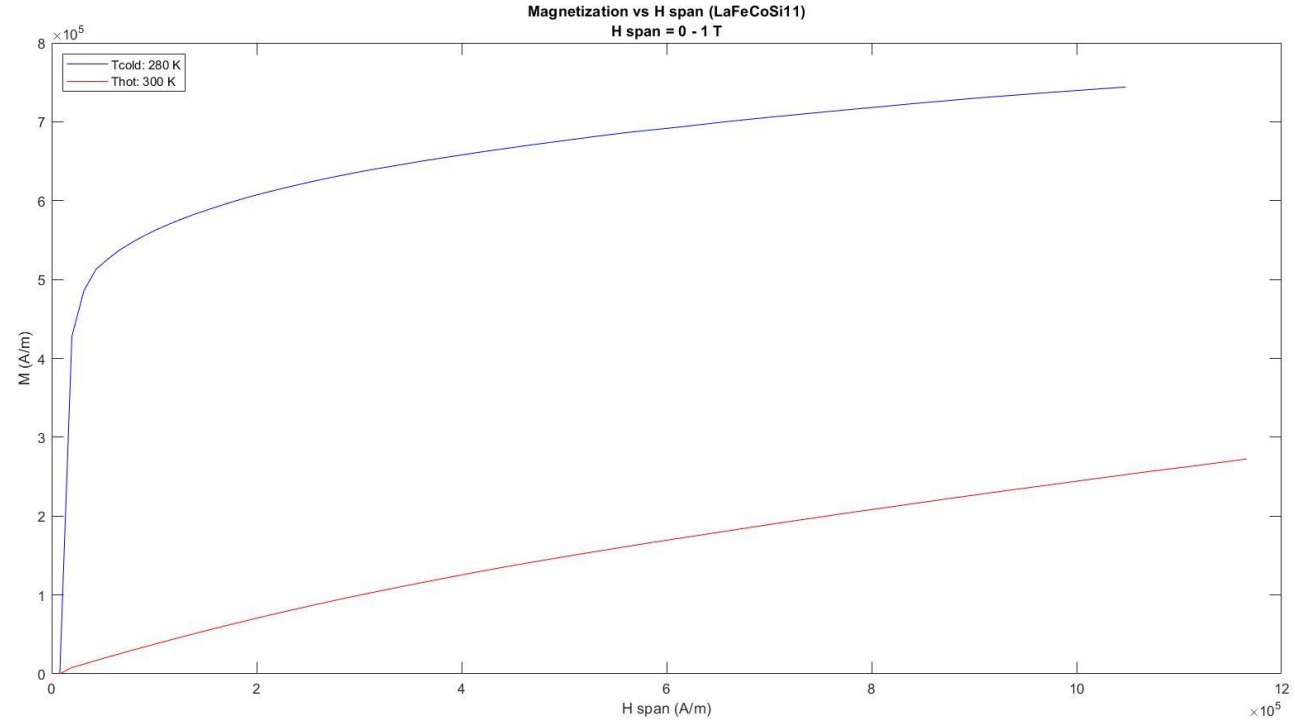
First, the *Harvest_integration_Gd_E_vs_H.mat* and their respective script for other materials were ran being the temperature span 20 K and the field ranging from 0 to 1 T. The cycles showing the magnetization curves and the energy obtained according to the applied field span can be seen in the charts below:



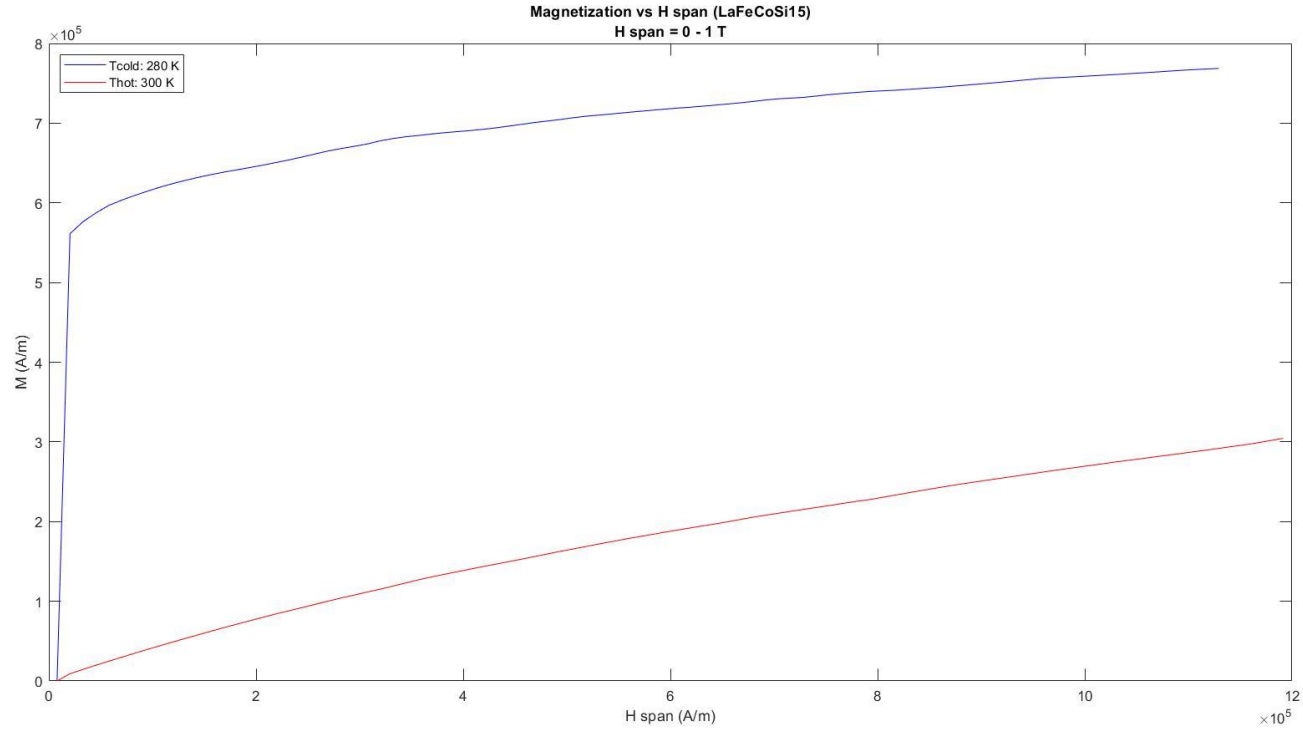
21 Thermomagnetic cycle of Gd. Field span from 0 - 1 T. Temperature span from 280 K (blue) - 300 K (red).



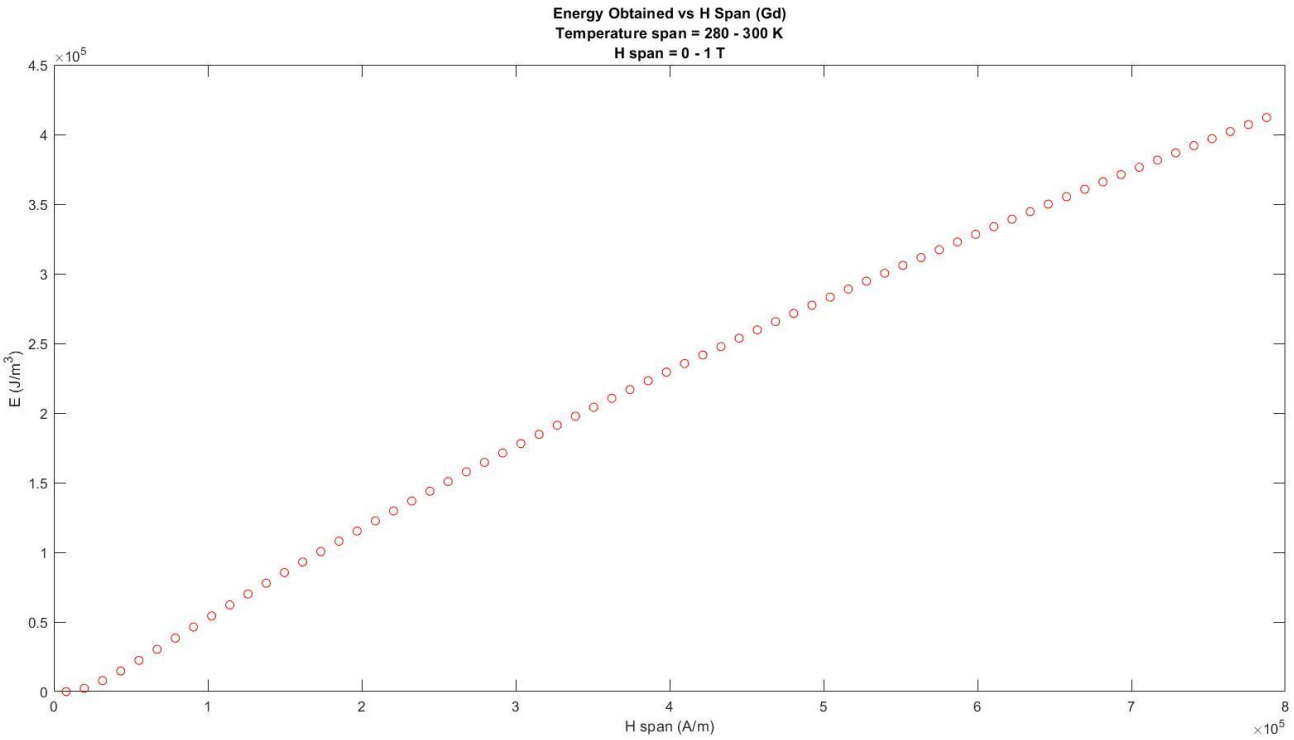
22 Thermomagnetic cycle of LaFeCoSi5. Field span from 0 - 1 T. Temperature span from 280 K (blue) - 300 K (red).



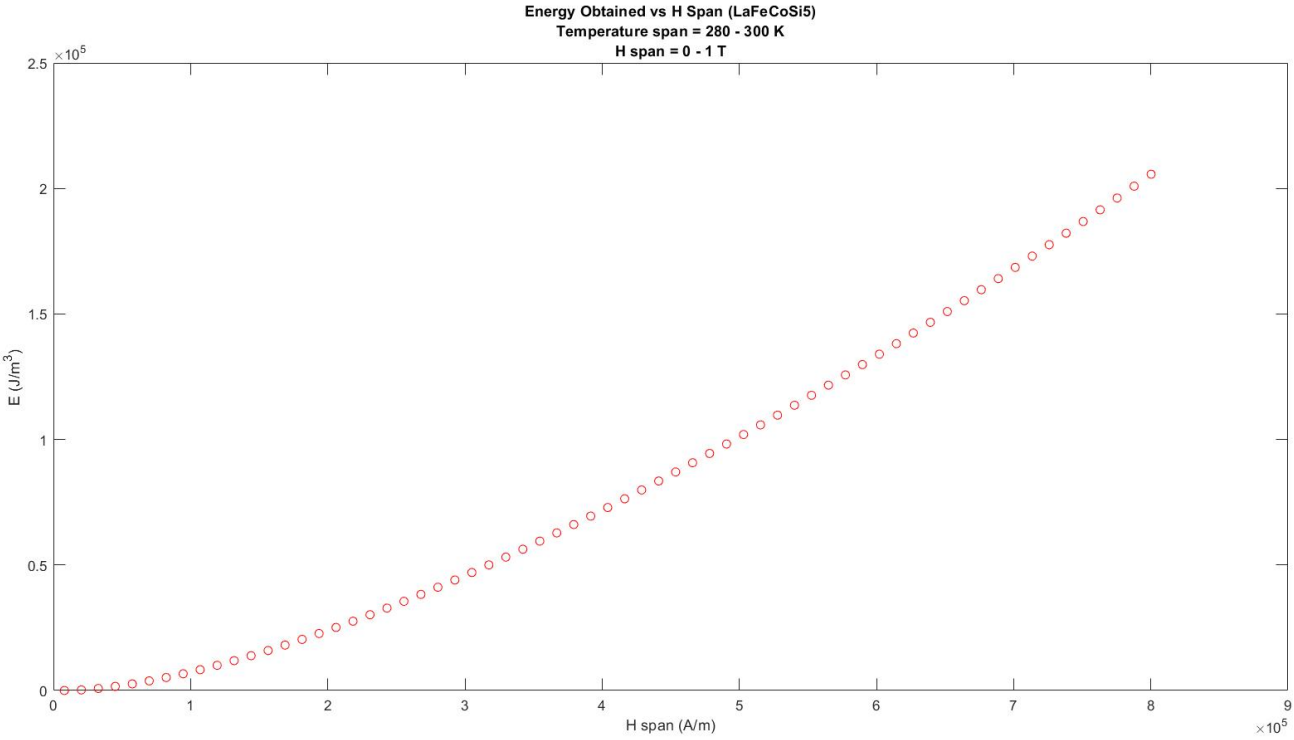
23 Thermomagnetic cycle of LaFeCoSi11. Field span from 0 - 1 T. Temperature span from 280 K (blue) - 300 K (red).



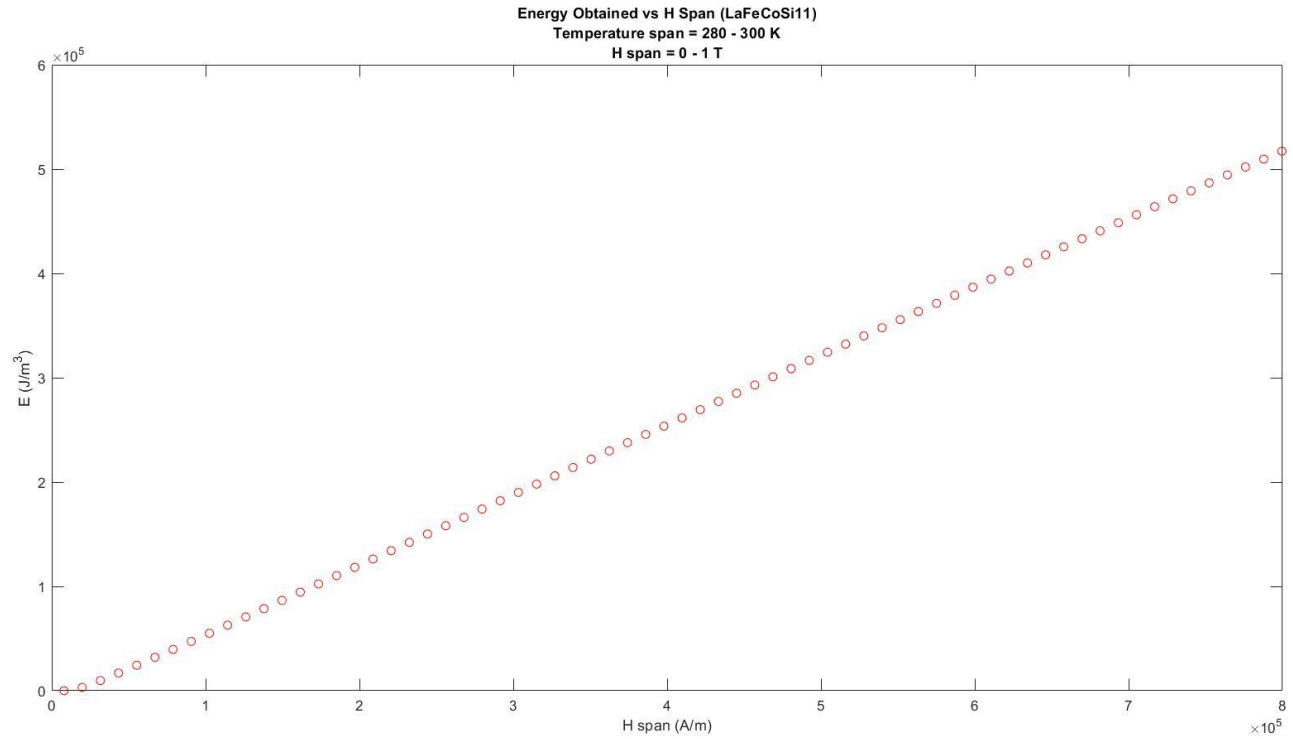
25 Thermomagnetic cycle of LaFeCoSi15. Field span from 0 - 1 T. Temperature span from 280 K (blue) - 300 K (red).



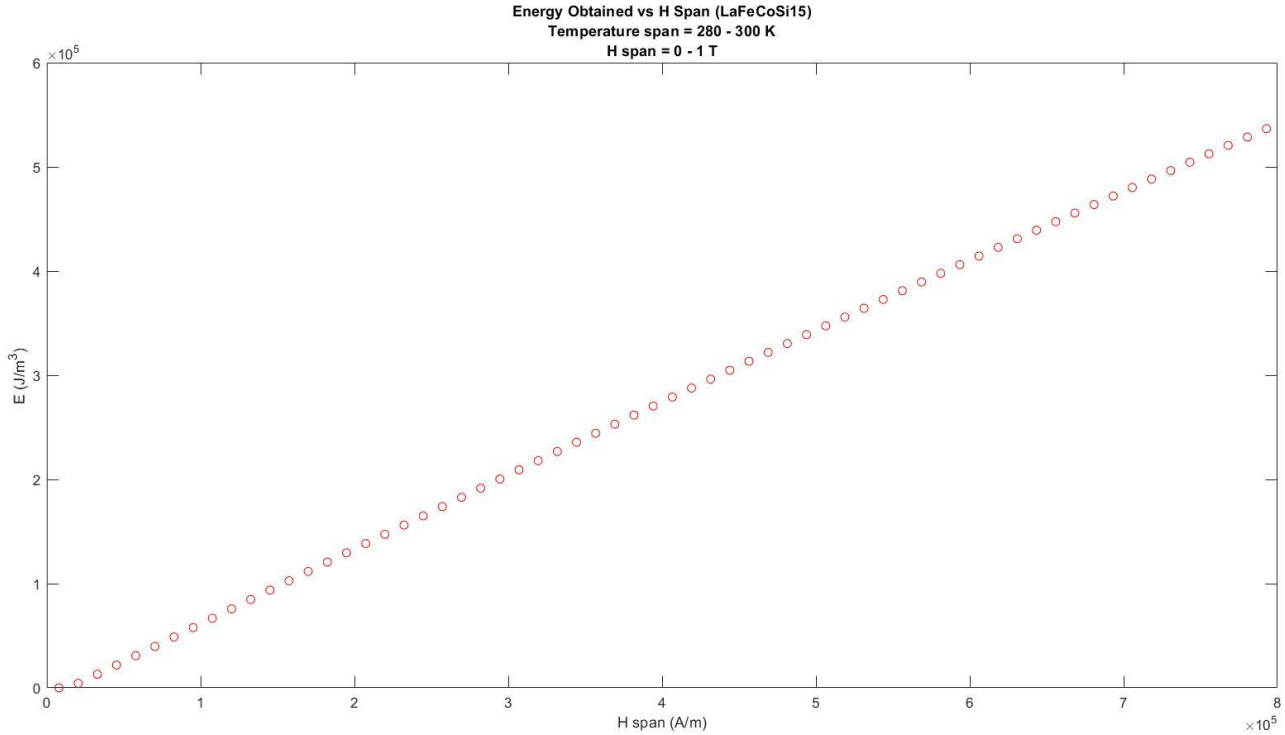
24 Energy per volume output for Gd. Field span from 0 - 1 T. Temperature span from 280 K - 300 K.



26 Energy per volume output for LaFeCoSi5. Field span from 0 - 1 T. Temperature span from 280 K - 300 K.

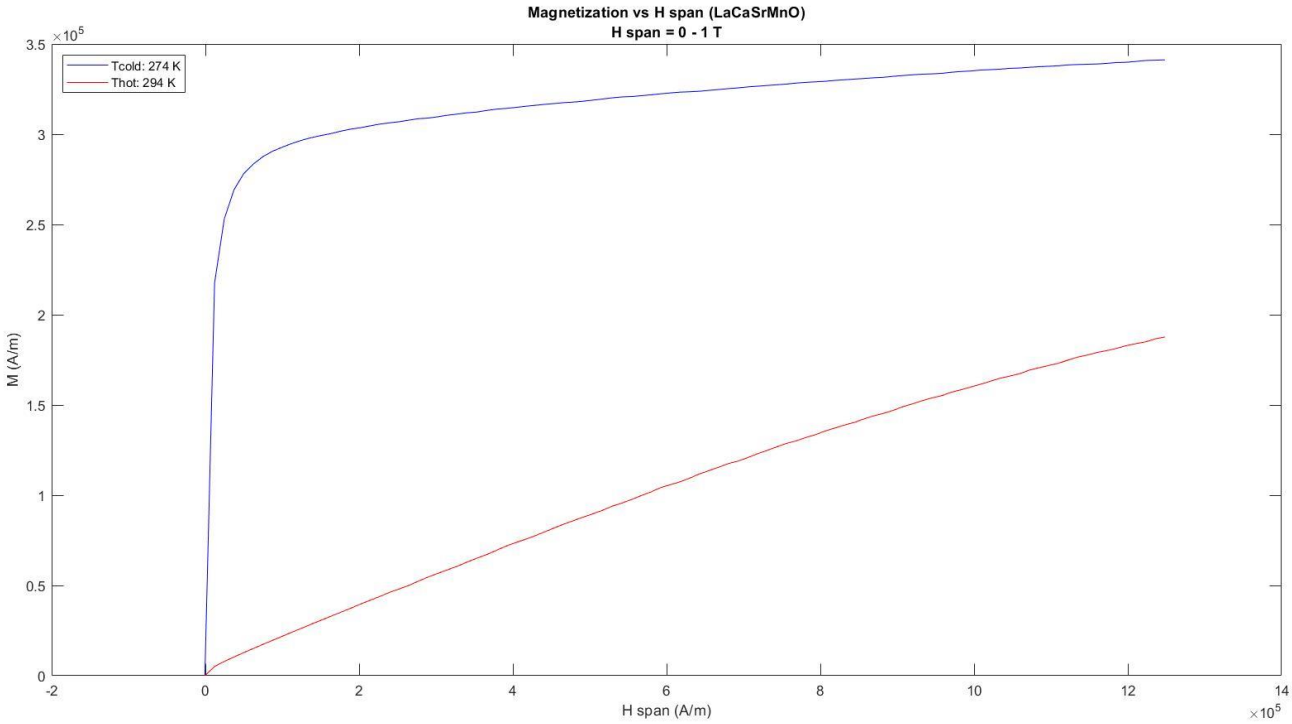


27 Energy per volume output for LaFeCoSi11. Field span from 0 - 1 T. Temperature span from 280 K - 300 K.

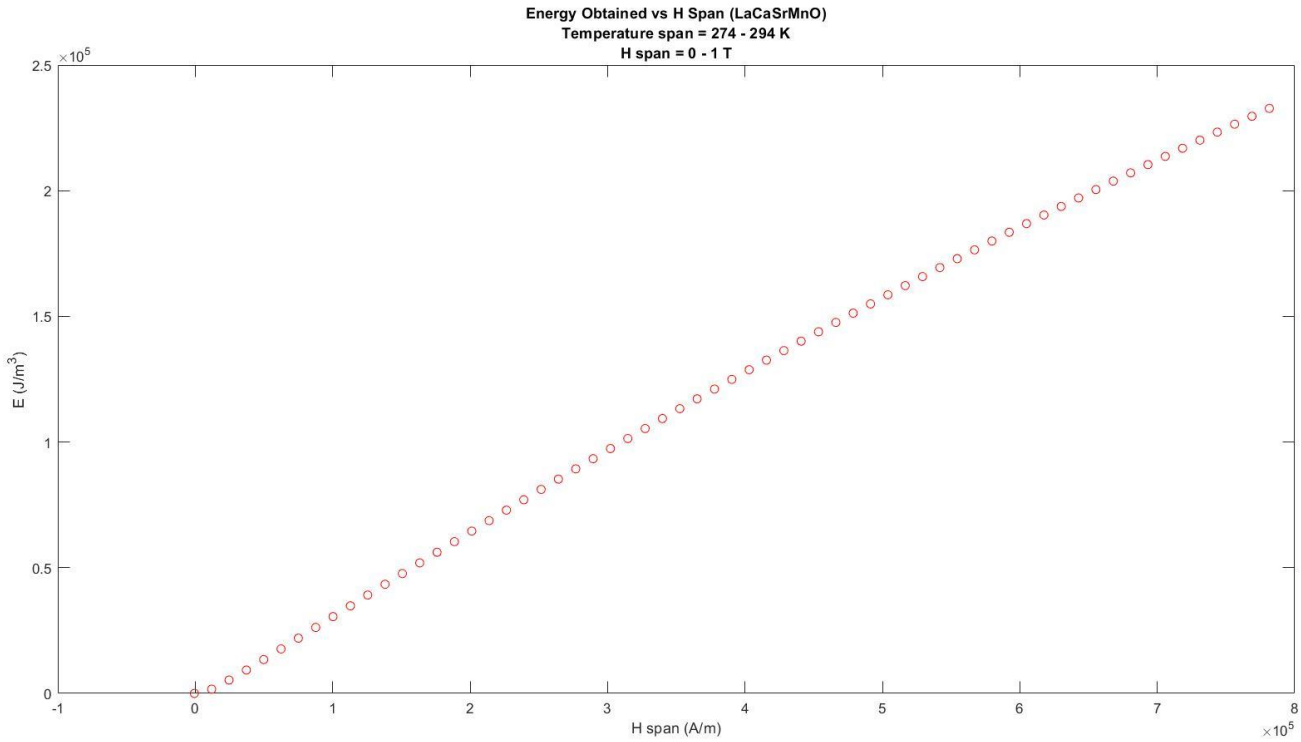


29 Energy per volume output for LaFeCoSi15. Field span from 0 - 1 T. Temperature span from 280 K - 300 K.

LaCaSrMnO is not included in these charts due to datasets limitation, which is available for limited temperatures and fields. The closest available temperature span (274 – 294 K) is presented for comparison with the other materials:



28 Thermomagnetic cycle of LaFeCoSi11. Field span from 0 - 1 T. Temperature span from 280 K (blue) - 300 K (red).



30 Energy per volume output for LaCaSrMnO. Field span from 0 - 1 T. Temperature span from 274 K - 294 K.

The energy output can be seen in the table on *figure 21*. It is noticeable how under the same range and parameters the output can be so different. For example, Gd output is under the same conditions around two times greater than LaFeCoSi5. Gd has a Curie temperature of 293 K, near the center of the span considered. Thus, a question rises regarding the performance in relation with the location of the span, and weather using the Curie temperature as center of span has an impact and maximizes the output.

To this end, the temperature span is now considered around the Curie or a similar temperature:

- Tcenter (Gd) = Tcurie = 293 K
- Tcenter (LaFeCoSi5) = 276 K
- Tcenter (LaFeCoSi11) = 282 K
- Tcenter (LaFeCoSi15) = 288 K

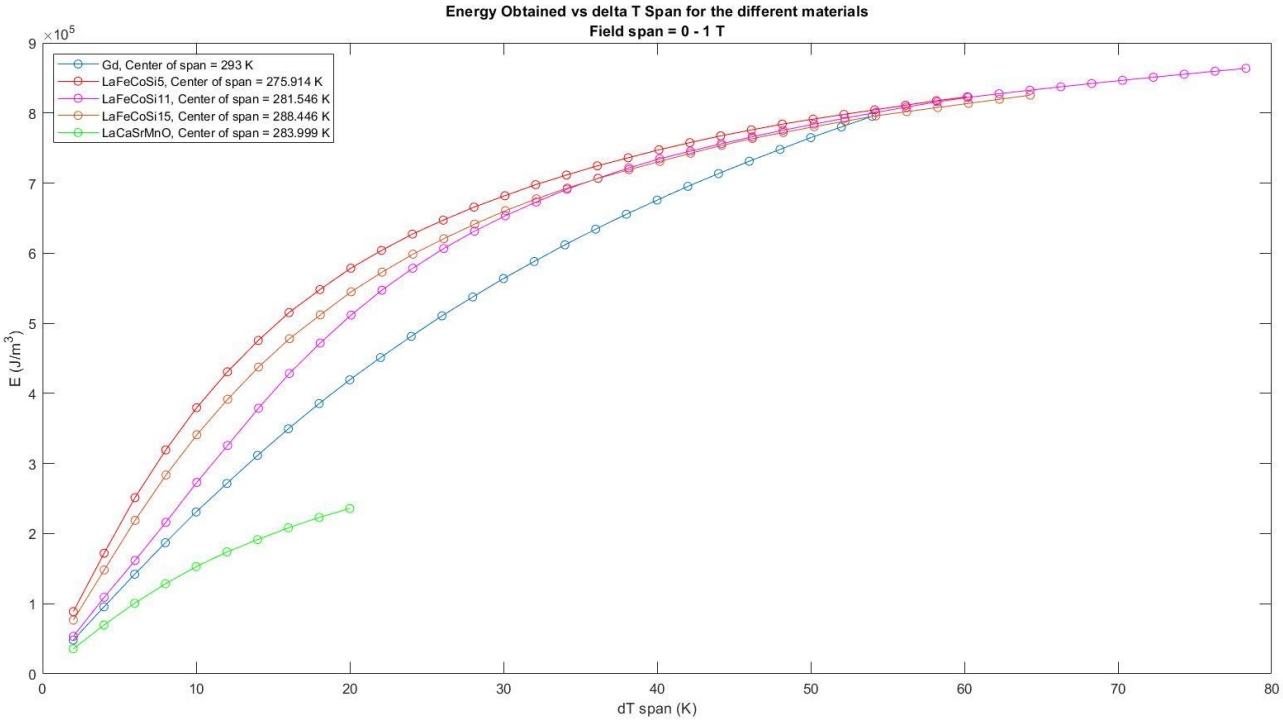
Using these temperatures as center of spans, the materials get a higher output, as can be seen in the table. The new charts can be seen in the APPENDIX A. This means that the location of the temperature span does have a significant impact on the outcome, as expected.

	ENERGY [J / m^3]				
	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
280 - 300 K (274 - 294 K for LaCaSrMnO)	4.12E+05	2.06E+05	5.17E+05	5.37E+05	2.33E+05
Centered around temperature	4.19E+05	5.79E+05	5.12E+05	5.45E+05	-

31 Table containing the energy per volume obtained for all materials studied for the first study case (280 - 300 K) and for a temperature span centered around the Curie temperature. Some materials experiment an increase in outcome.

It is reasonable to state that the length of the temperature span also influences the energy harvest: the greater the span the larger the area between curves in the magnetization-field chart, thus the greater the energy output. However, different lengths of temperature span can have a different impact. It is important to know this relationship, as larger spans are more difficult to implement and can be not worthy.

With this objective, a study is made showing the energy output in relation to the temperature span. The center of the temperature span is chosen to be the Curie temperatures used before:



32 Energy outcome as a function of temperature span for all materials. Field span from 0 - 1 T.

This study reveals that energy harvesting and temperature span clearly relate linearly for all materials studied when considering short and large spans of temperatures. This

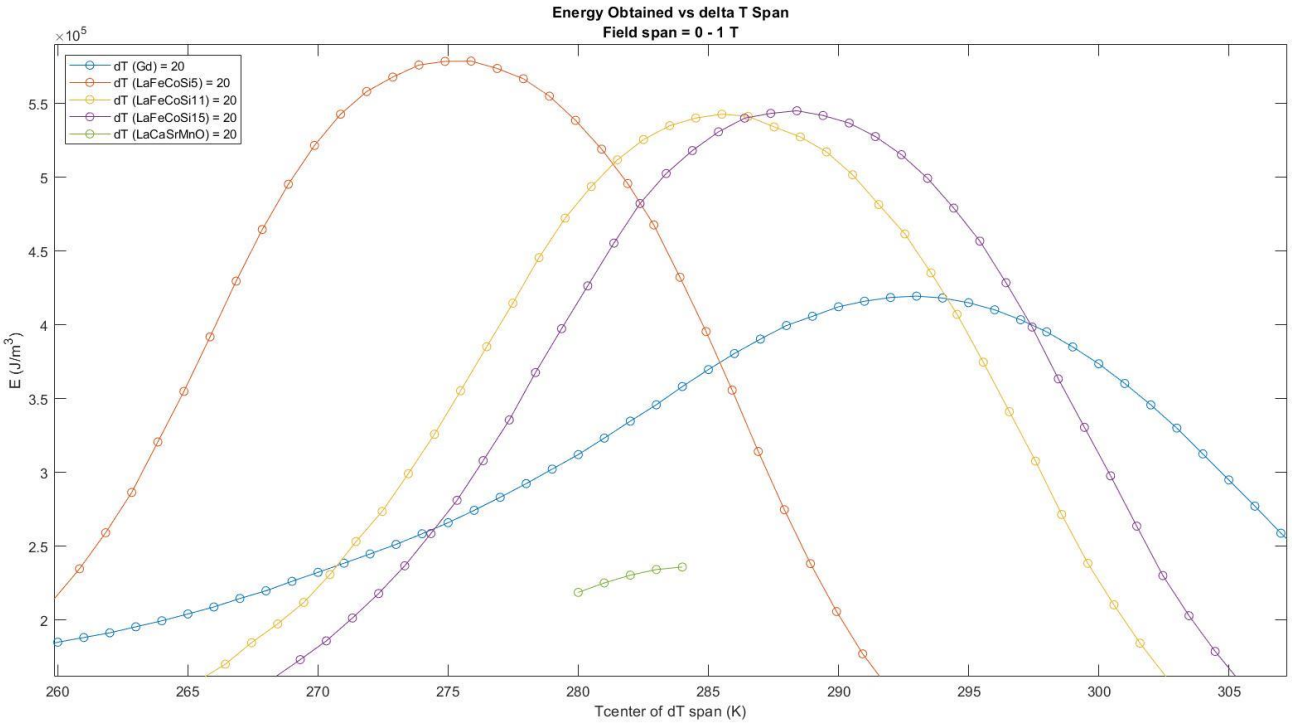
divides the energy harvesting curve in three regions: a first region of up to approximately 20 K of span (depending on the material), where the approximated line is very steep, an intermediate region with no clear relation, and a third from approximately 40 K span and on where the slope is less significant.

This is worth noting when considering the model implementation: for spans within the first region investing in a larger span results in a high increase in energy production, for ranges within the last region investing in larger spans increases the energy output slightly and may therefore be or not be profitable; for spans in the intermediate region the increase of energy output with temperature span is not easily related.

Therefore, **the impact of the temperature span length is not constant for all ranges of spans, and is more important for spans under 20 K**.

It was seen before that the location of the temperature span is determinant to the output, and that setting the center of the span at the Curie temperature could improve performance for the given conditions (0 – 1 T, 20 K span). However, it is not clear that the optimal performance is reached for the Curie temperature. In order to get a better understanding about this, the energy output was studied as a function of the location of the span, being the span of a fixed length (20 K).

A complete graph can be seen in the APPENDIX A for each material showing the energy harvest for a fixed temperature span across all the different temperatures, including a detailed explanation. The result here shows that the optimal temperature to work as center of span is very close to the Curie temperature; this is shown by the peak location. The exact temperatures can be seen in the table of *figure 24*.



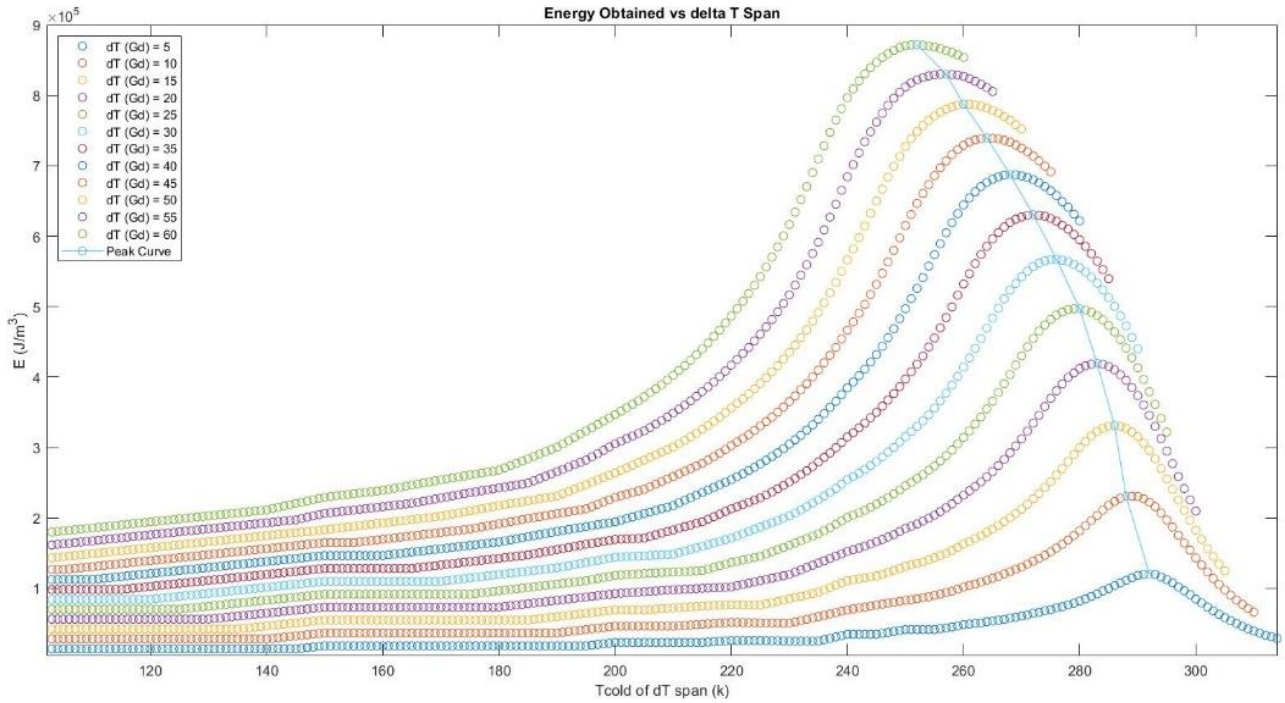
33 Energy obtained for all materials for a temperature span of 20 K as afunction of the center temperature. Note that the location of the span can optimize the output.

	T center of maximum energy for different field spans (20 K span)				
Field	Gd	LaFeCoSi5	LaFeCoSi11	LaFeCoSi15	LaCaSrMnO
0 - 0.5 T	291	273.9	284.5	286.4	283
0 - 1 T	293	275.9	285.5	288.4	284
0 - 1.2 T	293	275.9	285.5	288.4	284
0.5 - 1 T	296	276.9	286.5	290.4	-

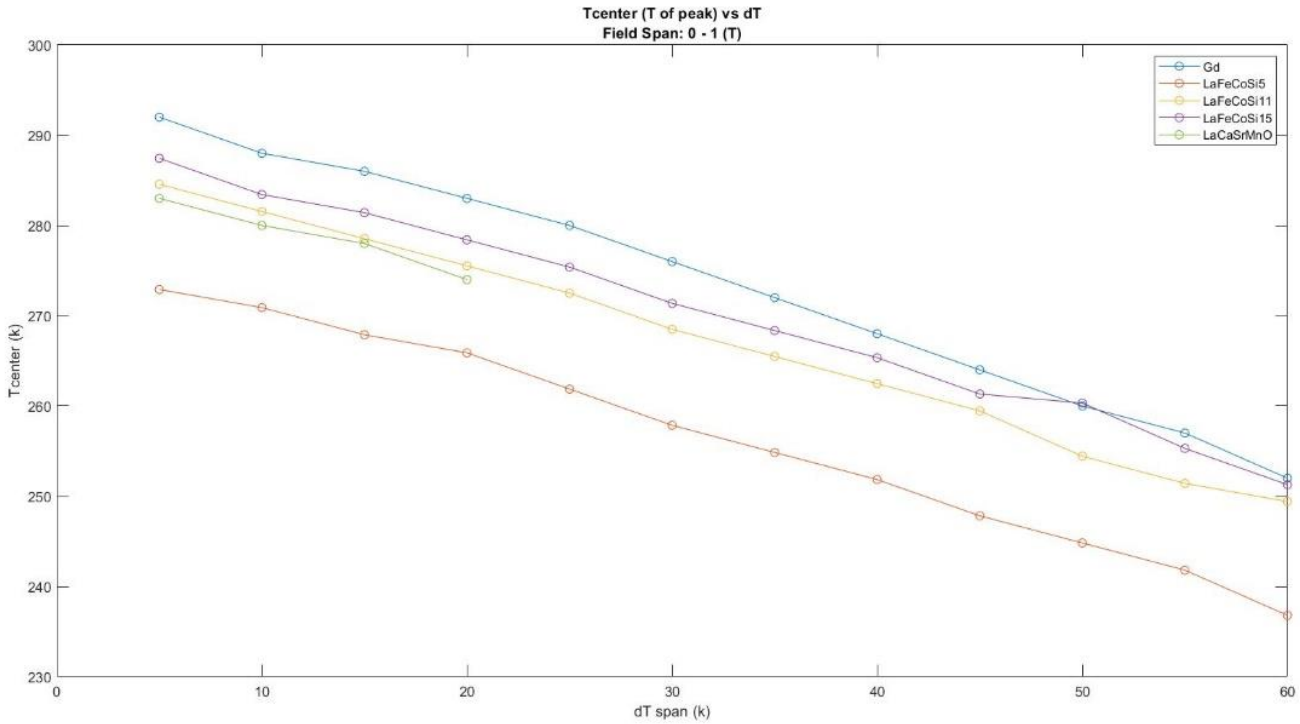
34 Center temperature of maximum energy output for various field

Besides this result, it is still not clear the relation between the optimal center temperature and the Curie Temperature. This result was obtained for a specific length of temperature span (20 K) and a specific field span (0 – 1 T). If the temperature span is changed, it can be seen on *figure 25* that the optimal temperature varies, **the optimal center temperature decreases for increasing spans**. *Figure 26* shows that this relation is approximately linear for all materials. The same is true when considering other field spans as can be seen in the APPENDIX B. This means that **the Curie temperature cannot be taken as the optimal temperature for the center of spans**, as this only happens under certain circumstances as described before.

As expected, the figure also shows that the larger the temperature span the greater the energy output. An example for Gd can be seen in the figure below (*figure 25*), a peak curve joining all peaks of maximum performance for the different temperature spans has been added to better reflect the tendency:



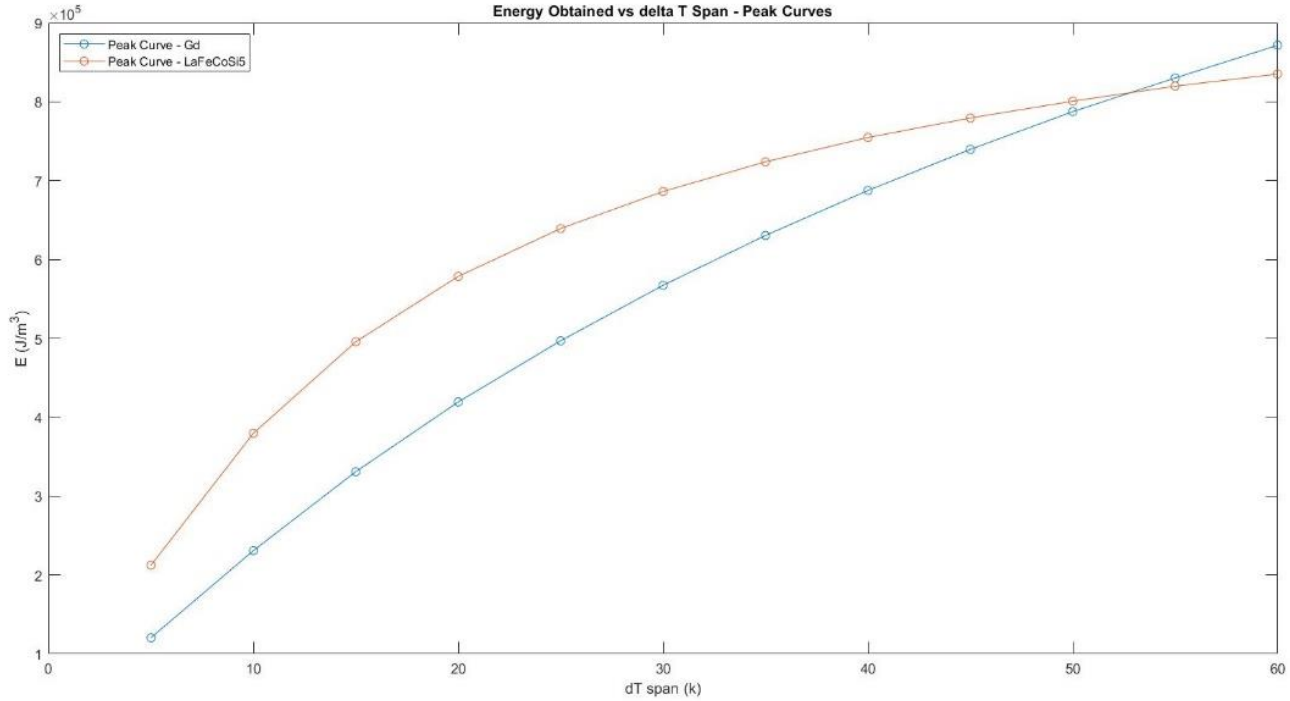
35 Energy obtained for different temperature spans (Gd, 0 - 1 T).



36 The optimal center temperature decreases linearly for increasing spans.

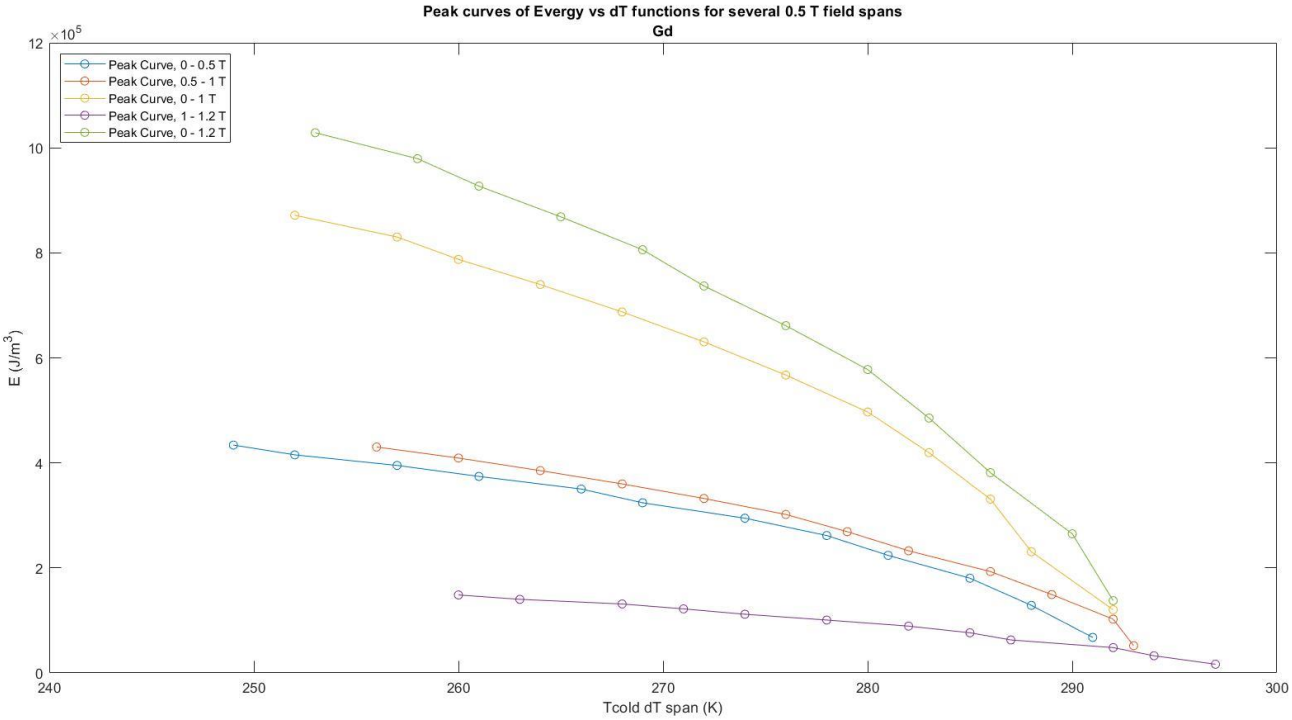
Other graphs in APPENDIX B show the same tendency for the rest of materials and for varied field spans, as well as a broader comparison between materials working with the same spans.

Figure 23 also shows that the maximum energy per volume output is achieved when using LaFeCoSi5. The study revealed that LaFeCoSi5 is the most productive material until temperatures spans of up to 53 K, when Gd outperforms it. To represent this, a chart was created figuring the peak curves of the different spans for both materials. Adjusting a function to each curve permitted the calculation of the exact temperature span at which Gd outperforms LaFeCoSi5, where both curves meet, resulting at 53.16 K.

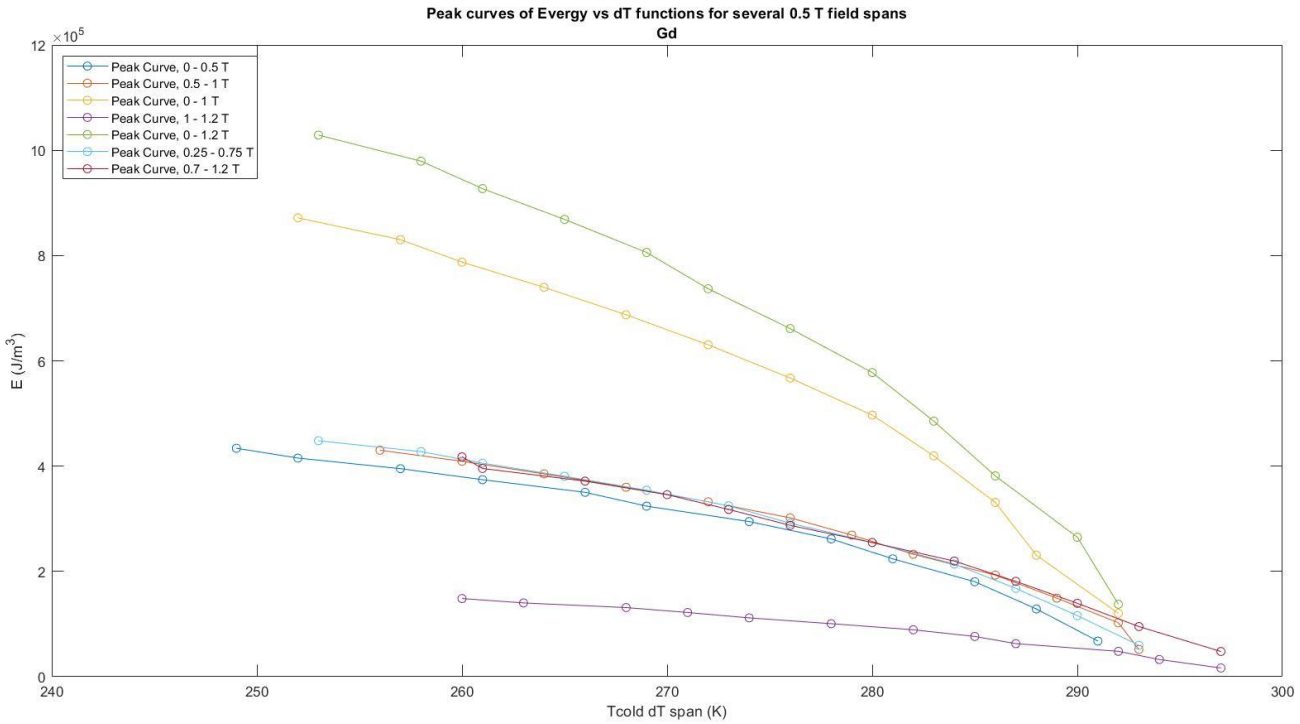


37 LaFeCoSi5 has the best performance for short and mead temperature spans. For large spans, Gd is performs best.

To study the effect of the field span on the optimal center temperature a chart was made considering the peak curves for various field spans.



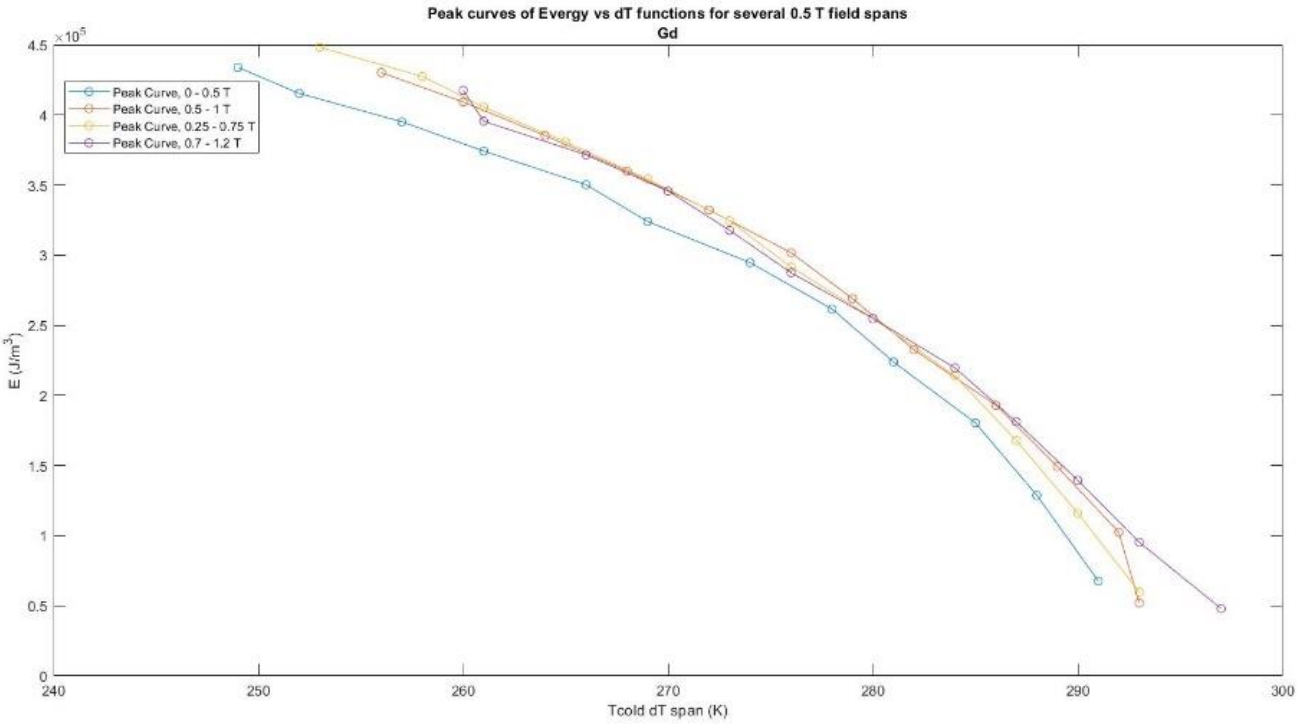
38 Effect of field on peak curves.



39 The close peak curves for 0.5 T spans suggest the location of the field spans does not matter. More 0.5 T spans are plotted at different field locations to confirm this.

Figure 28 shows the peak curve for various field spans. Temperature span increases in the peak curves from right to left, in the direction on lower temperatures. Note that the peak curves are steeper the larger the field span: the larger the field span the greater the energy obtained for the same temperature, the greater the slope of the curve. This means that **large energy harvest can be achieved even for small temperature spans if the field span is high enough**.

Another interesting result worth noting is the little difference between peak curves 0 – 0.5 T and 0.5 – 1 T. This could indicate that the range location of the field span is not determinant. To prove this, more field spans of the same length were plotted across the range of field available in the datasets (0 – 1.2 T). *Figure 29* and *figure 30* show a very interesting result, were all field spans considered (all with the same 0.5 T length) show almost identical results with the peak curves together, confirming the rule mentioned before; only 0 – 0.5 T is a bit different.

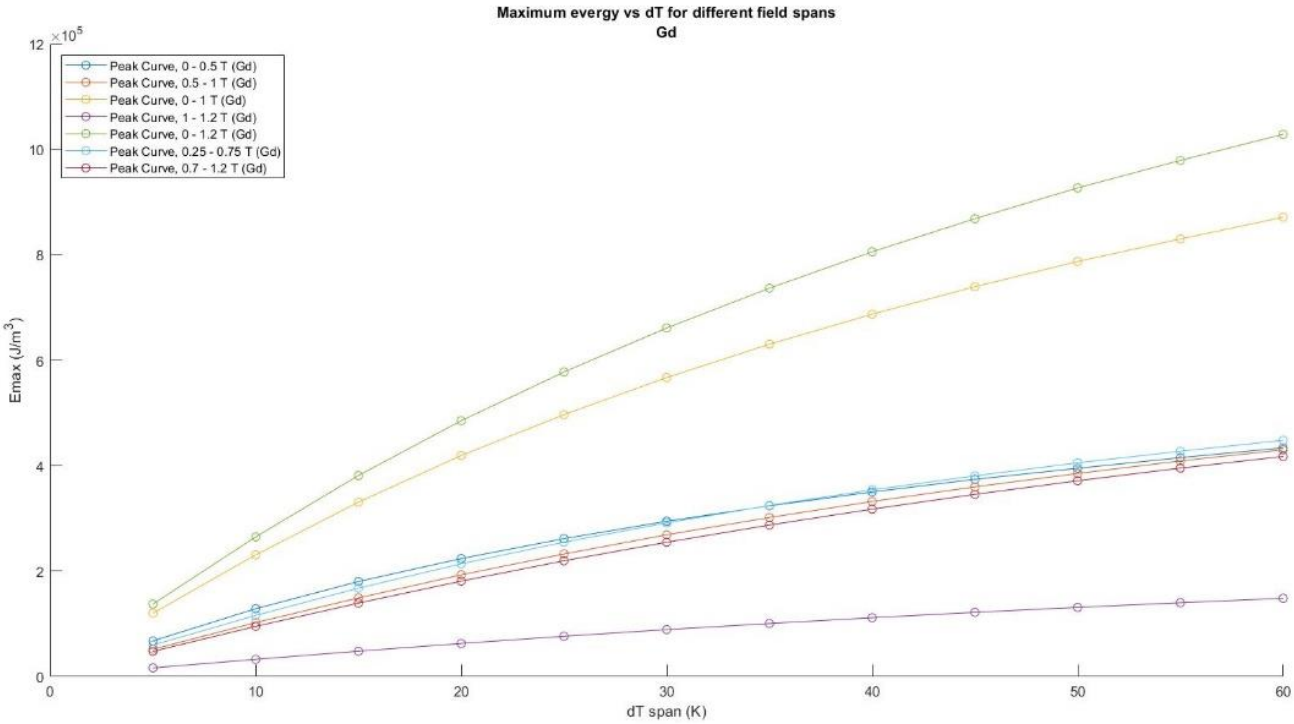


40 The energy output is fairly the same for a given field length regardless of its location, as shown in the concurrent peak curves.

This means that **the energy per volume output is nearly the same for a given range of field, despite where the range is starting from**. This is very important because similar results in energy harvest can be obtained regardless of the span location, which can be difficult to set. For example, in the Curie motor proposed the use of a permanent magnet makes it really difficult to reach a zero-field situation, no matter how far the thermomagnetic material goes. However, with this results in mind the energy harvest can still be the one desired as long as the maximum field is increased by an amount that compensates the minimum field achievable, so that the field span desired is maintained.

Figure 31 shows a similar plot comparing the maximum (optimal) energy obtained as a function of temperature span for different field spans for Gd. The figure is useful to illustrate the rules discussed before: larger field spans give larger energy harvest; larger temperature spans return higher energy output and equal field spans have similar performance regardless of their location.

Similar graphics are provided in the APPENDIX C for the other materials, showing the same rules. Comparisons between materials are also included.



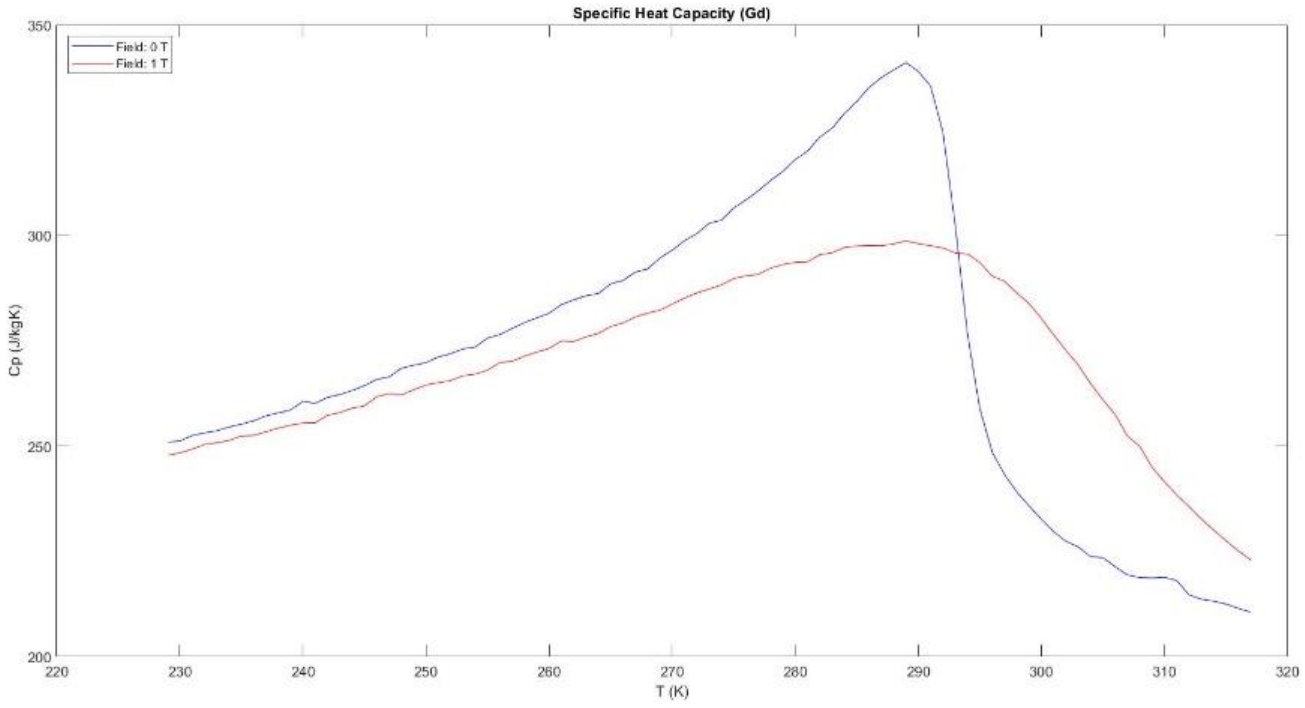
41 Peak curves showing maximum energy as a function of temperature span for different field spans.

About Energy Consumed:

Before harvesting energy using the thermomagnetic cycle, a larger amount of energy has to be transferred to the material from an external source. Most probably, this will be done by a warm liquid like water. When the water is in contact with the thermomagnetic materials it gives part of its thermal energy to the material, this thermal energy will be responsible for the disordering of internal spins and the demagnetization of the thermomagnetic. The conduction of heat during the process will be considered to be ideal (there will be no heat conduction losses considered) as it is out of the scope of this project. This way, the amount of heat that can be transferred from the water to the material is mainly a function of its heat capacity coefficient Cp.

The heat coefficient is at the same time, a function of temperature and field applied as descried before, and can be calculated as:

$$Qin = \rho \int_{Tc}^{Th} Cp\,(T,H)\,dT = \rho \sum Ci * \Delta T$$



42 Illustrative example of specific heat dependent on both temperature and the field applied (Gd).

In the process of ideal heat transfer from the water to the material, the temperature of this one varies from Tc (*cold temperature*) to Th (*hot temperature*), and the field from a low field value to a high field value. The integral can be calculated as the summation of C_i times the individual temperature change associated to each C_i. This method is adequate as a computational method and was used in this project to calculate Cp values.

A value of Cp was calculated through the designated temperature span for the low field value. Then another Cp value was obtained for the high value of field. The Cp is assumed to be linearly distributed so that a mean Cp can be calculated as the mean between these two values.

Once the mean Cp value is obtained, multiplying times the temperature span times the density, the energy per volume consumption is obtained. This, along with the energy per volume harvested obtained before can give the performance of the device.

As an example, for a temperature span of 20 K from 280 – 300 K and a field applied from 0 – 1 T, the efficiency achieved for Gd is by the formula below equal to 4.2*10^5 / (4.7196*10^7) = 0.89%

$$\eta(or\,\eta_{abs}) \cong \frac{\mu_0 \oint H\,dM}{\rho \int_{T_c}^{T_h} C_p(T)\,dT}$$

<u>The search for an energy formula</u>

It would be interesting to have a formula for the energy production as a function of the temperature span, so that predictions and results comparisons would be possible and that an idea of the needed ΔT could be obtained. The problem when trying to find a formula for the energy harvesting is that the energy obtained depends on many different variables that are interdependent at the same time: the field and temperature span, specific heat capacity, entropy…

Here, a formula for energy harvesting using thermomagnetic materials in a Curie motor is provided, derived from the Curie's Law of Magnetization making certain assumptions. Later, the legitimacy of this formula will be discussed.

As it was described before, the energy harvest of a Curie motor thermomagnetic cycle can be obtained as:

$$E = \mu \oint H\,dM$$

Curie's Law of Magnetizations states that magnetization is proportional to the external field applied and inversely proportional to the temperature of the material, relating these parameters with the following formula:

$$M = c\ \frac{Bext}{T}$$

Where *M* is the magnetization of the material, *Bext* is the external magnetic flux density applied, T is the material's temperature and c the Curie constant specific to the material. The magnetic flux density can be written as:

$$B = \mu\, H$$

Where *H* is the magnetic field strength, here referred to as external field applied. Hence, Curie's Law can be rewritten as:

$$M = c\frac{\mu\, H}{T}$$

This permits the energy equation to be written as:

$$E = \mu \oint \frac{H^2 c\mu}{T} dH$$

The temperature component in this equation is not constant and varies along the cycle. To solve this, the closed integration along the cycle is divided into four integrals representing the four sequences of the cycle. As ideally, the external field applied is constant during phase 2-3 and 4-1 of the cycle, those integrals can be considered to be equal to zero. The remaining integrals for phase 1-2 and 3-4 take place while temperature is a constant (see *figure 33*), so the integrals can be solved:

$$E = \mu \int_{1\to2\to3\to4\to1} \frac{H^2 c\mu}{T} dH = \mu \int_1^2 \frac{H^2 c\mu}{Tc} dH - \mu \int_4^3 \frac{H^2 c\mu}{Th} dH =$$

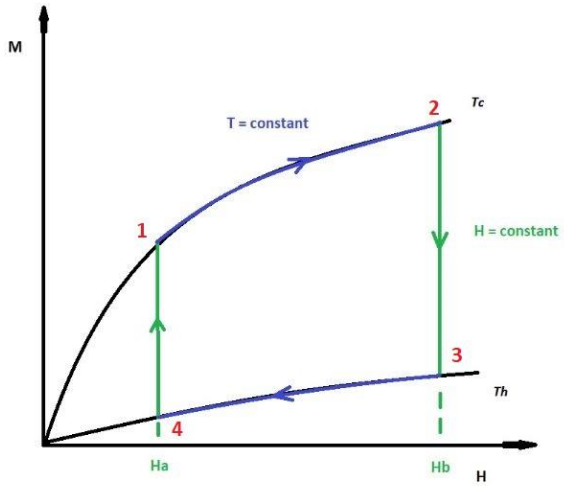
$$= \mu\left(\frac{c\,\mu_1}{Tc}\int_1^2 H^2 dH - \frac{c\,\mu_2}{Th}\int_4^3 H^2 dH\right) = \mu\left(\frac{c\,\mu_1}{Tc}\left[\frac{H^3}{3}\right]_1^2 - \frac{c\,\mu_2}{Th}\left[\frac{H^3}{3}\right]_4^3\right)$$

The equation can be further developed when considering that the field span from 1-2 is the same as in 4-3 and assuming $\mu_1 = \mu_2 = \mu$. The formulation is then:

$$\begin{cases} \Delta H_{1\to2} = \Delta H_{4\to3} \\ \mu_1 = \mu_2 \end{cases} \rightarrow E = \frac{c\,\mu}{3}\,(Ha^3 - Hb^3)\left(\frac{1}{Tc} - \frac{1}{Th}\right)$$

Considering the temperature span ΔT = Th – Tc, the energy equation results:

$$E = \frac{c\,\mu}{3}(Ha^3 - Hb^3)\frac{-\Delta T}{Tc\,Th}$$



43 Ideal thermomagnetic cycle. Note that temperature and field is constant at certain phases of the cycle, allowing the simplification of the integral.

Determining Gd parameters

In order to use the energy equation found, $c\,\mu$ parameters must be found, as they are unknown. The parameters can be obtained from an experimental point in the datasets. In the example of Gd at a field span of 0 to 1 T and a temperature spanning from 280 K to 300 K, the parameters result to be:

$$\begin{cases} Ha = 0\ T \cong 7957{,}75\ A/m\ ; Hb = 1\ T \cong 799633{,}017\ A/m \\ Th = 300\ K\ ,\ \ Tc = 280\ K\ ,\ \ \Delta T = 20\ K \\ E \cong\ 4{,}1215 * 10^5\ {}^{J}/_{m^3} \end{cases}$$

$$E \cong 4{,}1215 * 10^5\ {}^{J}/_{m^3} = \frac{c\,\mu}{3}\,(7957{,}75^3 - 799633{,}017^3) * \frac{-20}{300 * 280}$$

$$\rightarrow\ \ c\,\mu = 1{,}016 * 10^{-8}$$

Testing of the energy formula

Now the formula can be tested for another known point in the dataset:

$$\begin{cases} Ha = 0 \;\; T \cong 7957{,}75 \; A/m \;\; ; Hb = 1 \; T \cong 799633{,}017 \; A/m \\ Th = 305 \; K \; , \;\; Tc = 290 \; K \; , \;\; \Delta T = 15 \; K \\ E \cong \; 3{,}152 * 10^5 \; {}^{J}/_{m^3} \end{cases}$$

$$E \cong \frac{1{,}016 * 10^{-8}}{3} \; (7957{,}75^3 - 799633{,}017^3) * \frac{-15}{305 * 290} = 2{,}937 * 10^5 \; J/m^3$$

The resulting energy has an error of 6,8 % when compared to the computational simulations.

Validity of the formula:

The formula's error when predicting energy harvest is due to the error within the estimation of the parameters c μ. To obtain better working parameters with less error the estimation of the parameters could be done as a mean of many known cases of energy harvest across the dataset rather than just taking on calculation as in this example. However, it is difficult to believe that the formula will be always precise enough. Furthermore, the formula can be dismissed for low temperatures as it was derived from the Curie's Law of Magnetization, which works only for temperatures above the Curie temperature.

The conclusion is that **the thermomagnetic cycle is too complicated and depends on too many interconnected variables to be able to describe it with the formulas used, which justifies the use of computational methods** for obtaining results when considering temperature, field and magnetization variables, as it is done in this project.

How much energy can be taken from the water

The amount of energy that can be taken efficiently from the water is the maximum heat income that can experience the thermomagnetic material in contact with the water. This heat is ruled by the equation:

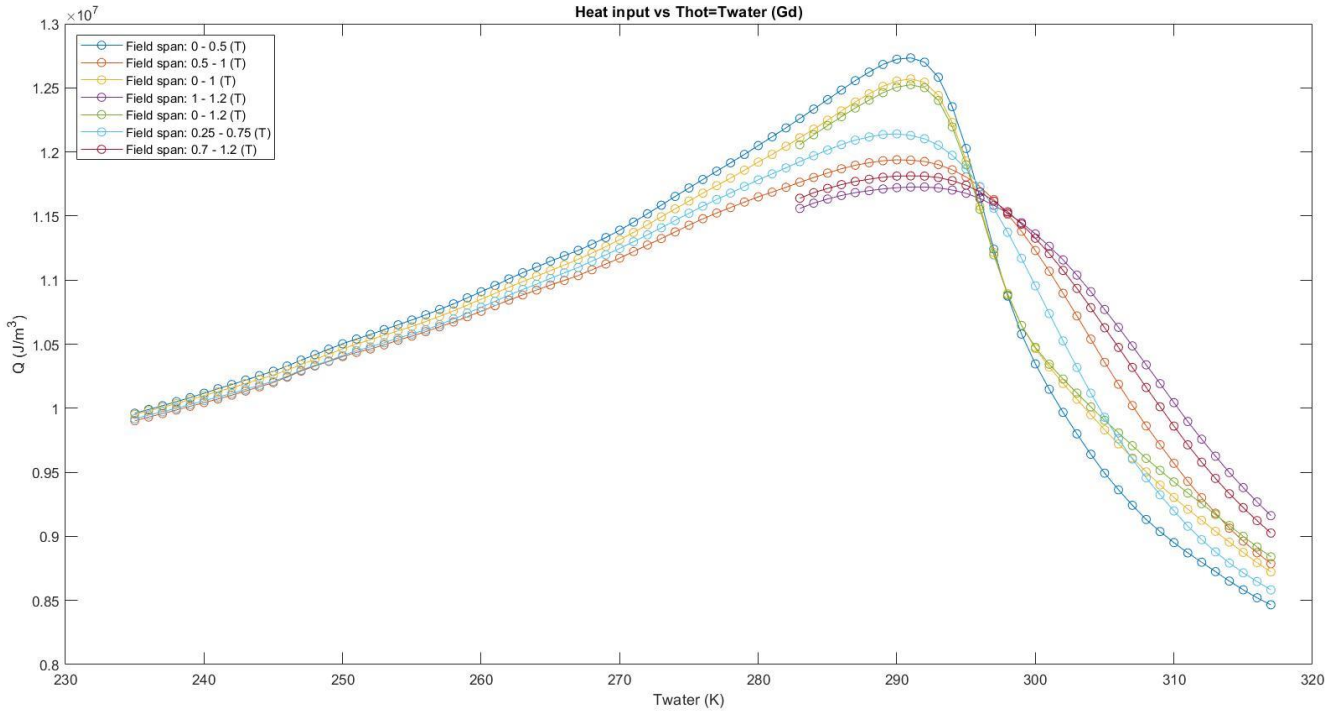
$$Q = m \; c \; \Delta T$$

Where Q is the thermal energy change of the material in [J], m is the mass [kg], c the specific heat capacity of the material [J/kg K] and ΔT the temperature change experimented by the thermomagnetic material [K]. If the mass is substituted by the density ρ, the result is then obtained as energy per volume [J/m^3].

This incoming energy can be studied as a function of field and temperature span. As the specific heat capacity will change with the field and the temperature, the mean Cp should be considered as described before. However, this formulation would require to know the change in temperature of the water, which is not known. In this project, the water will be considered as an ideal infinite heat source, so no difference is experimented by the water.

If a perfect heat transferred is considered, the temperature of the material will be the same as the temperature of the water, ΔT of the thermomagnetic material is the same ΔT of the water. Thus, considering a perfect heat transfer between the water and the material, the limit in energy that can be taken out of the water is given by the difference in the working temperatures and the properties of the material (specific heat capacity…), and is therefore given by the previous studies when the energy per volume output is optimized.

Figure 44 shows an example of the curve of energy extracted from the water. Note the resemblance with the specific heat capacity curve: the extraction of energy is determined by the material (by its specific heat capacity).



44 Energy extracted from the water for a temperature span of 5 K. Note the resemblance with the Cp curves form the material on figure 42. The energy extracted is constrained by the material used.

CHAPTER 5 – CONCLUSIONS

5.1 Conclusions from Methodology

The methodology used evolved during the making of the study. From handwritten calculus, through Excel up to MATLAB and others, adaptability was required to deal with the higher demands of the project as it grew deeper. It is fair to consider one of the mayor achievements of this study the techniques acquired for coding and modeling in MATLAB, which prove to be a crucial tool for modeling the problem and engineering the study cases.

5.2 Conclusions from Results

The following conclusions were derived from the study:

- The cycle of a thermomagnetic motor must be set at specific working temperatures to give the maximum performance. These optimal temperatures vary for the same span of temperature depending on the field applied.

- The different materials have different output for the same parameters, due to their different qualities (Curie temperature…). This means that each material must be optimized separately regardless of analyzing them with the same working spans, which have to be located accordingly.

- There is no clear relation with the Curie temperature of the material and the optimal temperature. Instead, the optimal changes depending on the field applied, though for 1 T and 20 K spans the optimal temperature happens to concur with the Curie temperature.

- Generally, the Curie temperature cannot be taken as the optimal temperature for the center of spans. The optimal center temperature decreases for increasing spans of temperature linearly, for all materials and field spans considered.

- The larger the temperature span the higher the energy obtained, but the impact of the temperature span length is not constant for all ranges of spans, and is more important for spans under 20 K. Three regions of spans can be observed with different impact on energy harvest.

- The highest energy output is achieved by LaFeCoSi5 material for short and intermediate field spans, and by Gd for large field spans. The point at which Gd outperforms LaFeCoSi5 changes slightly on the field applied, and is at 53,16 K for a 1T span.

- Large energy harvest can be achieved even for small temperature spans if the field span is high enough. Field and temperature span can compensate each other.

- The energy per volume output is fairly the same for a given range of field, regardless of where the range is located. This makes construction of a working thermomagnetic cycle device easier to implement.

- The performance of the cycle in a Curie motor device is dependent on the field span, the temperature span and the specific heat capacity of the material. These variables are each related to each other in the thermodynamic cycle.

- A formulation to predict energy output has been proposed according to Curie's Law of Magnetizations for an ideal thermomagnetic cycle where temperature and field applied are constant during certain phases of the cycle. The formula is an approximation based on assumptions and can only be considered for temperatures above the Curie point, as this is a prerequisite for to Curie's Law of Magnetizations from which is derived. Furthermore, the formula needs parameters that are obtained experimentally.

  However, if experiments are available, then the data processing made in this study is possible, giving far better results than the formula's approximation. Hence, it can be concluded that the thermomagnetic cycle is too complicated to be able to describe it with the formulas used and is best understood with computational methods.

- The extraction of thermal energy form the water in an ideal heat transfer situation is given by the characteristics of the material, most notable the specific heat capacity, and the working temperatures.

5.3 Recommendations for future studies

LaFeCoSi5 is recommended as working material for a future prototype implementation: this material showed a very good behavior when subject to different working conditions and achieved the maximum energy harvesting in all cases except for very large temperature spans, in which Gd was better. It was seen however, that for larger temperatures spans the extra energy harvest decrease. This makes LaFeCoSi5 recommendable, also because large temperature spans may not always be possible as working conditions. On top of this, LaFeCoSi5 is a much cheaper material, making it the best option to work with.

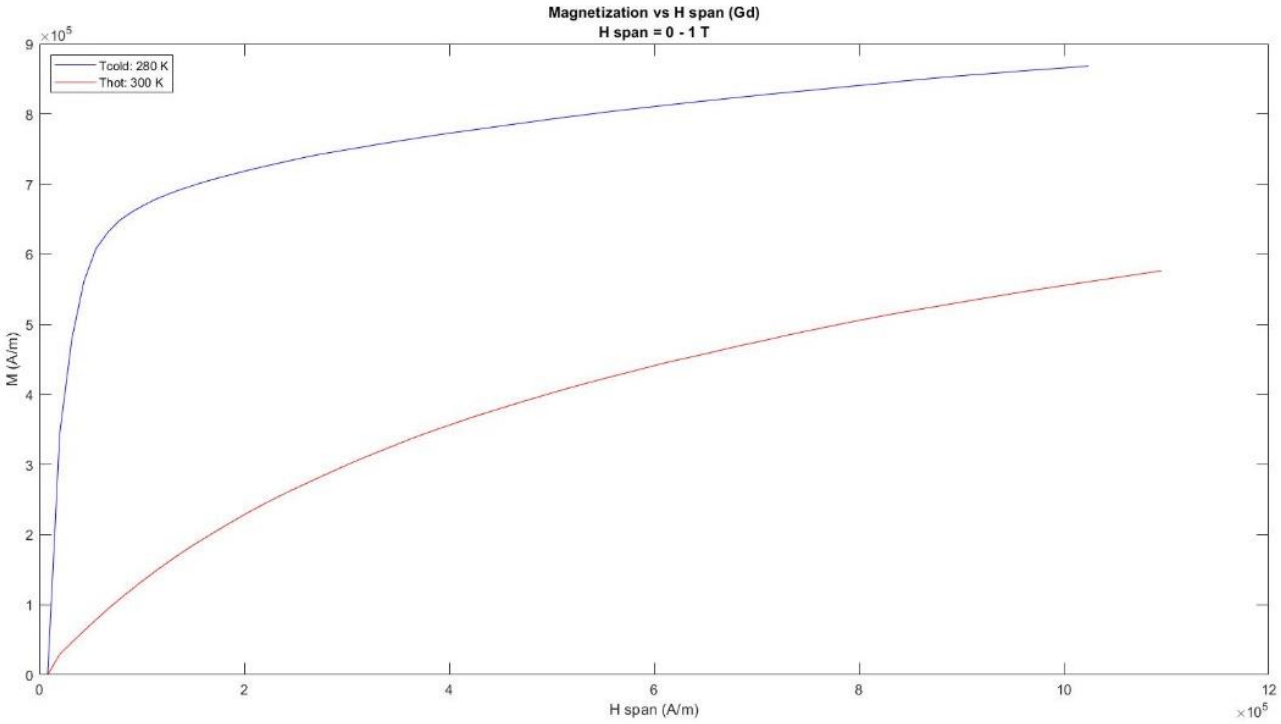
It is also recommended for a future study to evaluate at what rate energy can be extracted from the water. As it was stated, the warm water is wastewater from

industry, so no expense is necessary for heating the water. This means that harvesting energy at a profitable rate will be determined by the time it takes to heat the material and complete the cycle. As such, a future study of energy transfer and cycle time is recommended, studying time and water needed and how big the system should be to harvest at a profitable pace.

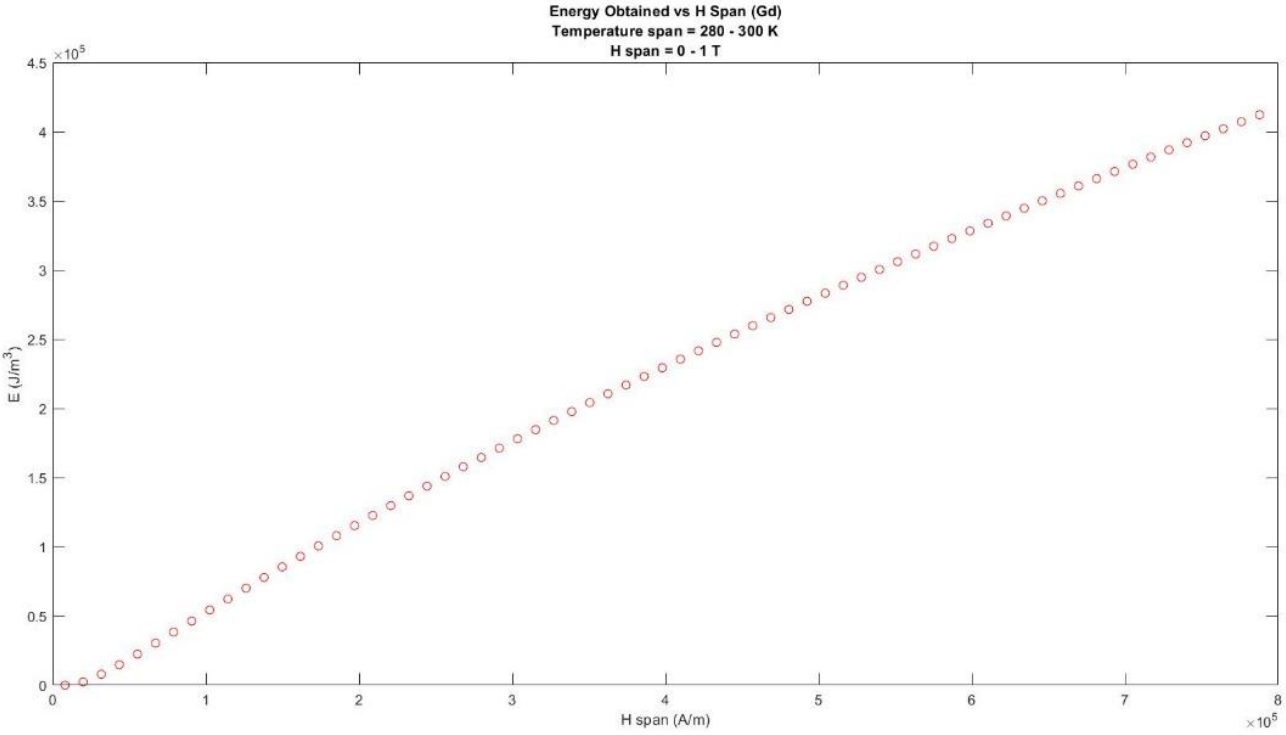
APPENDIX A – EXTENDED RESULTS FOR THE STUDY OF THE CYCLE

Gd:

The script *Harvest_integration_Gd_E_vs_H.mat* was ran to calculate the energy obtained from the thermomagnetic cycle when operating from Tcold = 280k to Thot = 300k, varying the applied field from 0T to 1T. The following figure shows the magnetization curves for both temperatures. The energy obtained is calculated as the area between the curves, and between the working fields.

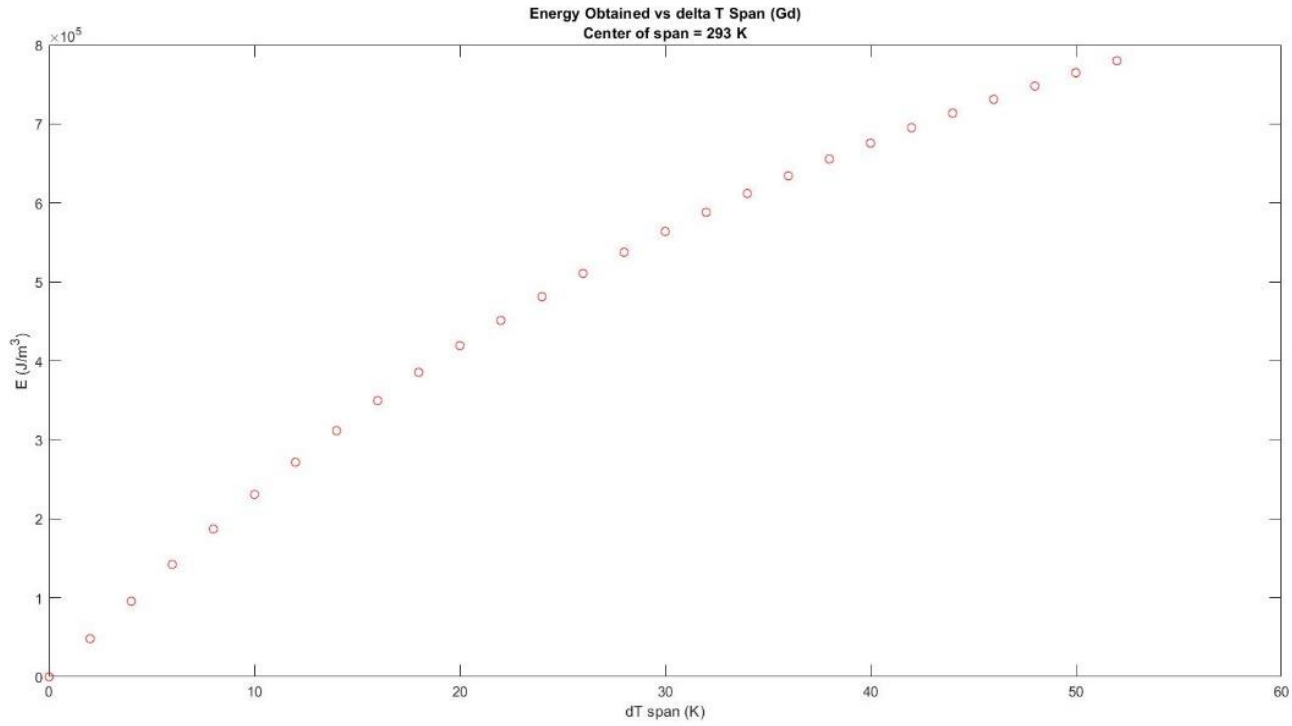


Then a graphic is produced to represent the energy obtained according to the applied field span:



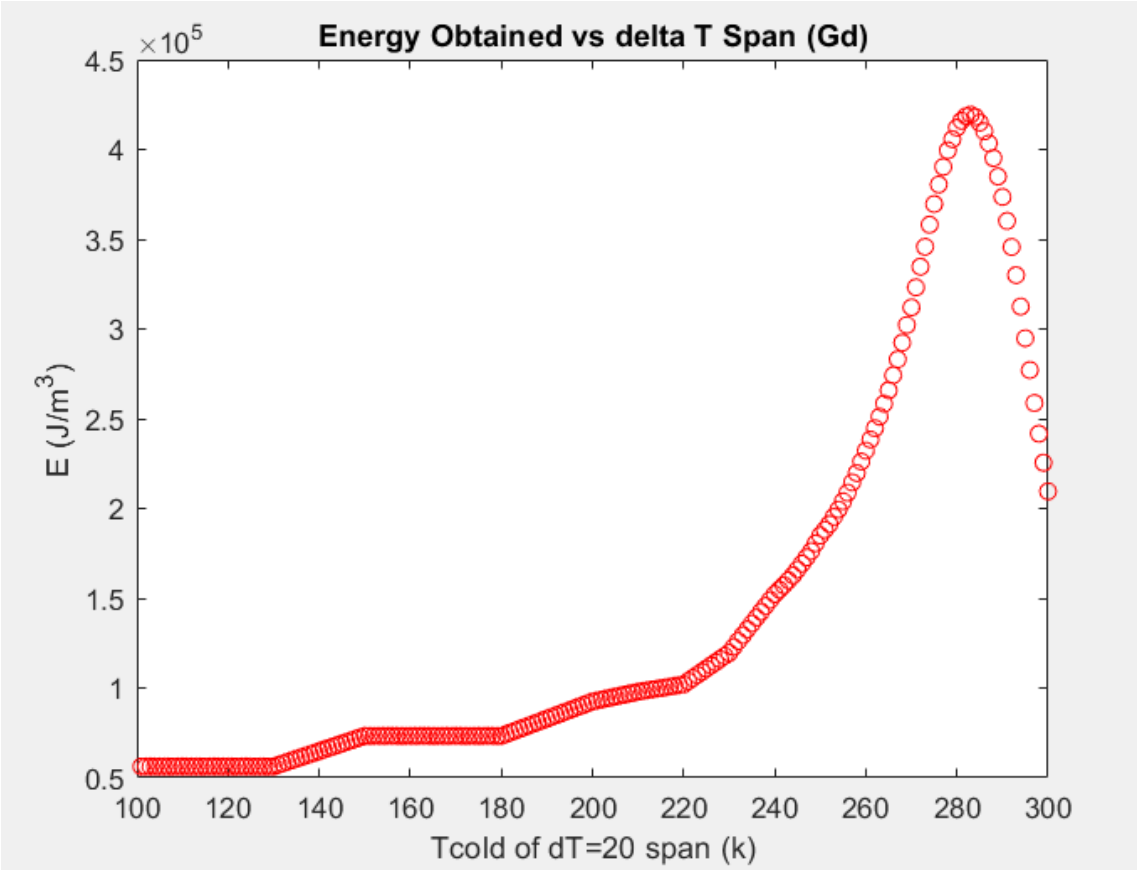
As it can be seen, the greater the field span the higher the energy obtained. Note that the energy obtained is in Jules per cubic meter, so the graphic results are showing energy per volume.

Now, obtained energy versus temperature span is studied. Tcurie = 293 K was chosen a center temperature around which the span increases. The results can be seen in the following figure:



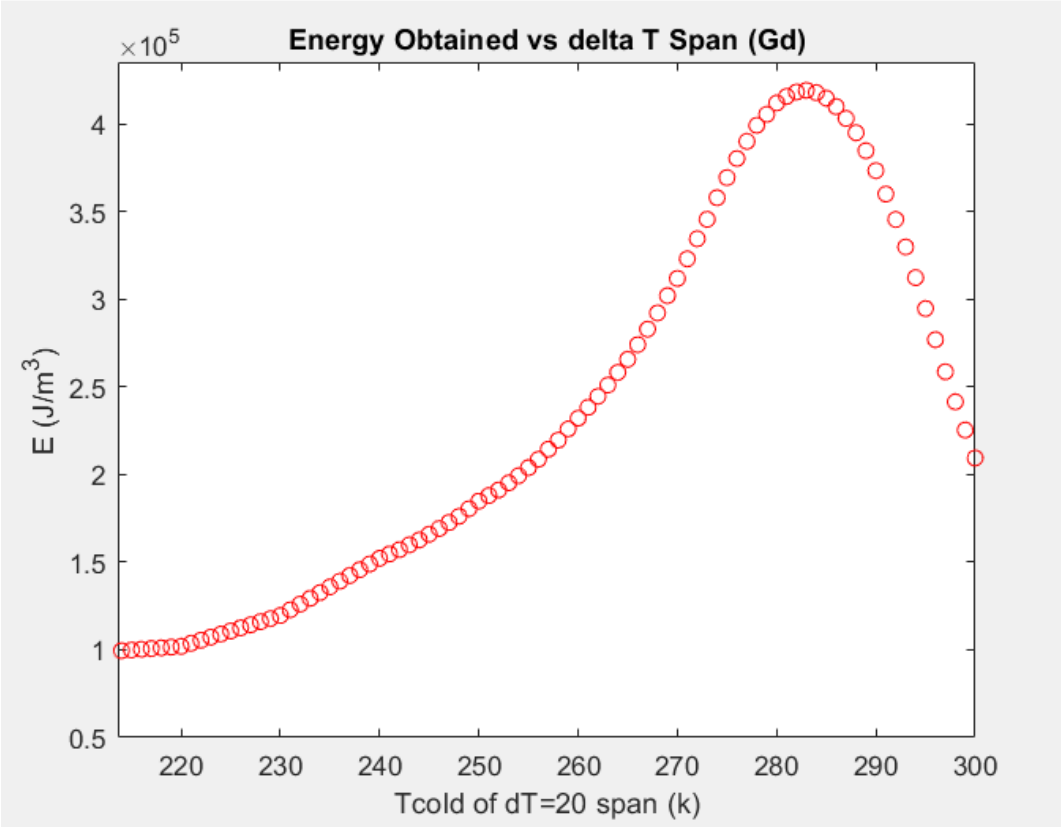
This study was made for a field ranging from 0T to 1T, so it can be seen that the same energy value is obtained for a 20k span. Note that the energy per kelvin difference, the slope of the curve, decreases when the temperature difference is big enough. This could be due to the fact that the bigger the span is, the more far the temperatures are from Tcurie, so the impact on the magnetization (and thus energy) is reduced.

Another script was used to represent the resulting energy per volume when a one Tesla difference is applied with a difference of 20k in the working temperatures. The temperature span is fixed and moves along the range of temperatures to find the optimal location of the span.



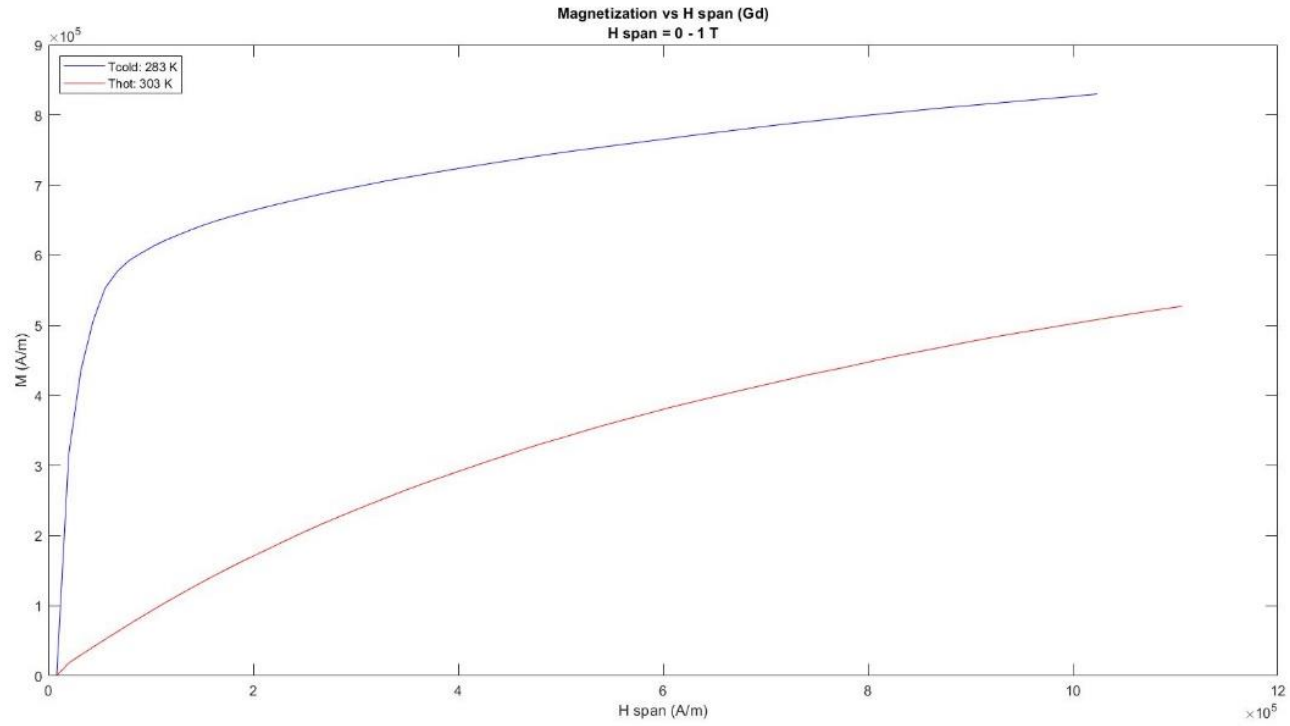
Note that the temperature represented in the horizontal axis of the graphic is the cold temperature. As we are considering a temperature span of 20k, the hot working temperature will be obtained adding 20k to the lower temperature. Similarly, the mid temperature acting as the center of the span will be simply 10k more than the cold temperature.

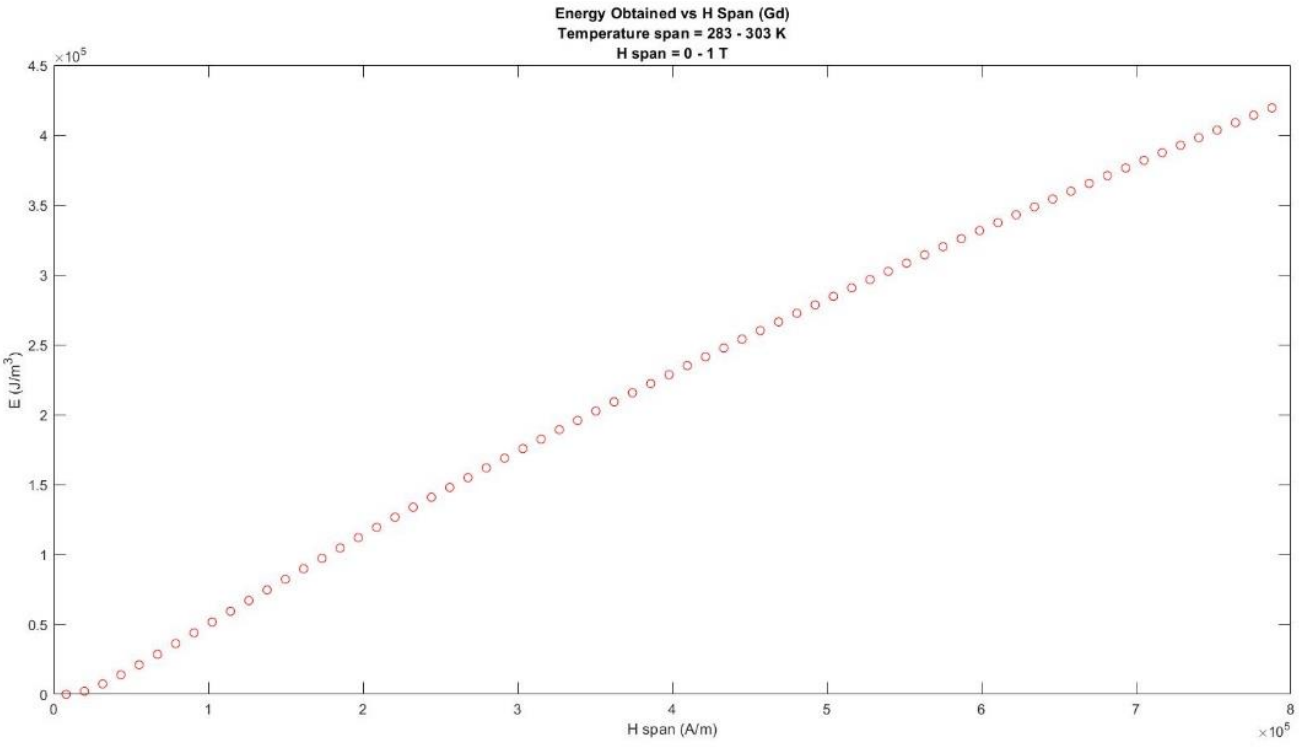
In order to achieve a more representative graphic of the working process, the graphic is zoomed, and the extreme of the temperature range is removed. It is reasonable to remove this part of the graphic as the data obtained for this marginal temperatures is not so accurate, and any way, not very useful as working temperatures.



The maximum energy per volume obtained is 4.194 *10^5 J/m^3 at a cold temperature of 283k. This implies the high working temperature is 303 K and the 20 K span is centered at 293 K, which is in fact the Curie temperature for this Gd material. This way, the peak in energy is achieved when the temperature span is centered at the Curie temperature. This is also seen as a tendency across the graphic, so the nearer the span is of the Curie temperature, the higher the resulting value of energy will be.

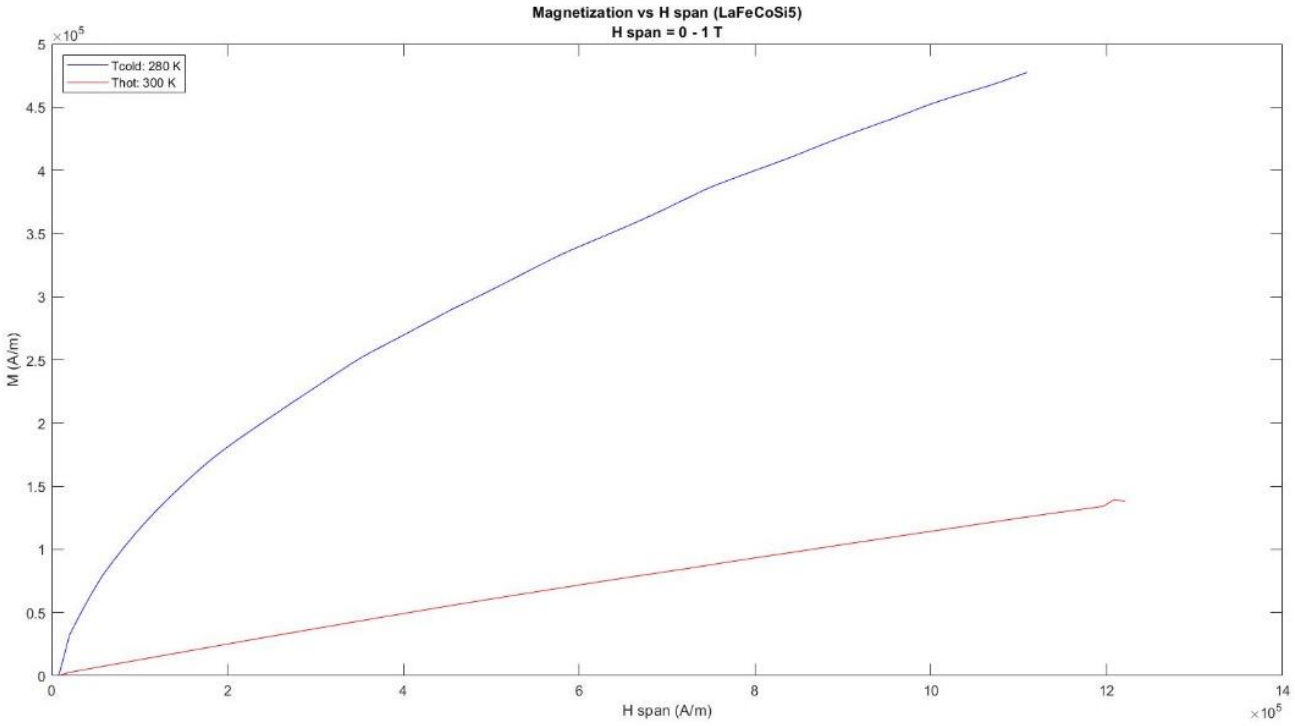
If the cycle is considered around the Curie temperature, the following is obtained:


Magnetization vs H span (Gd)
H span = 0 - 1 T
Tcold: 283 K
Thot: 303 K
M (A/m)
H span (A/m)
×10^5

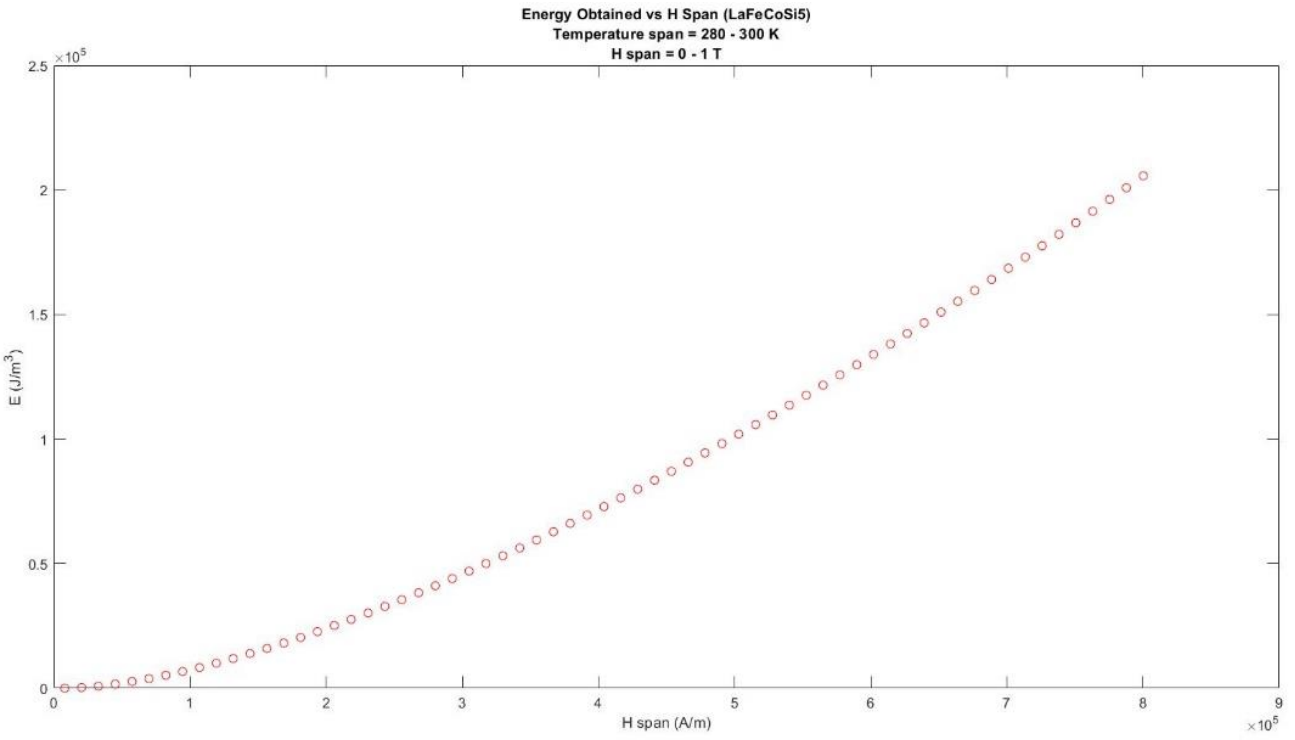

Energy Obtained vs H Span (Gd)
Temperature span = 283 - 303 K
H span = 0 - 1 T
E (J/m^3)
H span (A/m)
×10^5

LaFeCoSi5:

The *Harvest_integration_LaFeCoSi5_...* were scripted to evaluate the effect of field and temperature parameters on the energy output for LaFeCoSi5 material. For a field span of 1T (from 0T to 1T) and using as low working temperature Tcold =280k and high working temperature Thot=300k, the following results were obtained:

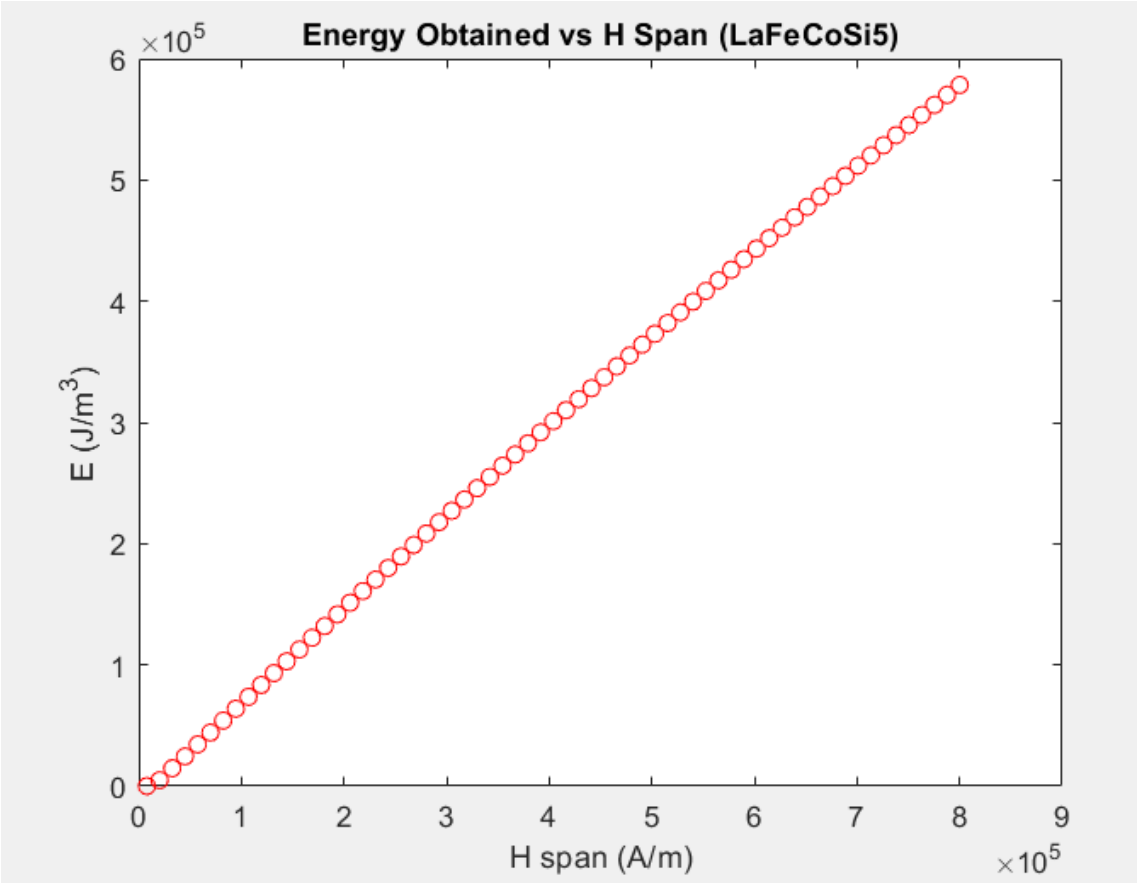


It can be seen that the curves are more linear to that of Gadolinium.

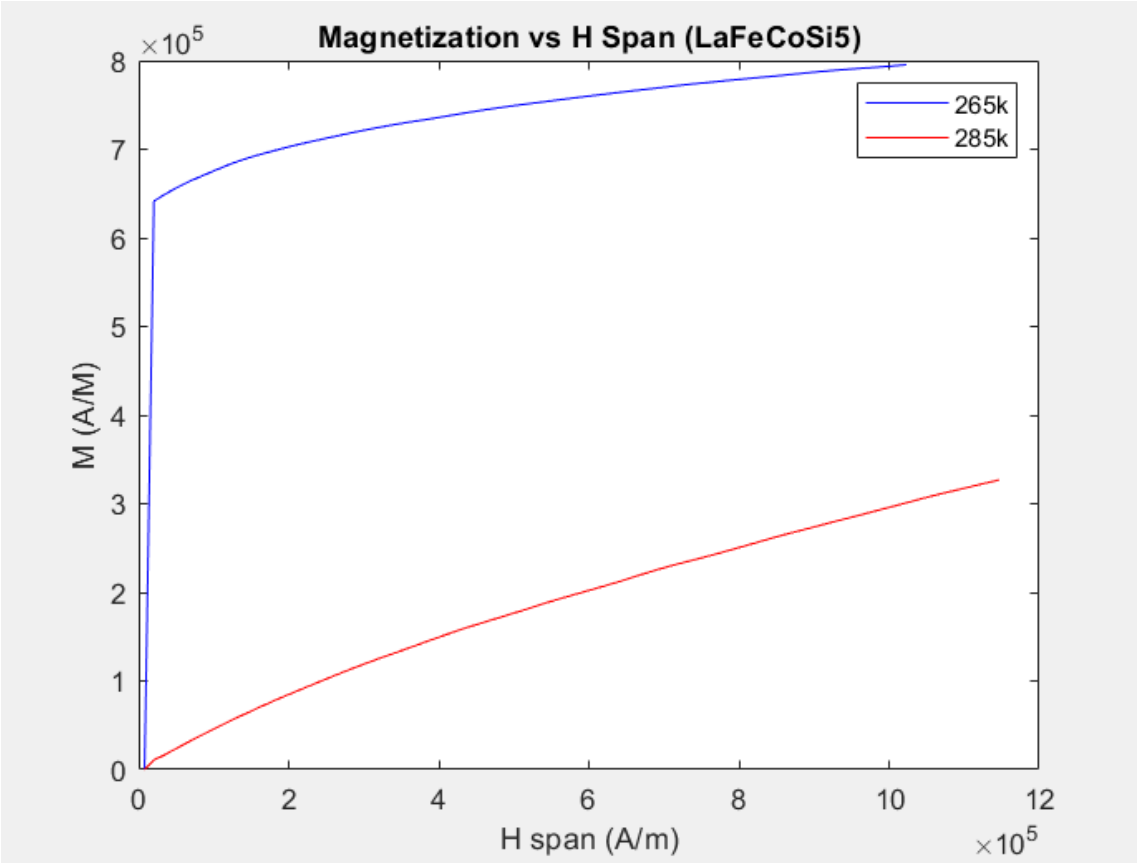


We can see from the resulting graphic that the energy obtained for a field span of 1T is far less than the same result for Gadolinium (2.057*10^5J/m^3 in comparison to 4,122*10^5J/m^3). This is because the same temperature span was considered (280k to 300k), were Gd Curie temperature is considered (293k), while LaFeCoSi5 Curie temperature is not (275k, peak in 20k span curve).

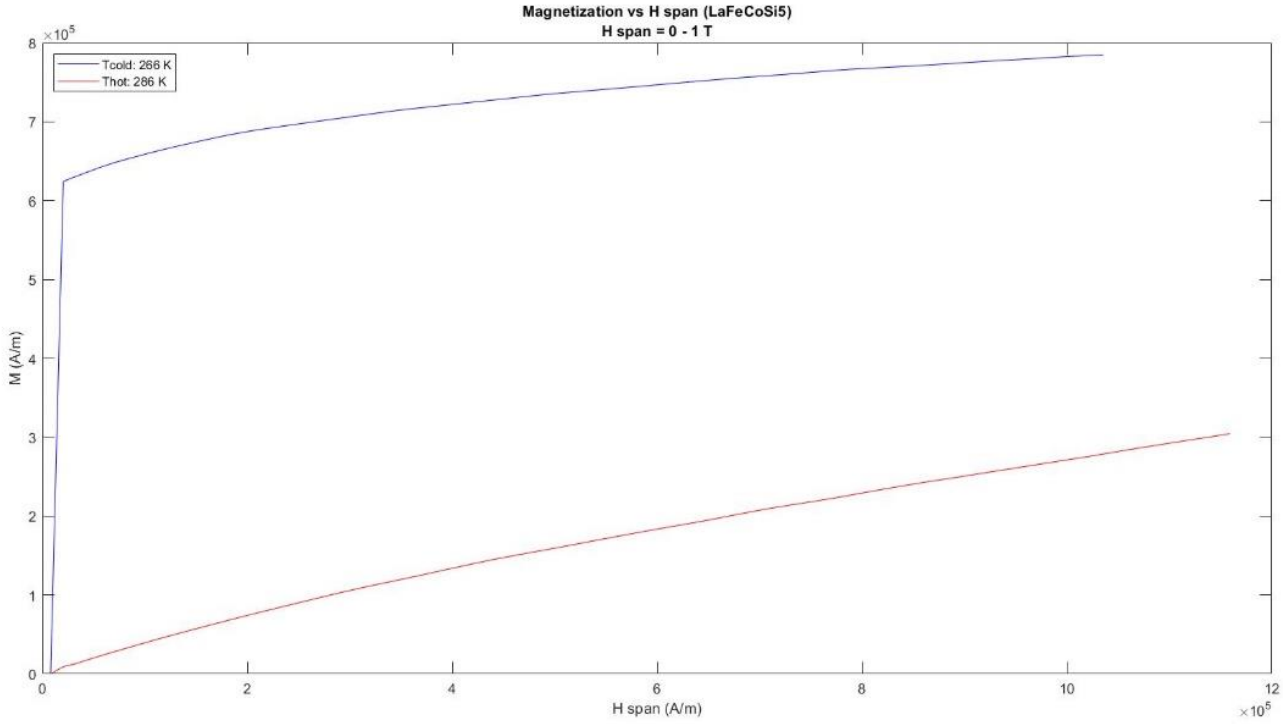
If a new Tspan of 265k to 285k is now considered, the following is obtained:

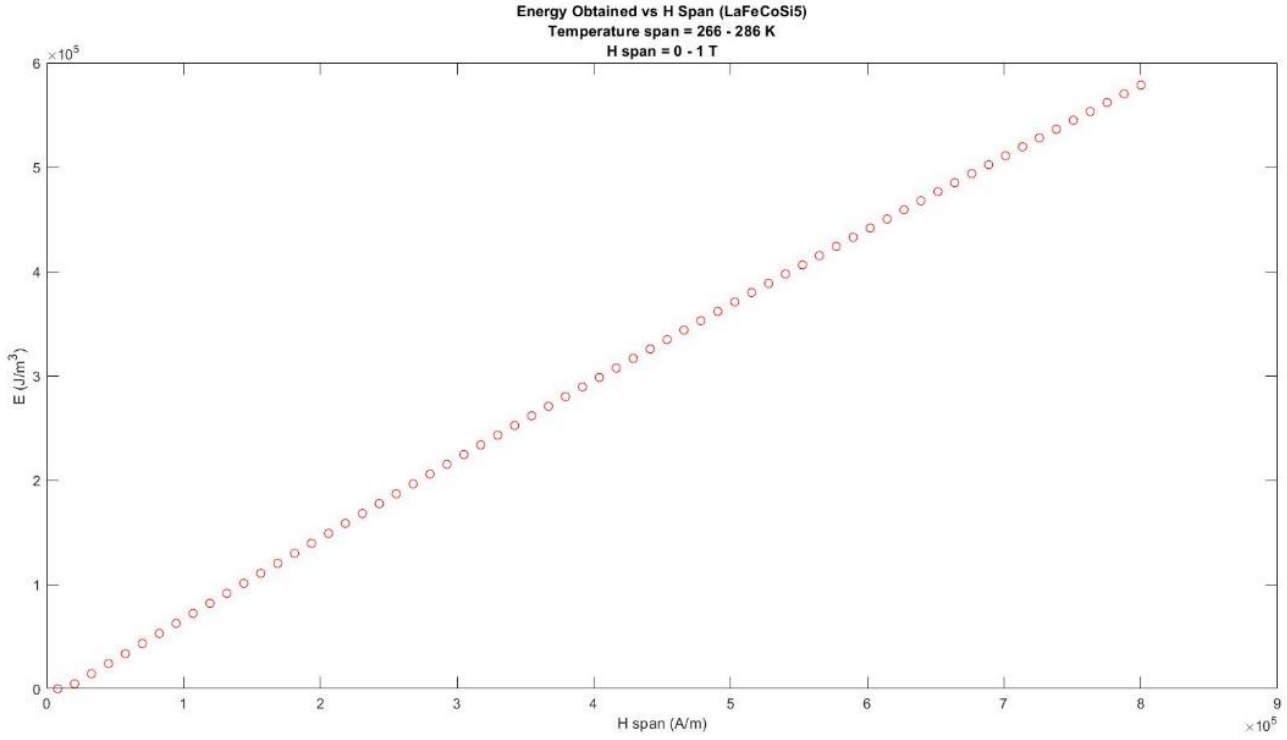


Where the maximum energy produced for a field span from 0T to 1T is 5.784*10^5 J/m^3. The new cycle is shown in the following figure:

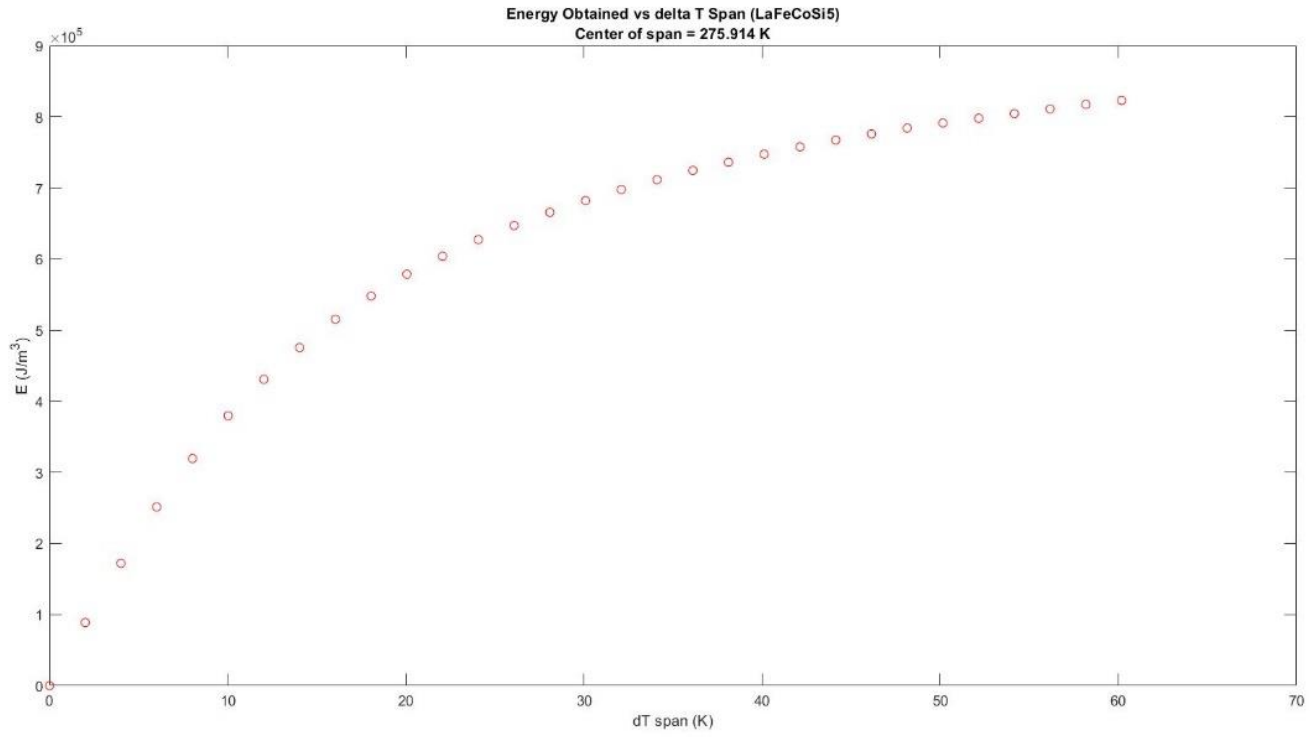


Centered around the Curie temperature:



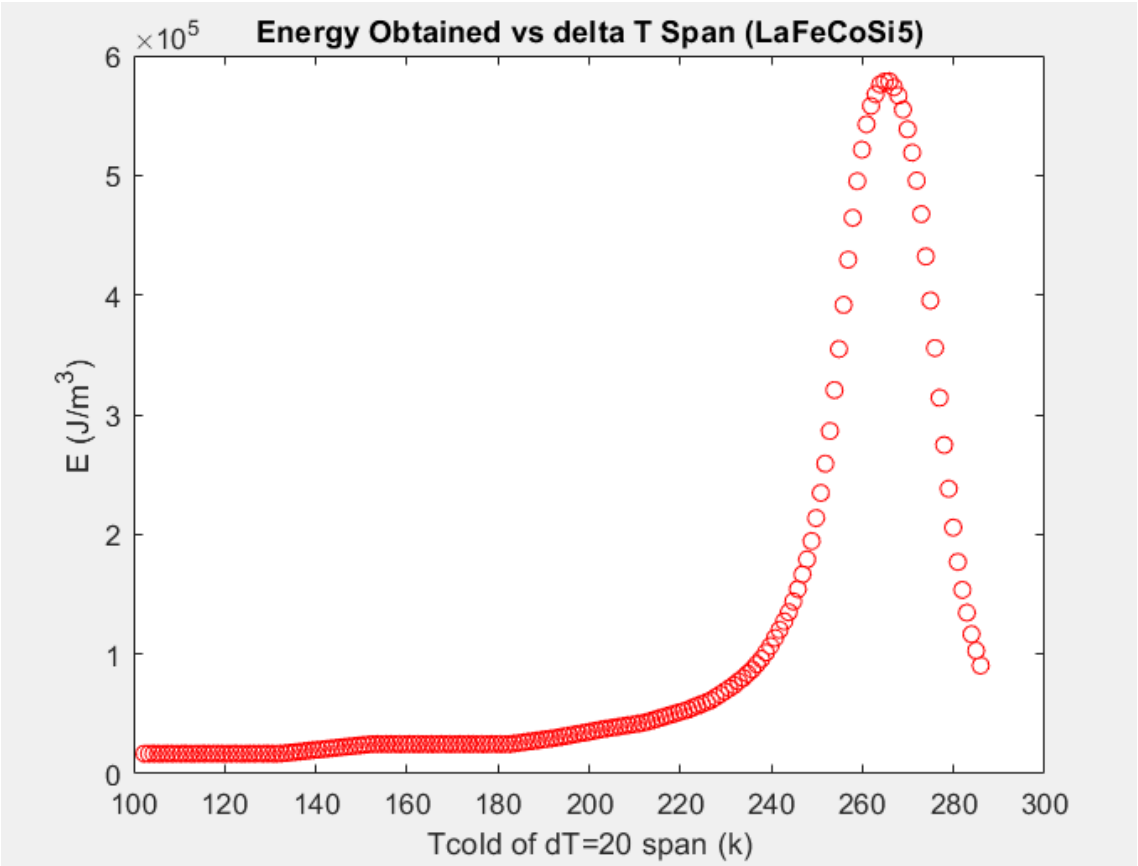


When energy versus temperature span is considered, the result is the following:

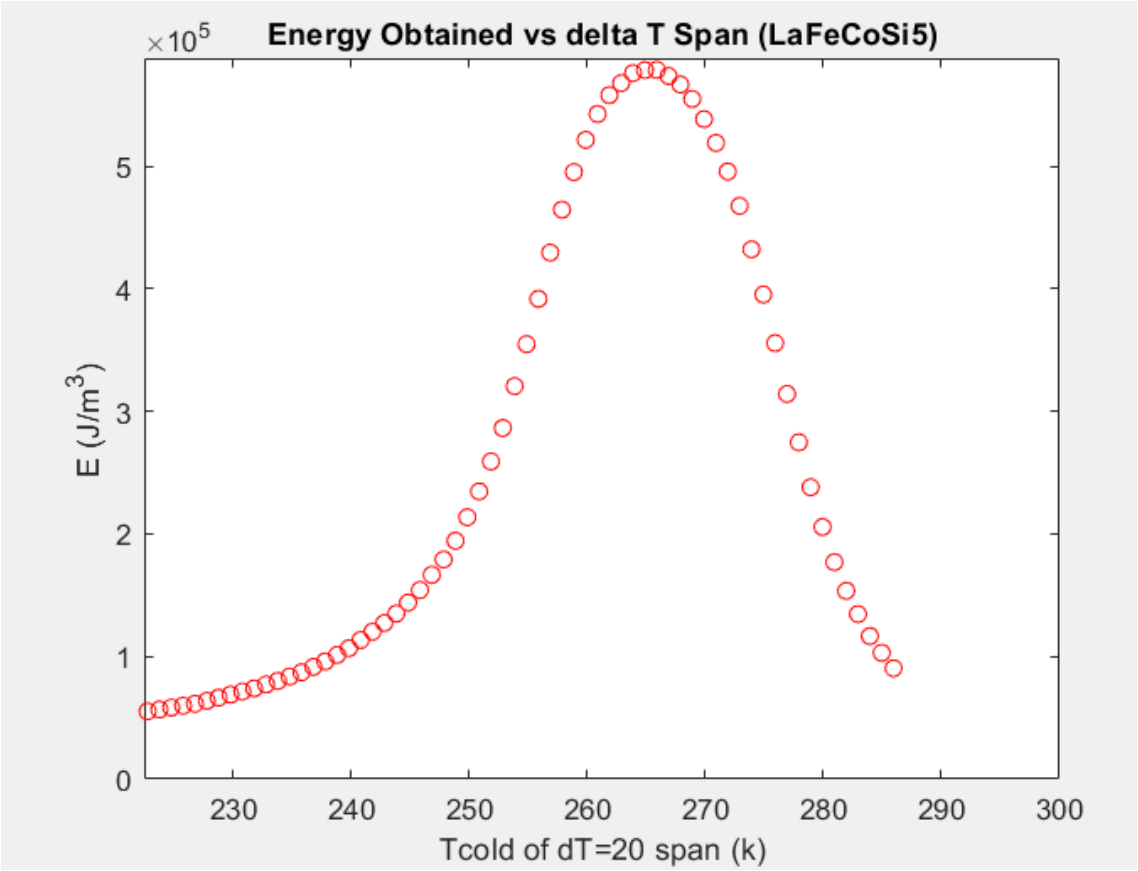


The same tendency towards a lower slope for larger temperatures spans can be seen as it was the case for Gadolinium. The result of 5.784*10^5 J/m^3 can also be seen for 20k span.

When considering the effect of a temperature span of 20k across the different temperature range, the next graphic is produced:



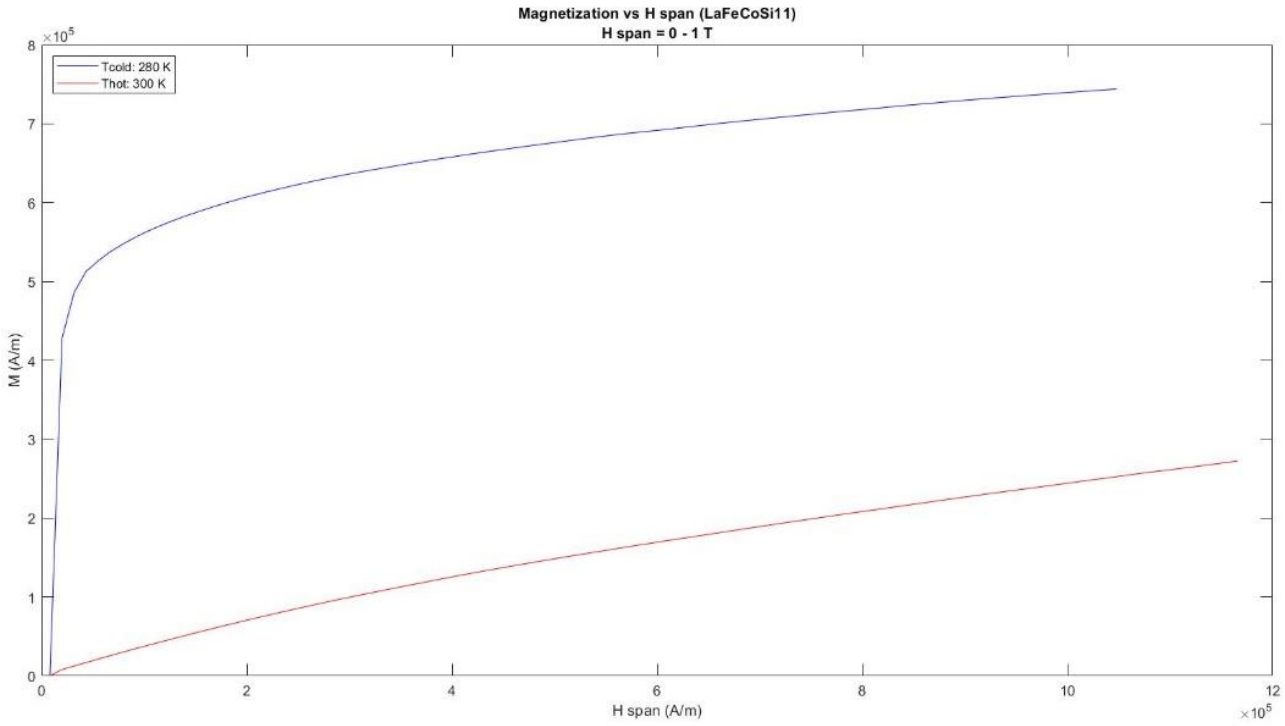
Similarly, to Gadolinium, the first range of temperatures can be left of consideration:

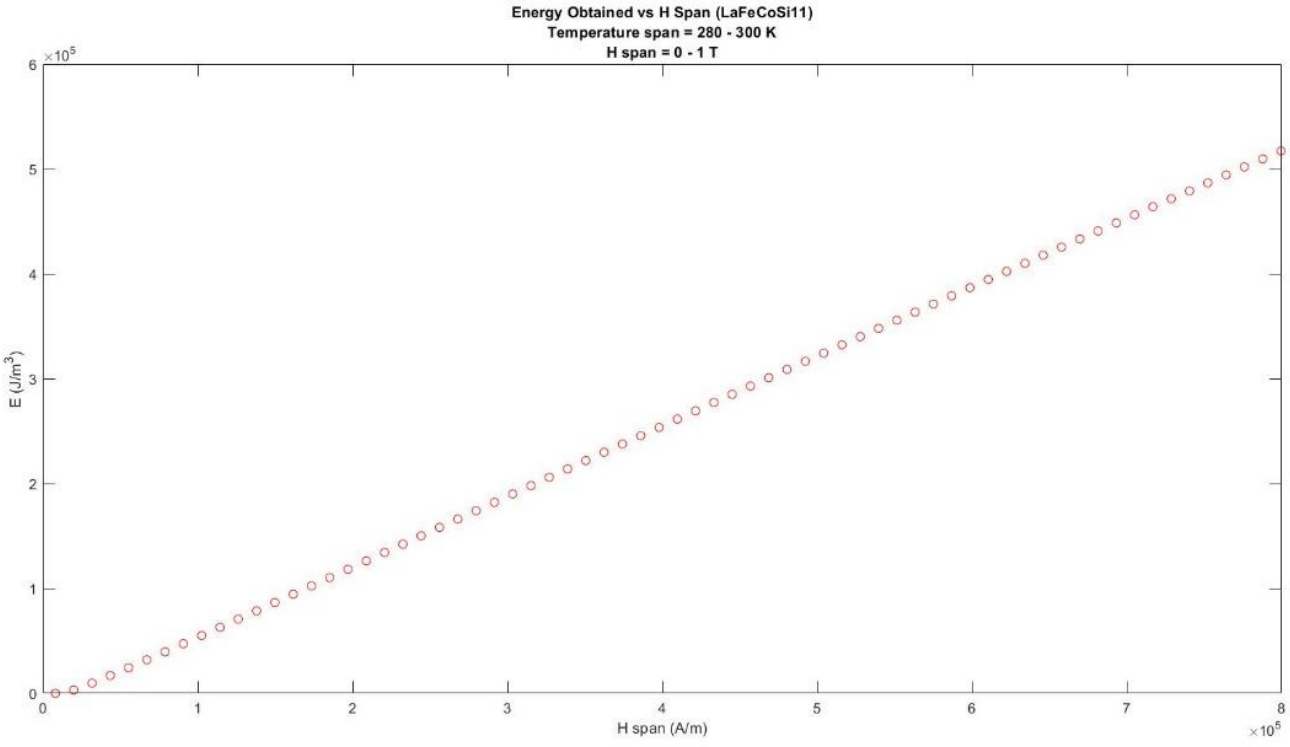


Note that the peak in energy harvesting is for cold temperature of around 265k, meaning a center of span temperature of 275k, around Tcurie.

LaFeCoSi11:

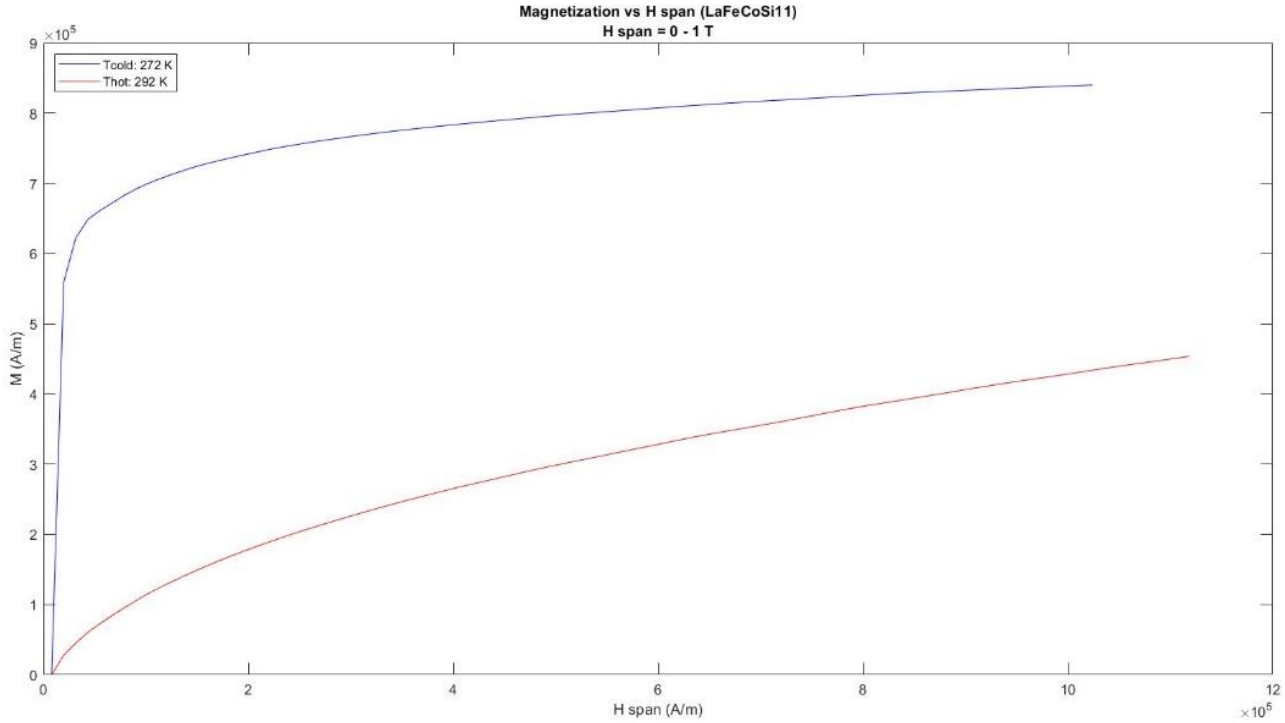
For LaFeCoSi5 in the same ranges as before and considering the same parameters we obtain the following:

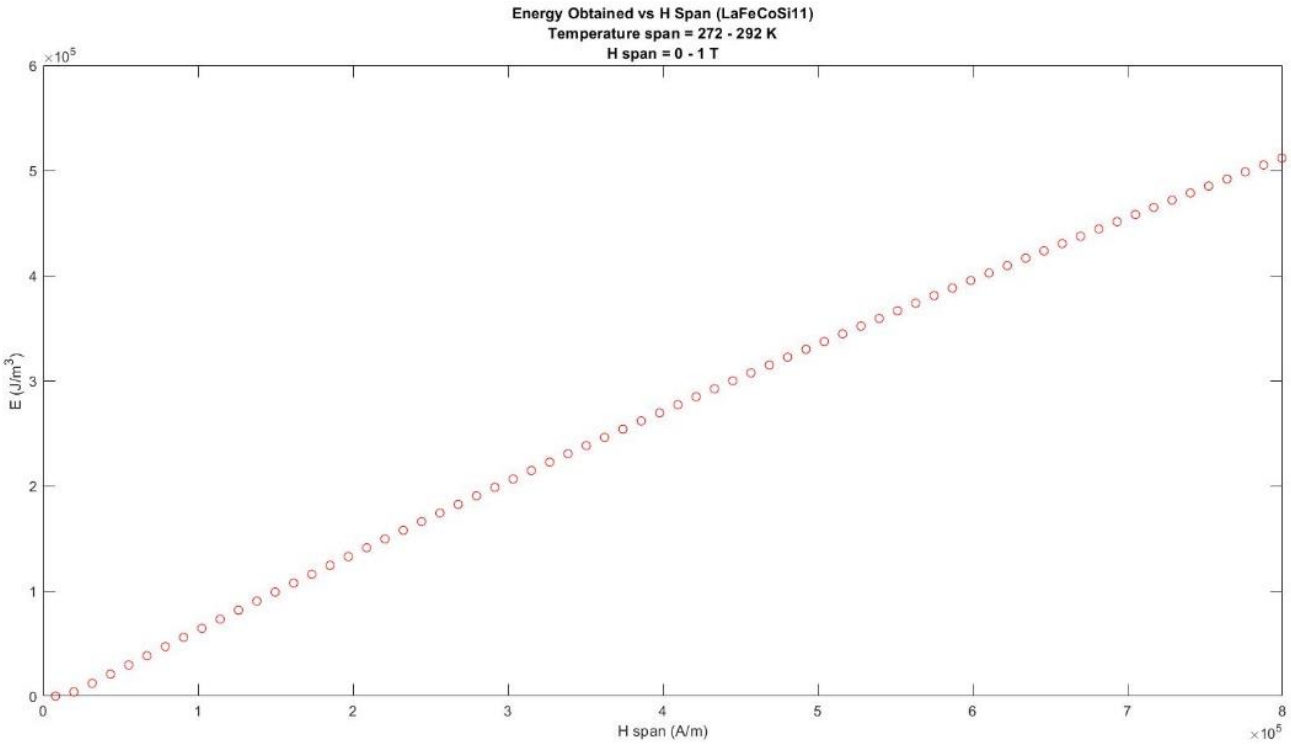

Magnetization vs H span (LaFeCoSi11)
H span = 0 - 1 T
Tcold: 280 K
Thot: 300 K
M (A/m)
H span (A/m)
×10^5


Energy Obtained vs H Span (LaFeCoSi11)
Temperature span = 280 - 300 K
H span = 0 - 1 T
E (J/m^3)
H span (A/m)
×10^5

The maximum energy per volume for a field span of 0T to 1T is 5.171*10^5J/m^3.

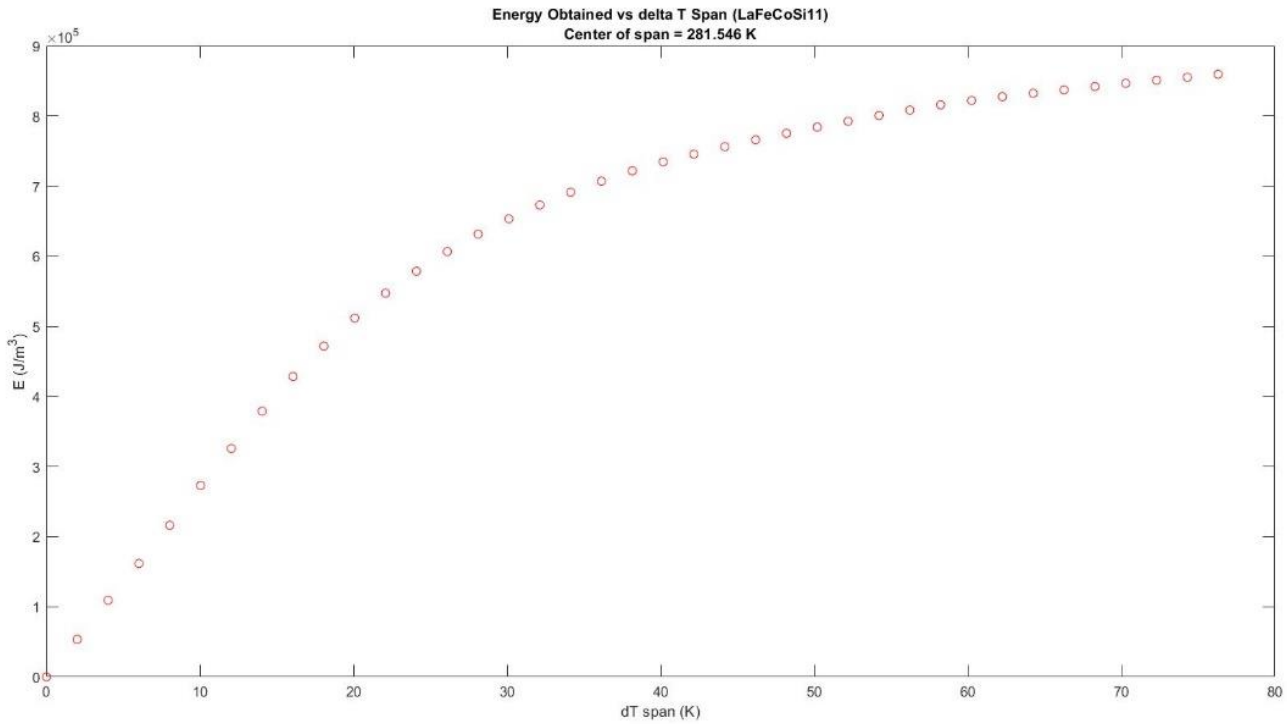
If we change the working temperatures to be Tcold = 275.5k and Thot = 295.5k the central temperature of the span will be near to the Curie temperature, improving the energy output:





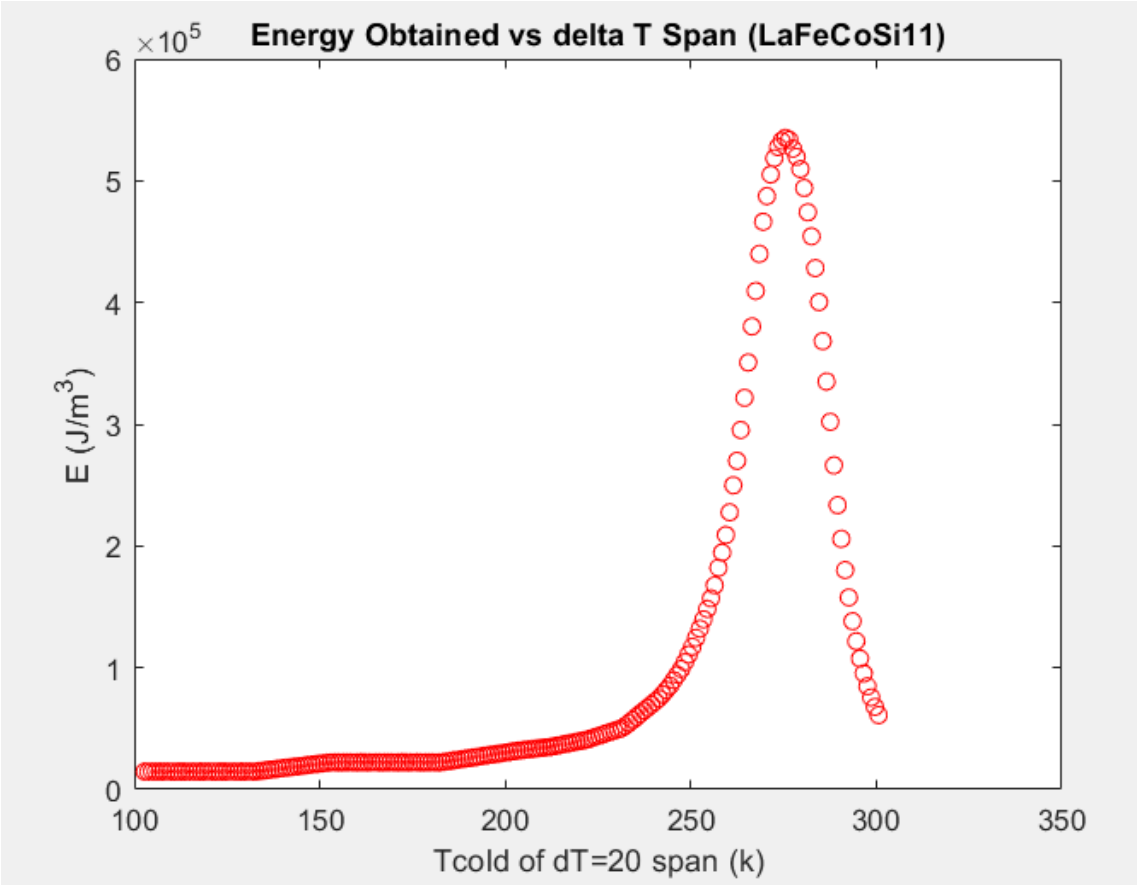
Now the output energy per volume is maximized to 5.426*10^3 J/m^3.

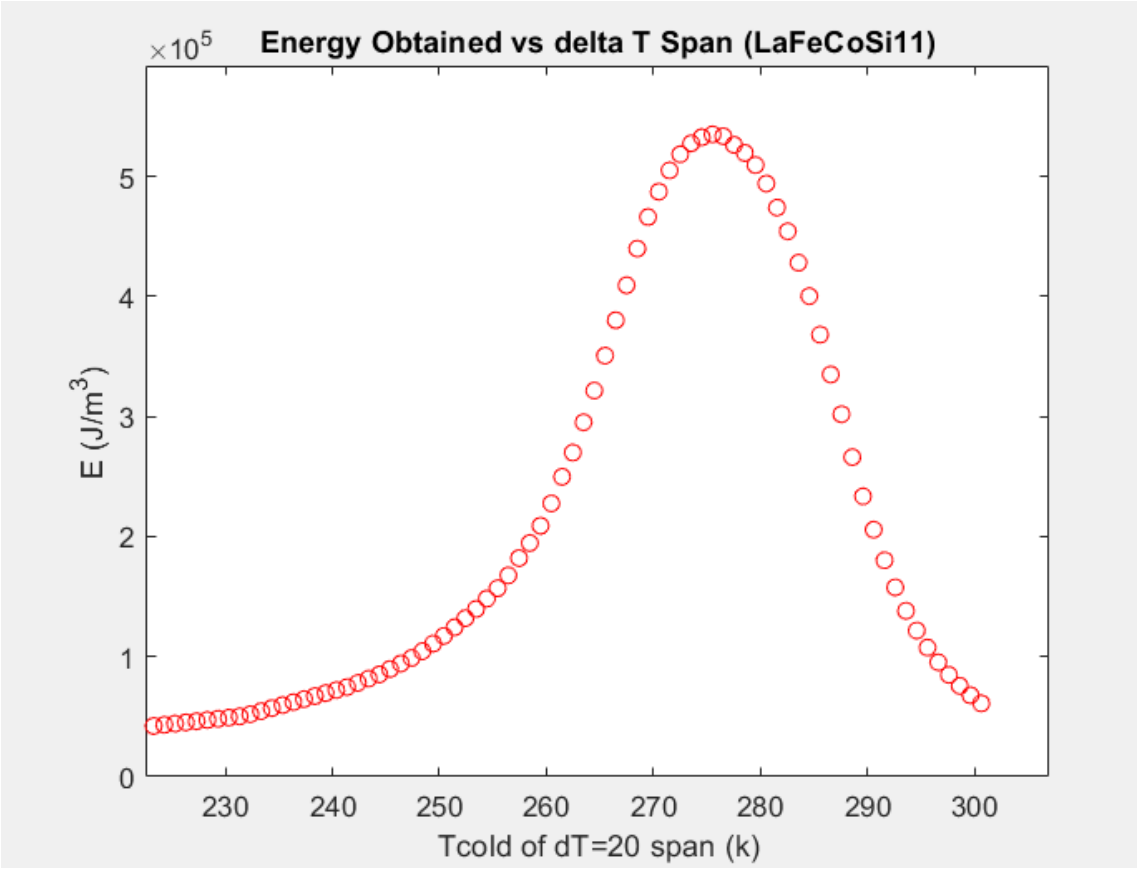
As for energy dependency on temperature span, it is illustrated in the following figure:



The same tendency for large temperatures spans is observed as in the other materials.

Considering a 20k span:



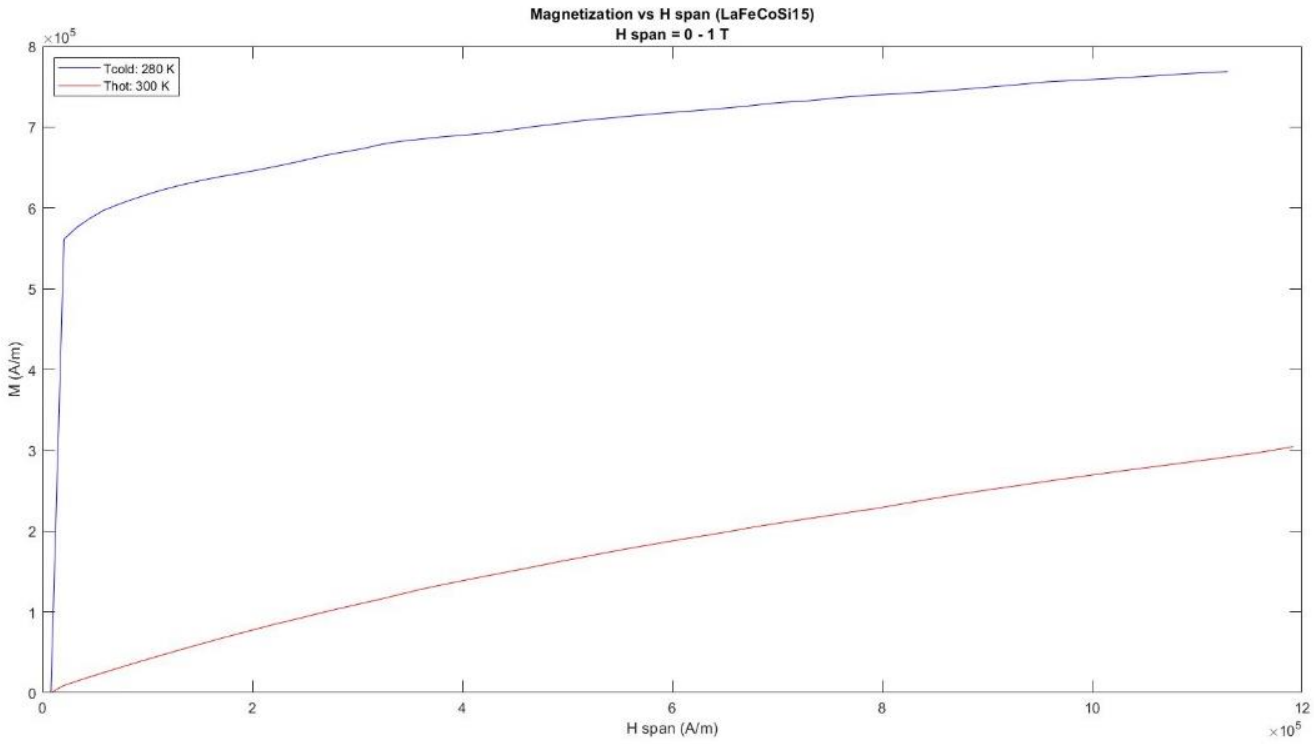


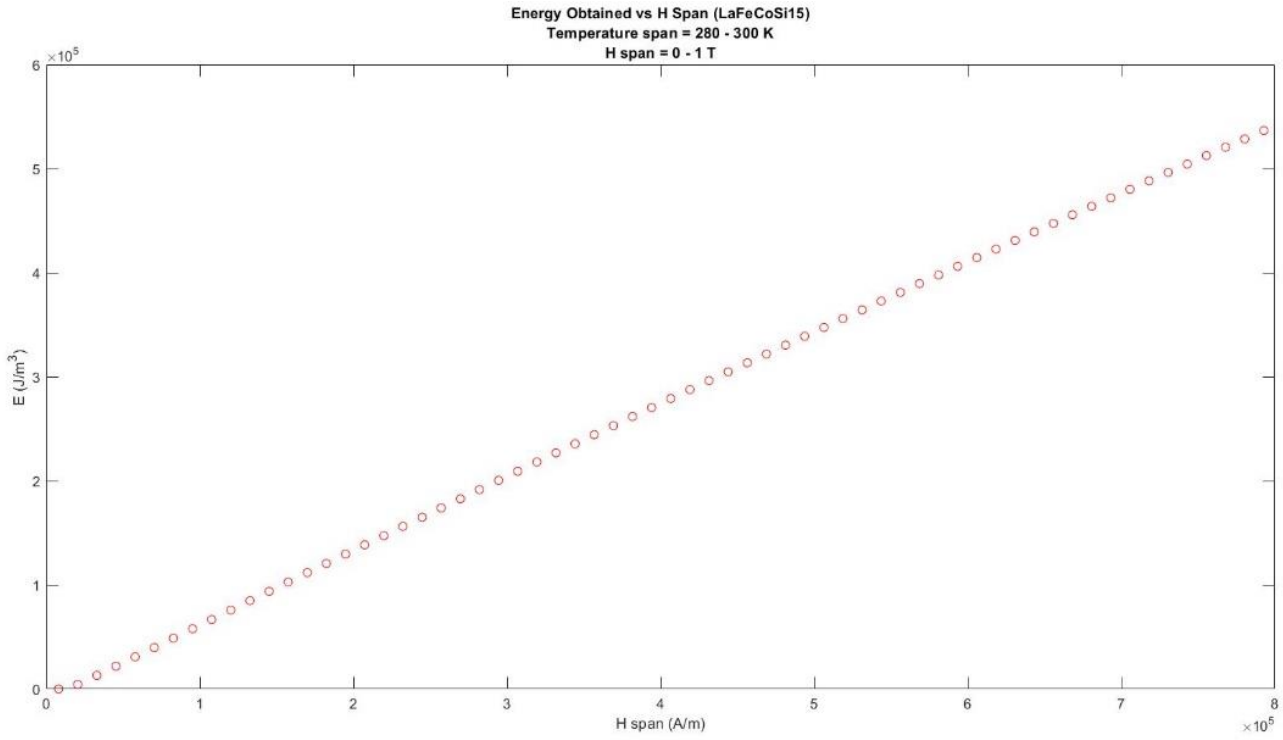
The maximum peak in energy production is 5.352*10^5 J/m^3 for a cold temperature of

275.5k, a center of span temperature (indicating near Curie temperature) of 285,5k.

LaFeCoSi15:

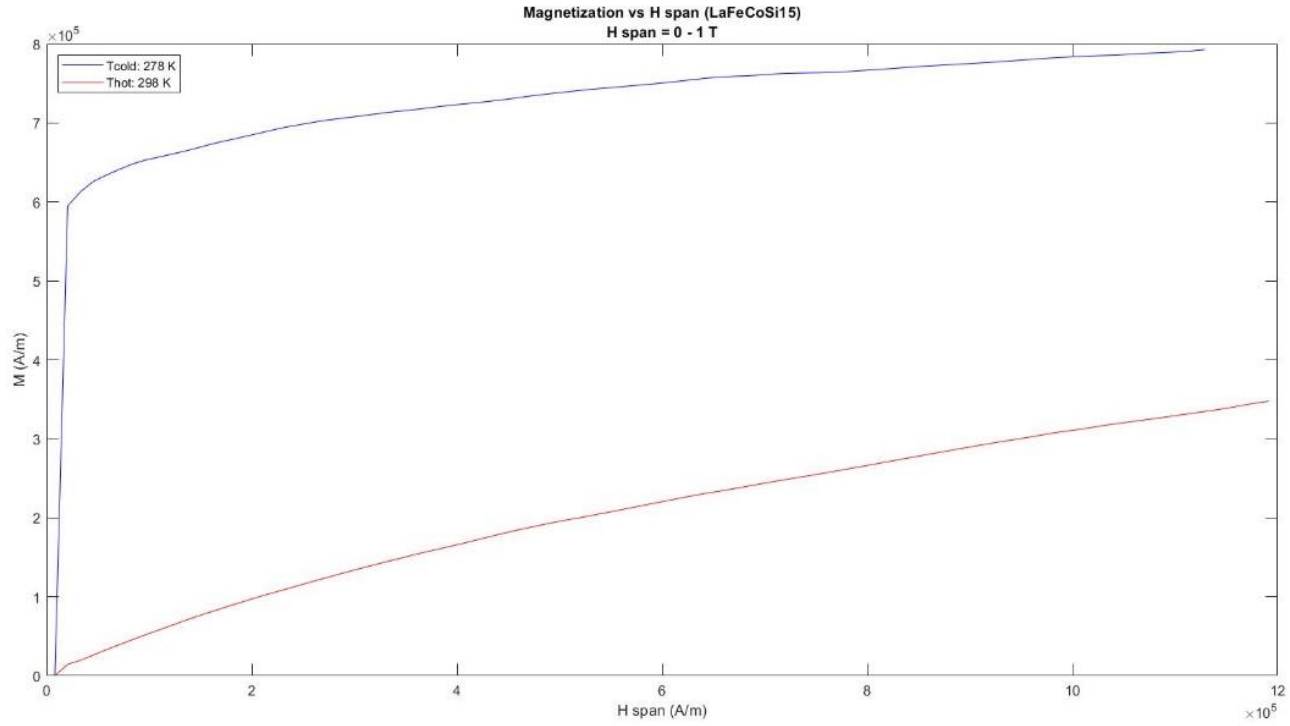
Same parameters as considered previously in other materials to proof differences and similarities between materials behaviors:

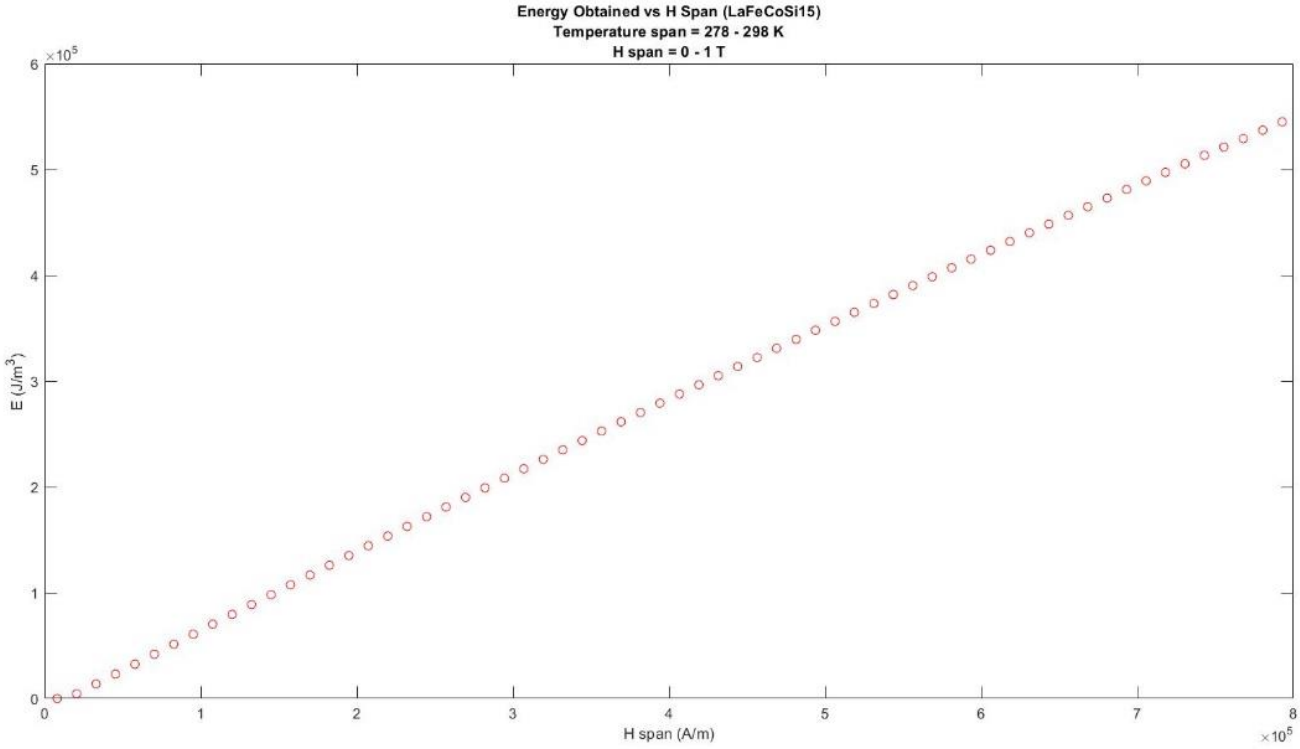




A linear dependency between energy and field span can be seen for the given field spans and working temperatures. The maximum energy obtained for these ranges is 5.367*10^5 J/m^3.

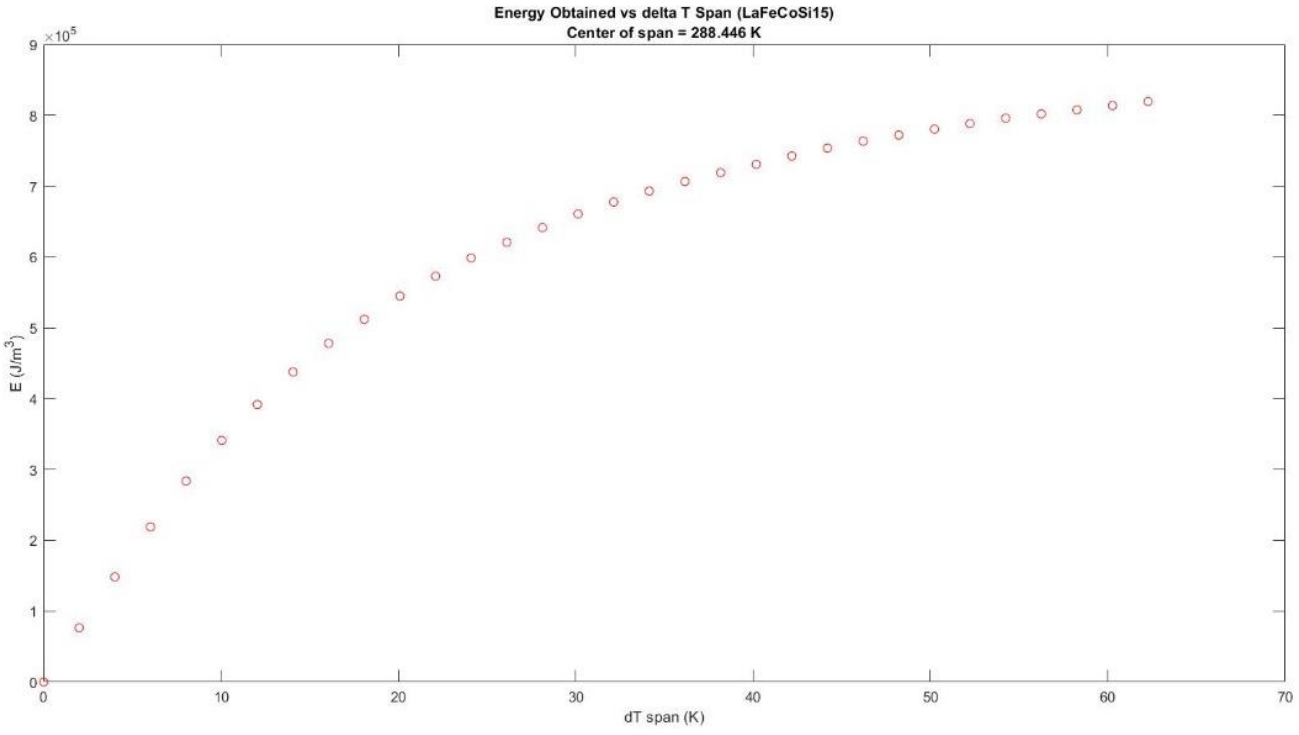
However, if a 20k temperature span centered on the Curie temperature (288 K) is considered, the results are as follow:



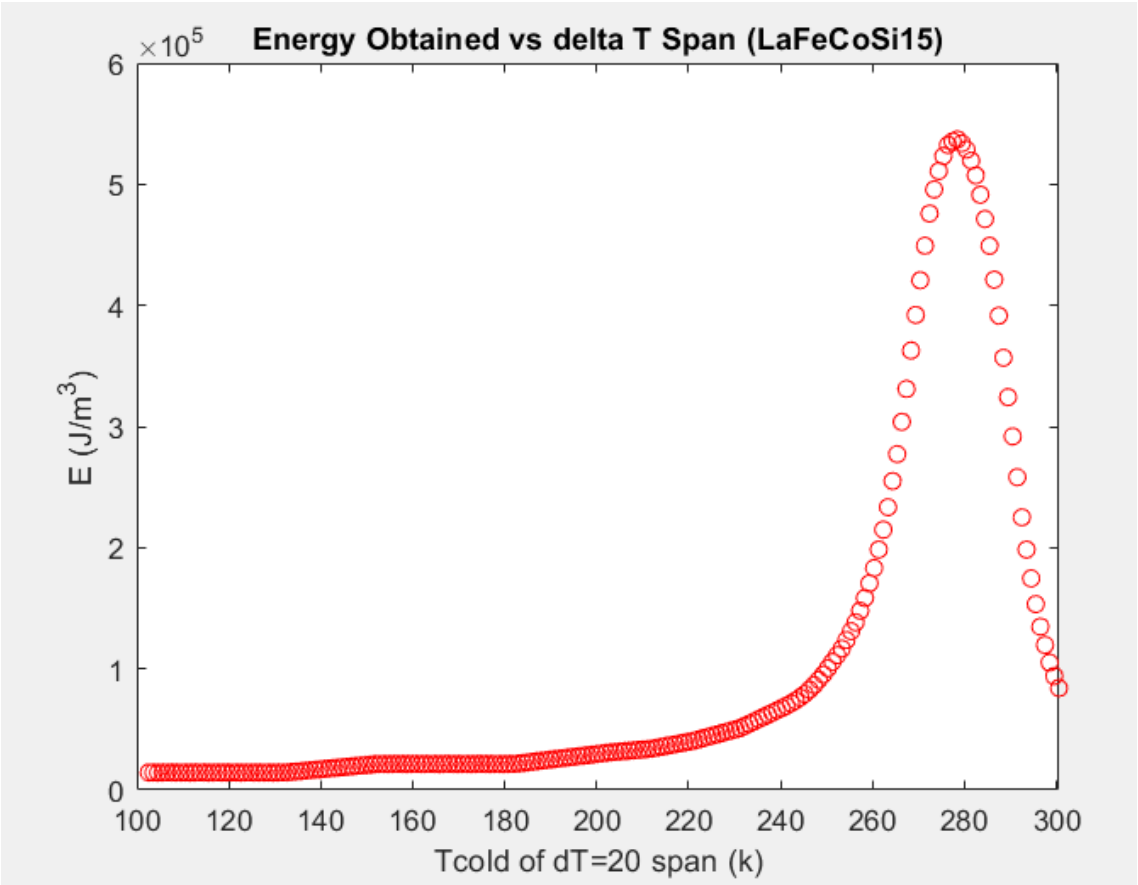


This way, the resulting maximum energy is risen up to 5.45*10^5 J/m^3.

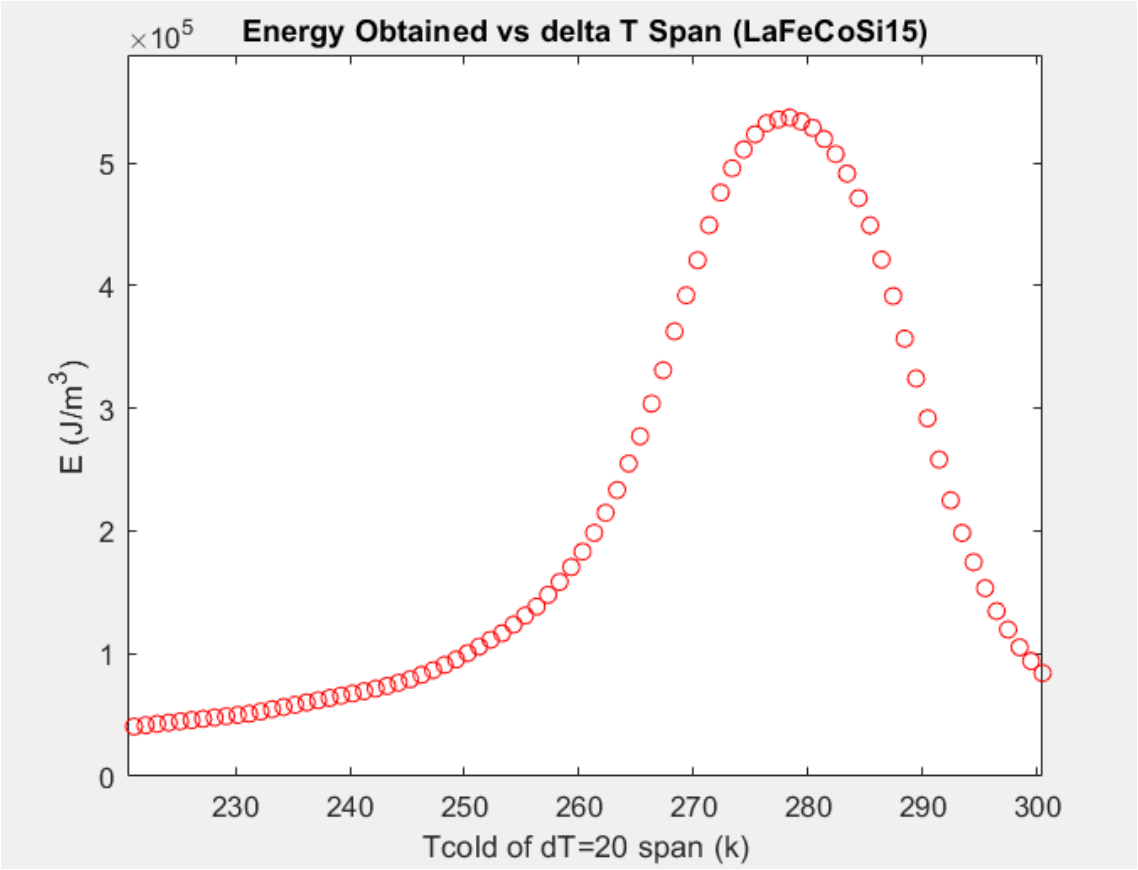
When considering temperatures spans, the following graphic is used:



It presents the same tendencies as described for other material. For a movable 20k span, the graph is the following:



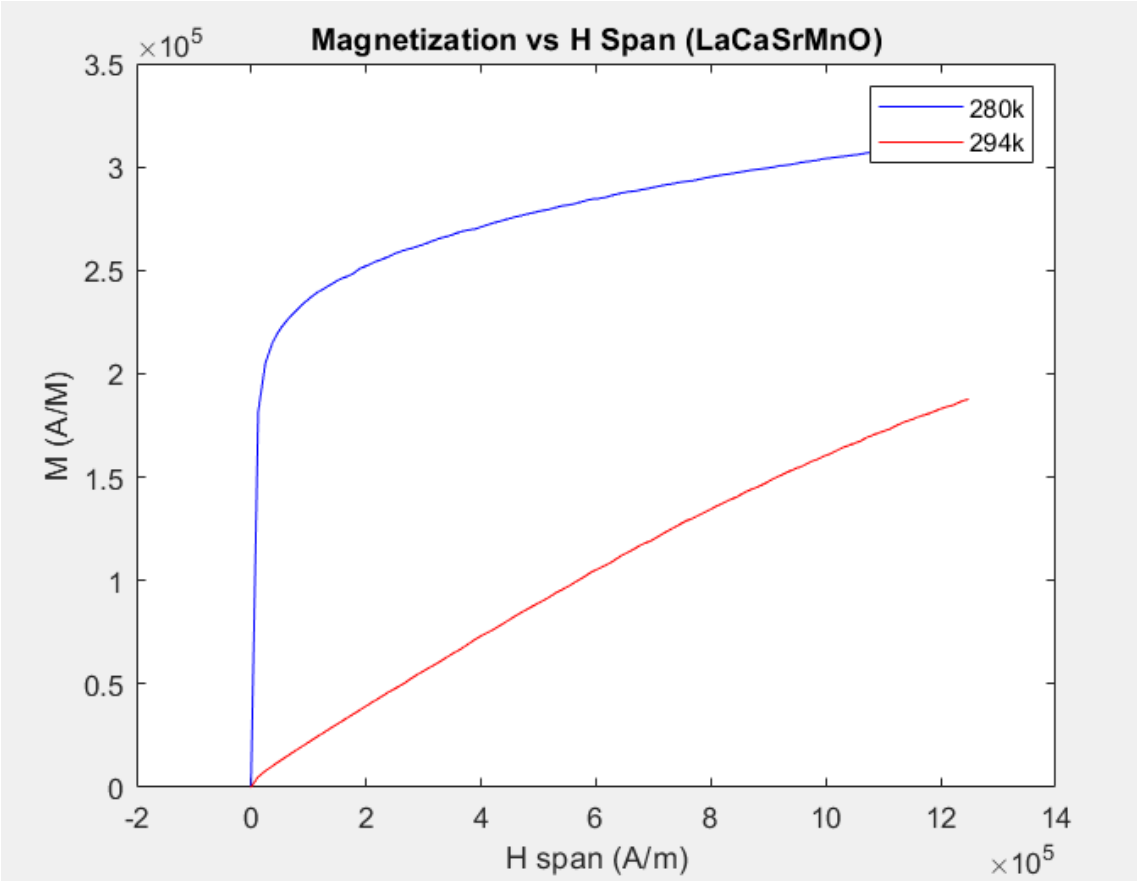
The

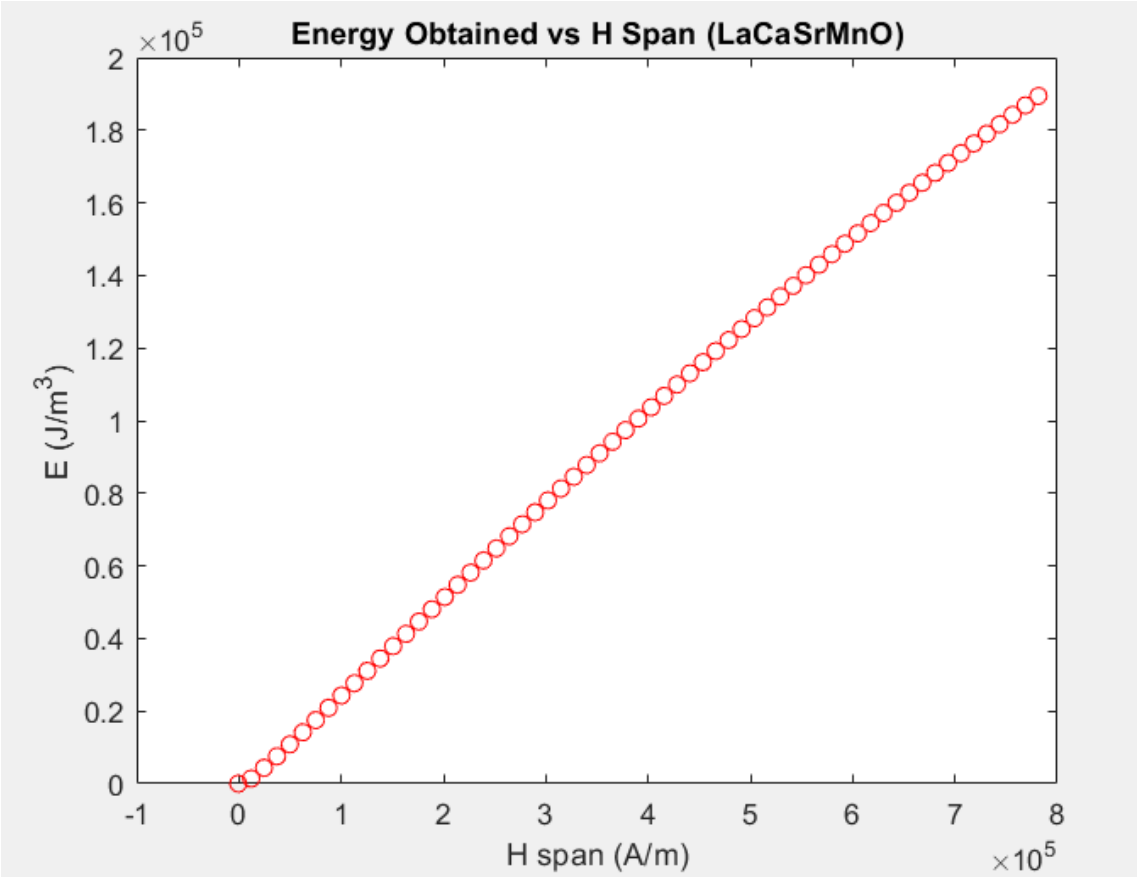


The peak in energy is reached at Tcold = 278.4k, being the center of the span 288,4k where the resulting energy per volume harvested is 5.371*10^5 J/m^3.

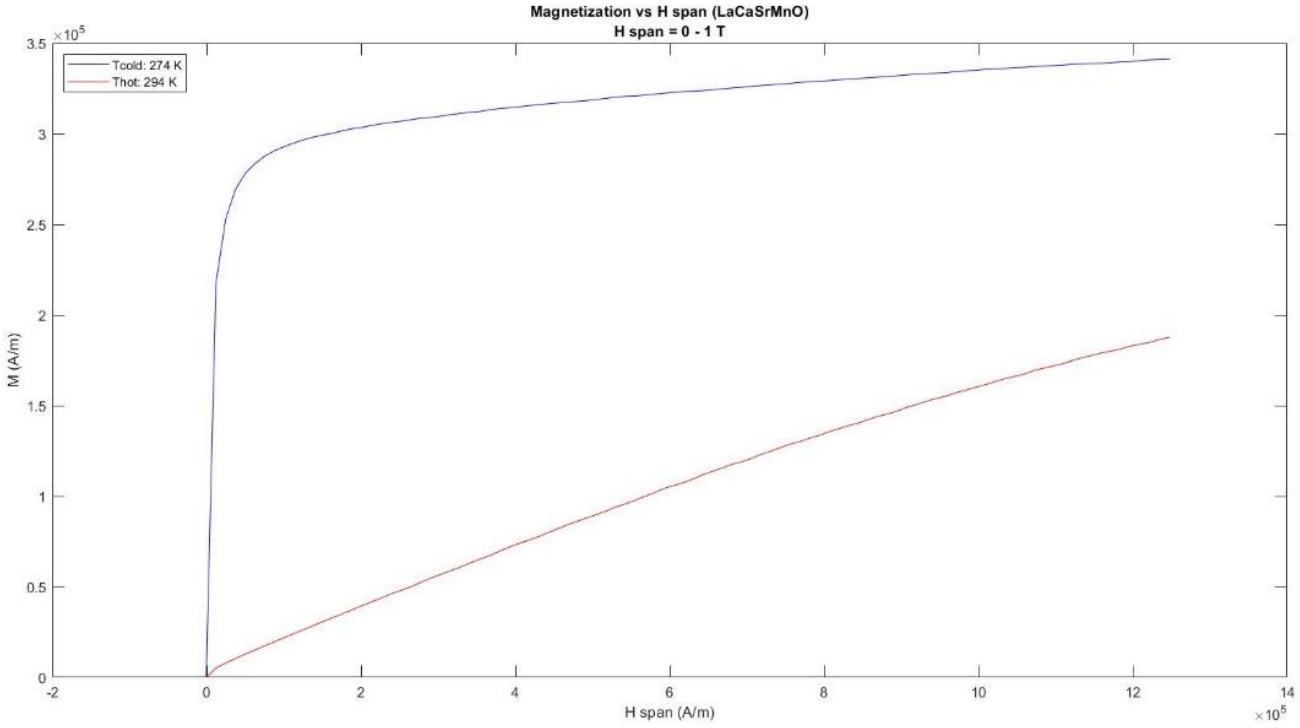
LaCaSrMnO:

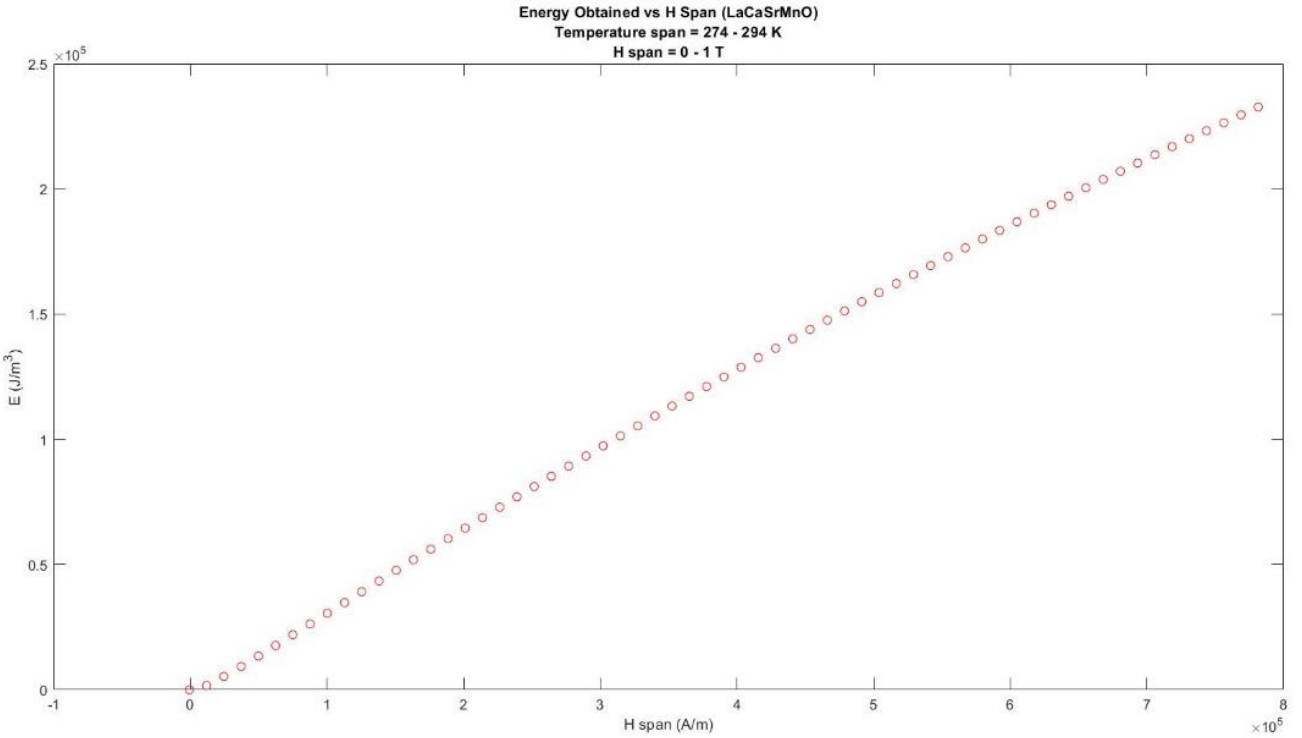
Studying LaCaSrMnO with the same parameters as the previous materials, the next figures are obtained:


Magnetization vs H Span (LaCaSrMnO)
×10^5
3.5
3
2.5
2
1.5
1
0.5
0
M (A/M)
280k
294k
-2
0
2
4
6
8
10
12
14
H span (A/m)
×10^5


Energy Obtained vs H Span (LaCaSrMnO)
×10^5
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
E (J/m^3)
-1
0
1
2
3
4
5
6
7
8
H span (A/m)
×10^5

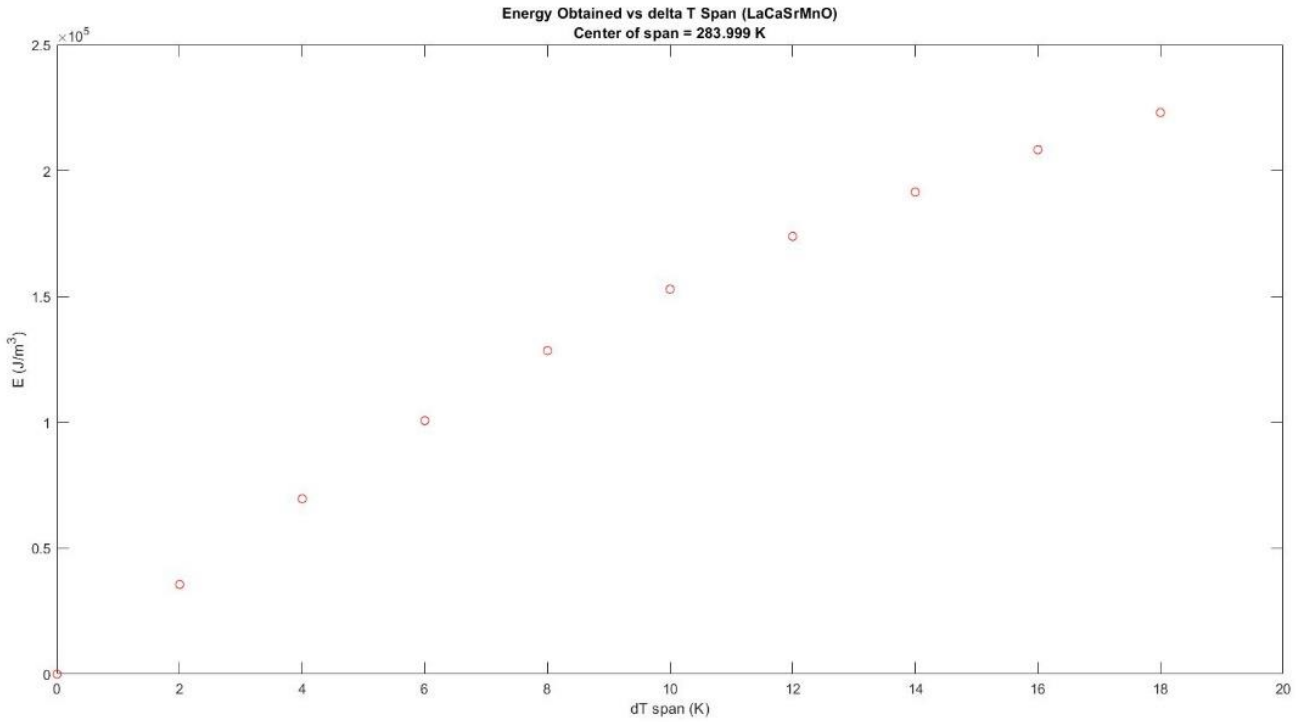
The maximum energy for a field span from 0T to 1T is then 1.894 J/m^3. If the working temperatures are changed to 274k and 294k, the graphs change:





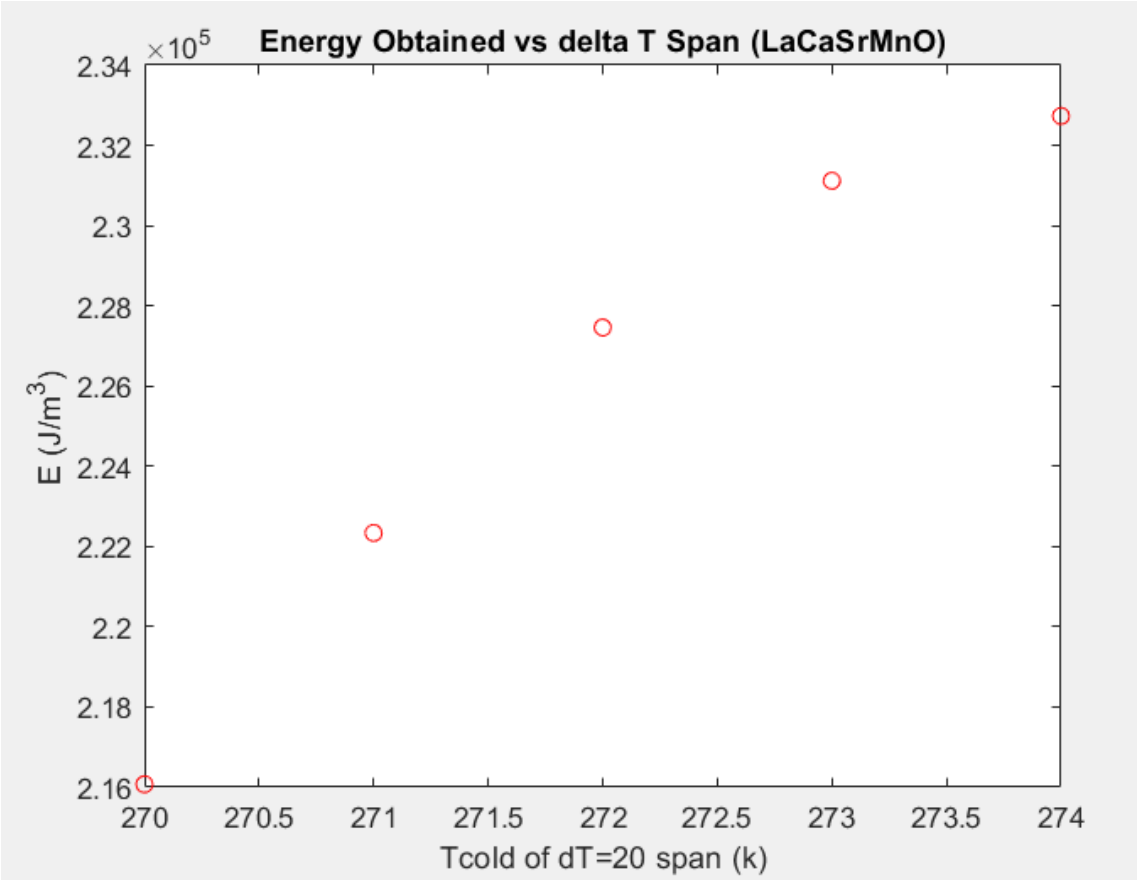
Now the energy obtained is 2.327*10^5 J/m^3. The ranges for this material go up to 294k because of the limited existing data for it. This is partially responsible for the lower energy output.

In order to take into account the influence of temperature span, the next graphic was produced:



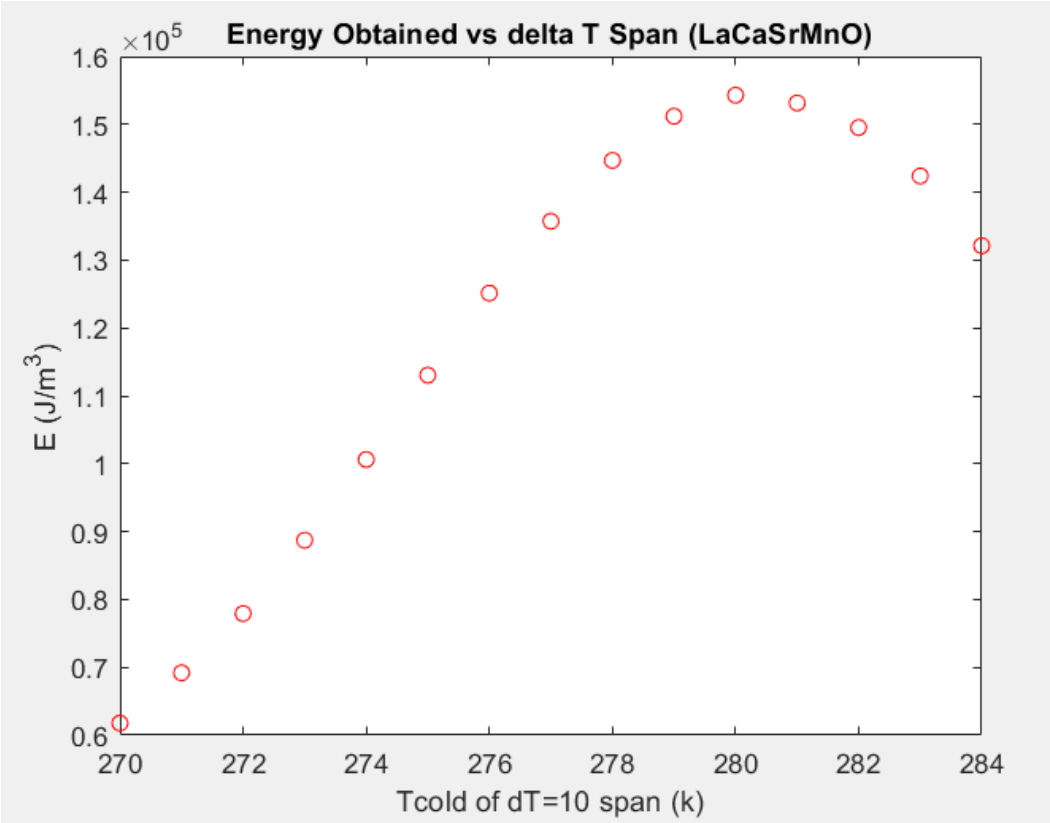
Note that the short curve is due to the short range of temperatures of the data.

Now the 20k span is considered along the temperature span:

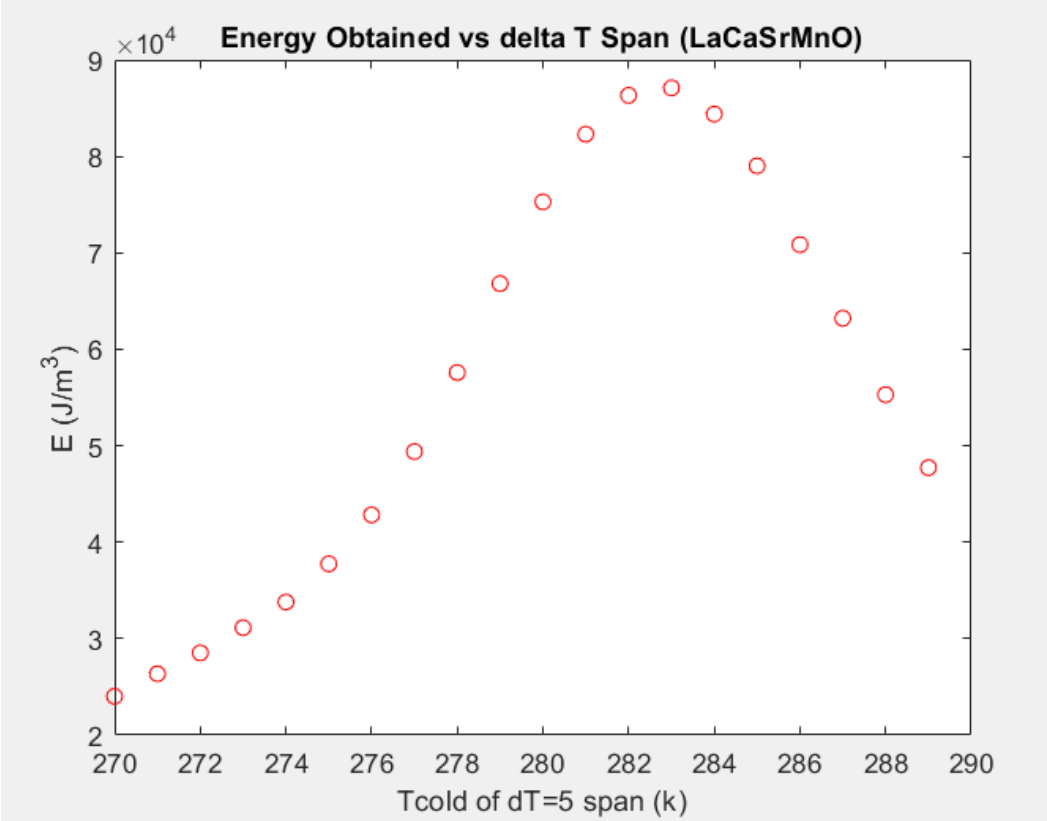


Note that the expected parabola is not obtained for dT=20, but for lower

dT values as the temperature data range of this material is quite short. If for example, we use dT=10k:



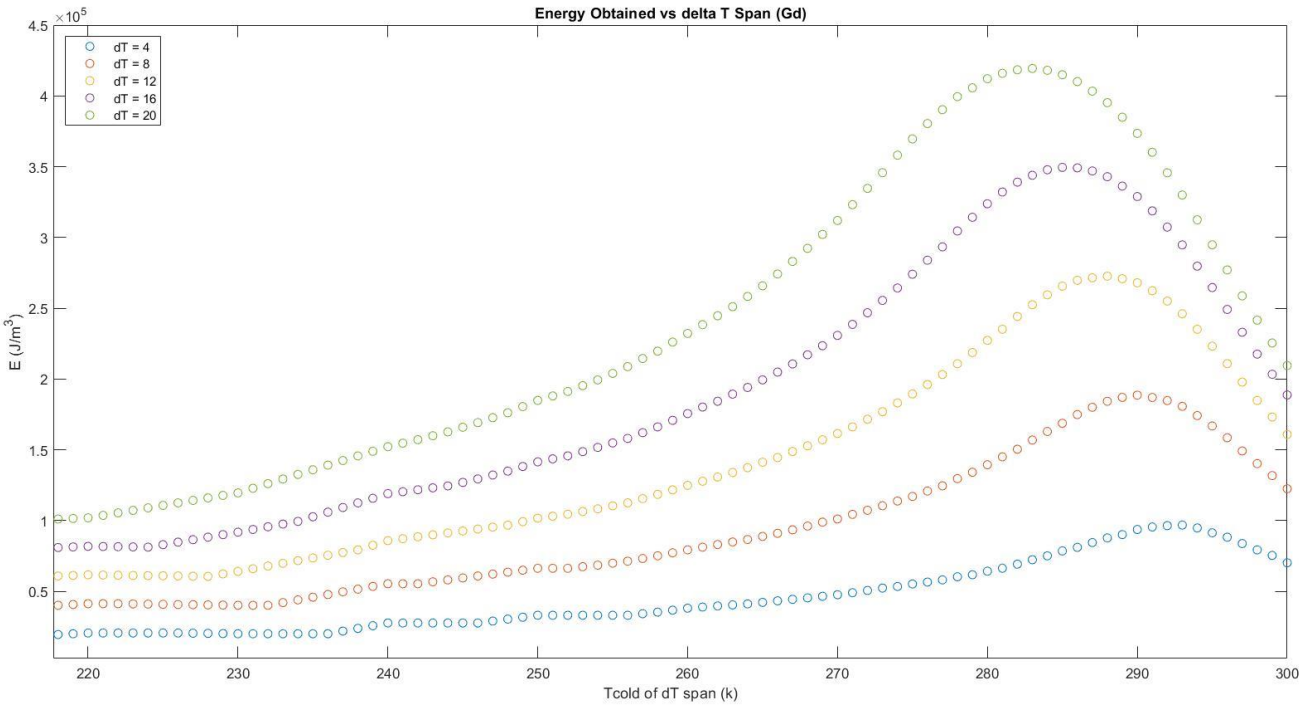
For dT=5k:



Now the expected parabola shape can clearly be seen.

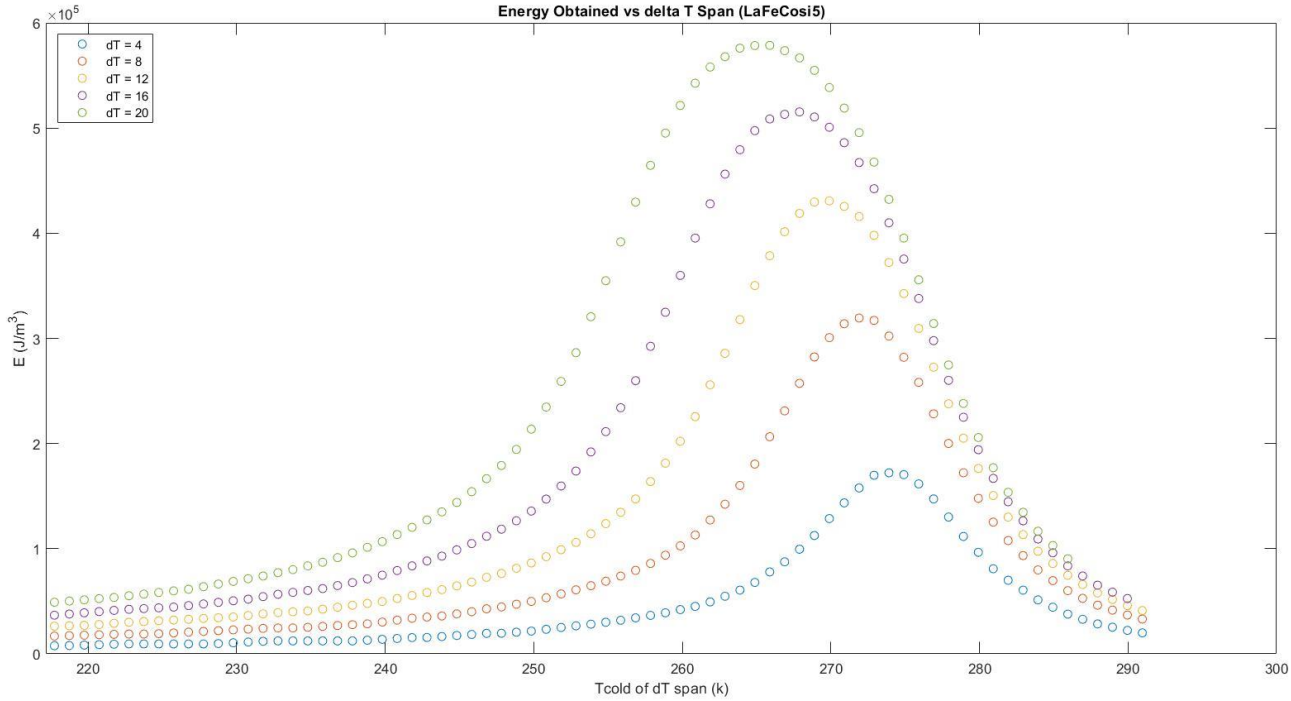
APPENDIX B - PLOTTING THE MATERIALS TOGETHER AND PARMATERS CHANGES

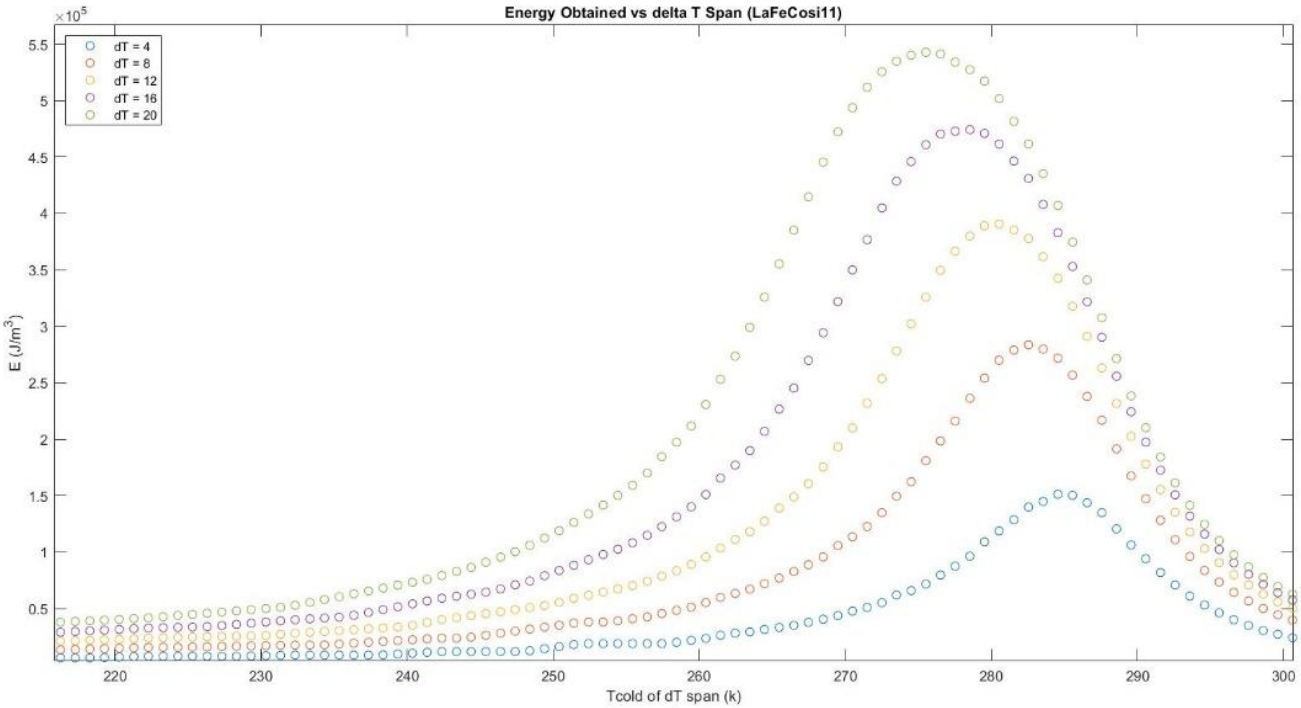
Now we consider Gd with a varying temperature difference of dT = {4,8,12,16,20}, the field span is still 0 to 1 T:

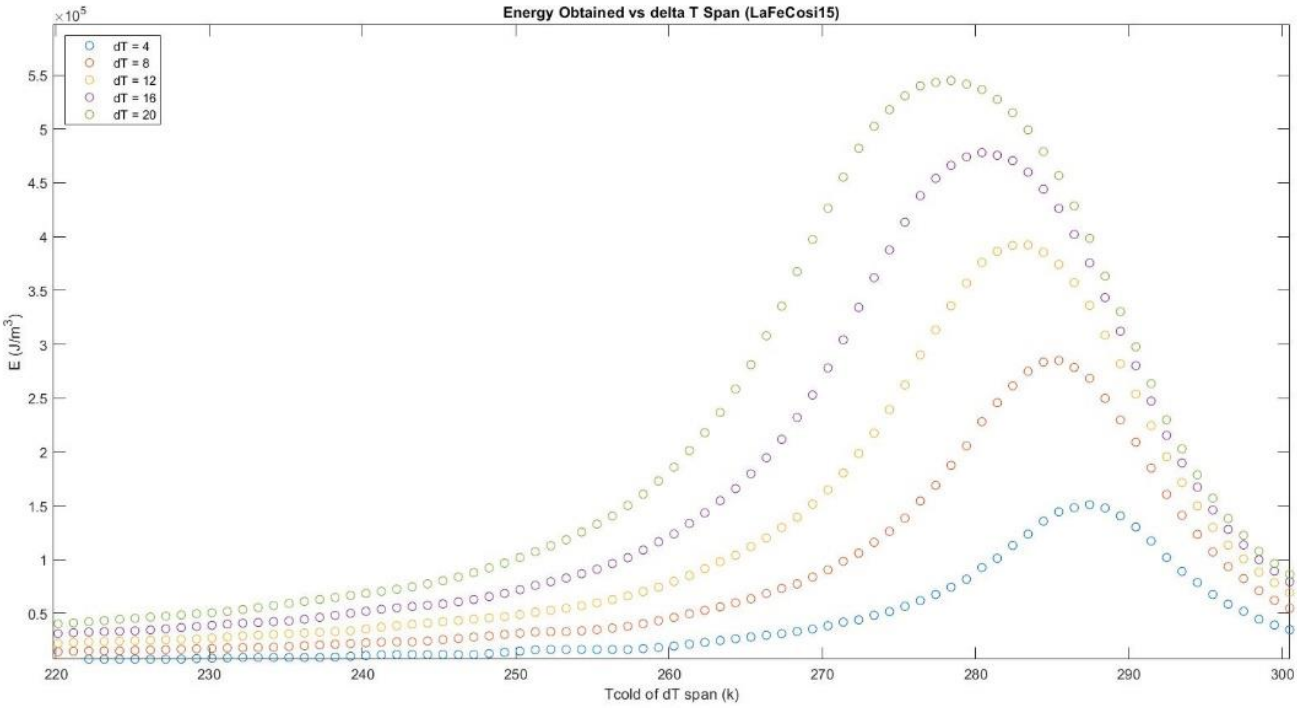


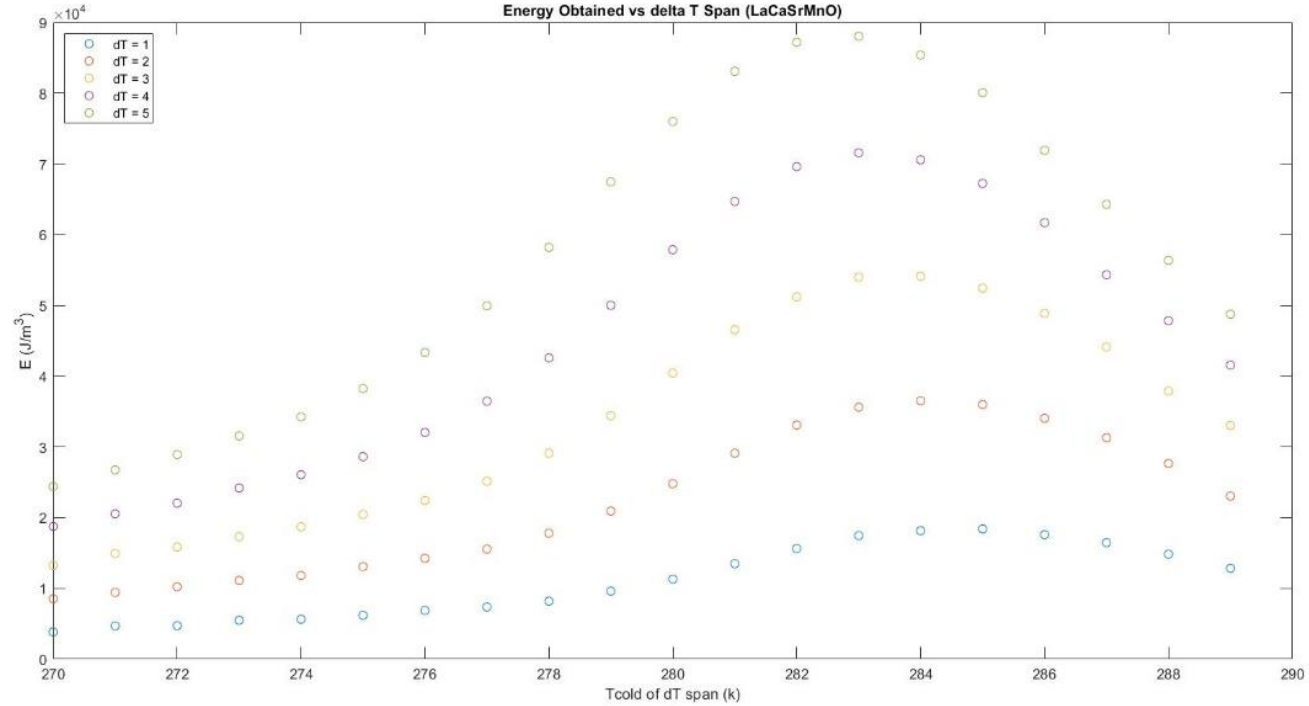
It can be seen that increasing the temperatures span improves energy output while leading the peak of the curve towards cooler temperatures. This is important when considering possible prototypes: not only the efficiency will increase but lowering the operating temperatures at which the peak is found permits the device to work near ambient temperature, resulting in easier and cheaper implementation.

A similar trend can be seen for other materials:

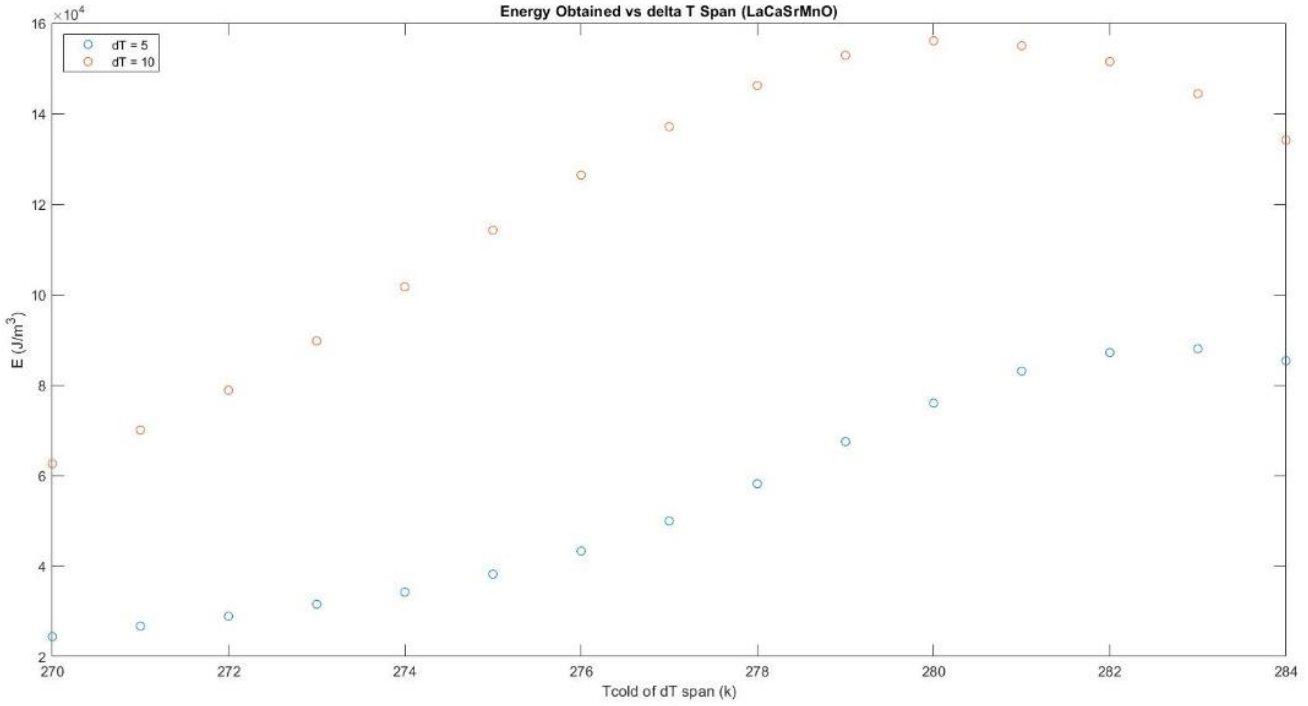

Energy Obtained vs delta T Span (LaFeCosi5)
×10^5
dT = 4
dT = 8
dT = 12
dT = 16
dT = 20
E (J/m^3)
Tcold of dT span (k)

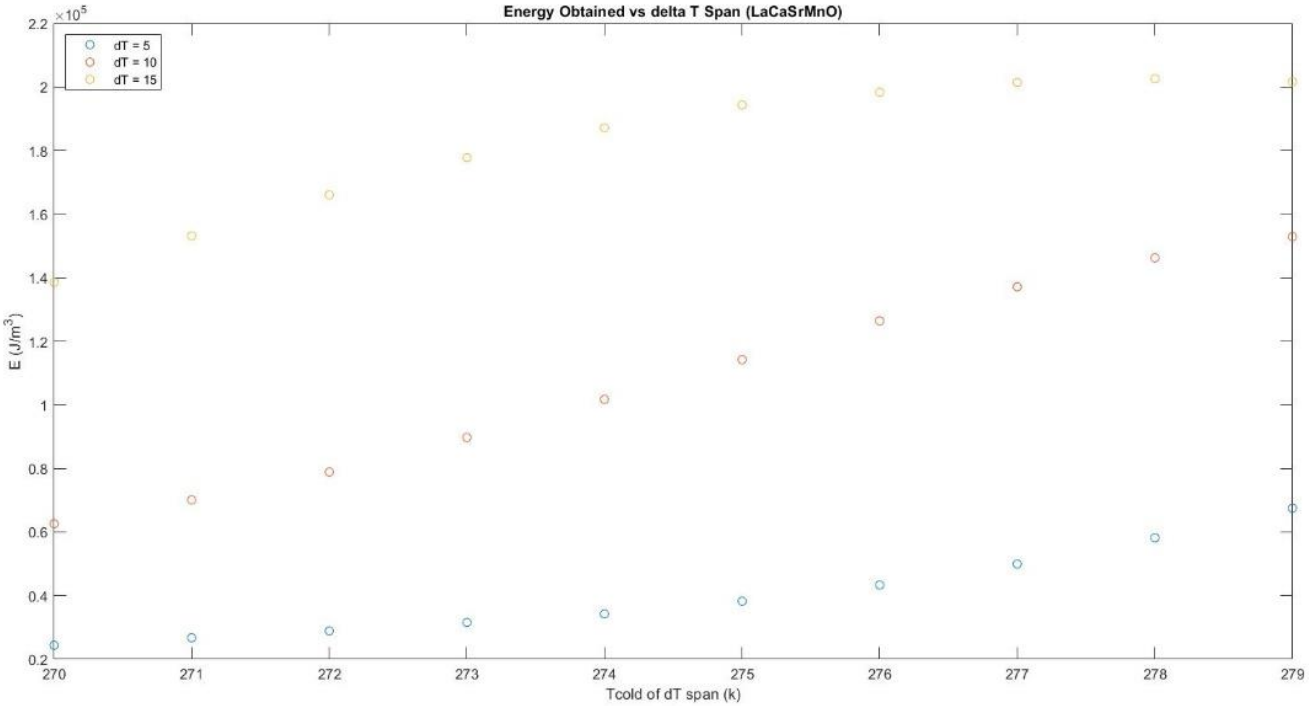

Energy Obtained vs delta T Span (LaFeCosi11)
×10^5
dT = 4
dT = 8
dT = 12
dT = 16
dT = 20
E (J/m^3)
Tcold of dT span (k)


Energy Obtained vs delta T Span (LaFeCosi15)
×10^5
dT = 4
dT = 8
dT = 12
dT = 16
dT = 20
E (J/m^3)
Tcold of dT span (k)

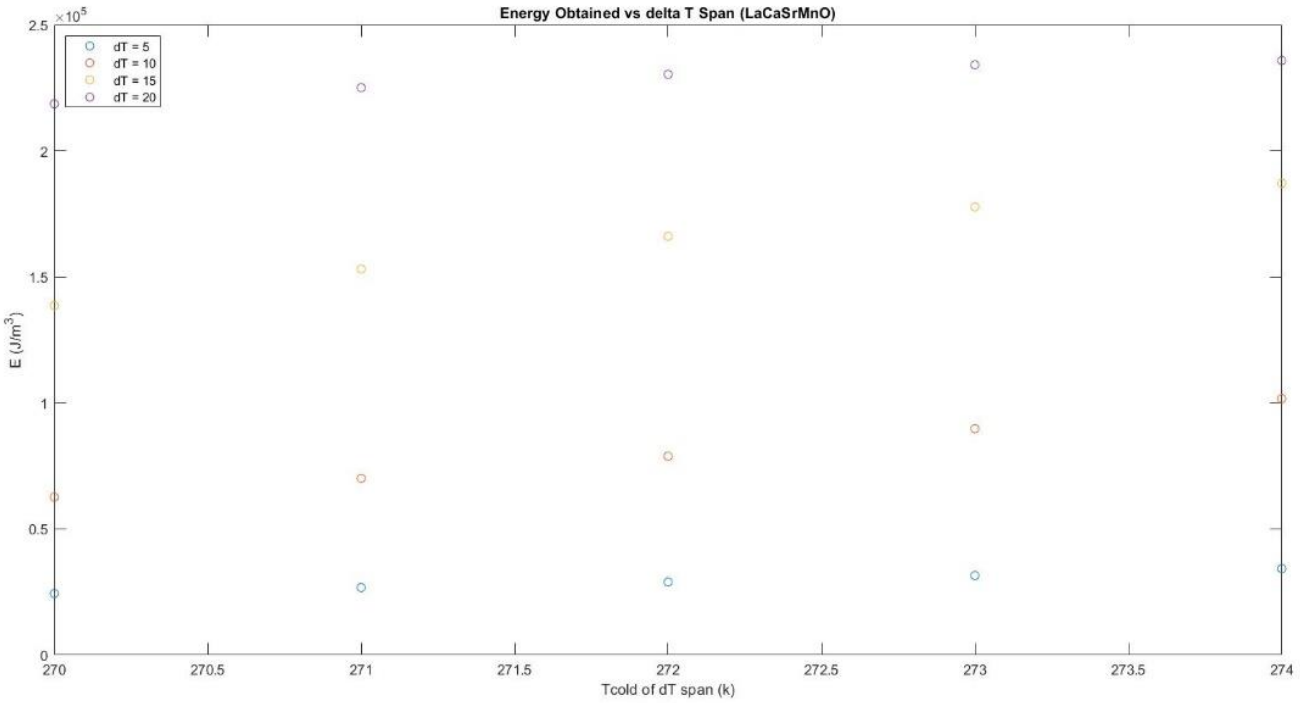

Energy Obtained vs delta T Span (LaCaSrMnO)
×10^4
dT = 1
dT = 2
dT = 3
dT = 4
dT = 5
E (J/m^3)
Tcold of dT span (k)

The same tendency is observed for LaCASrMnO, though the temperature span had to be decreased due to the limited data. Enlarging the spans results in graphics centering at lower temperatures to be able to adjust the whole span, but it is anyway to compare better with other materials in the ranges they have been studied:

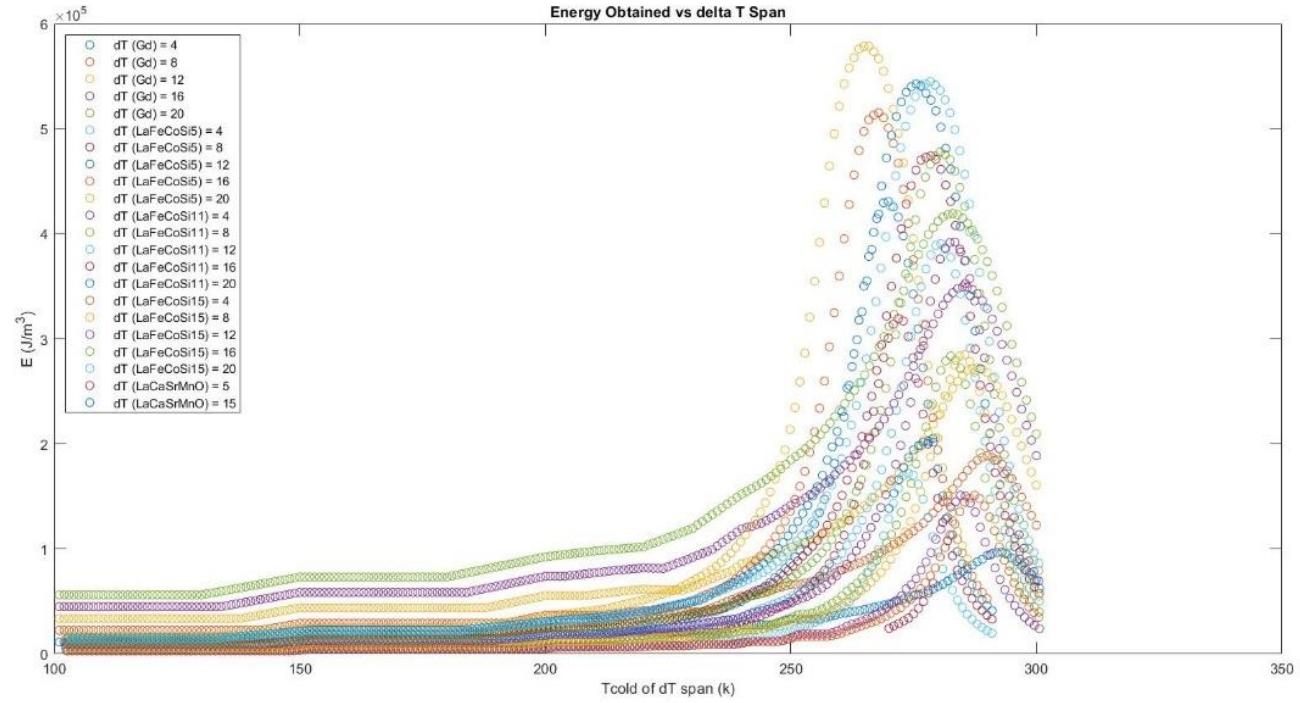


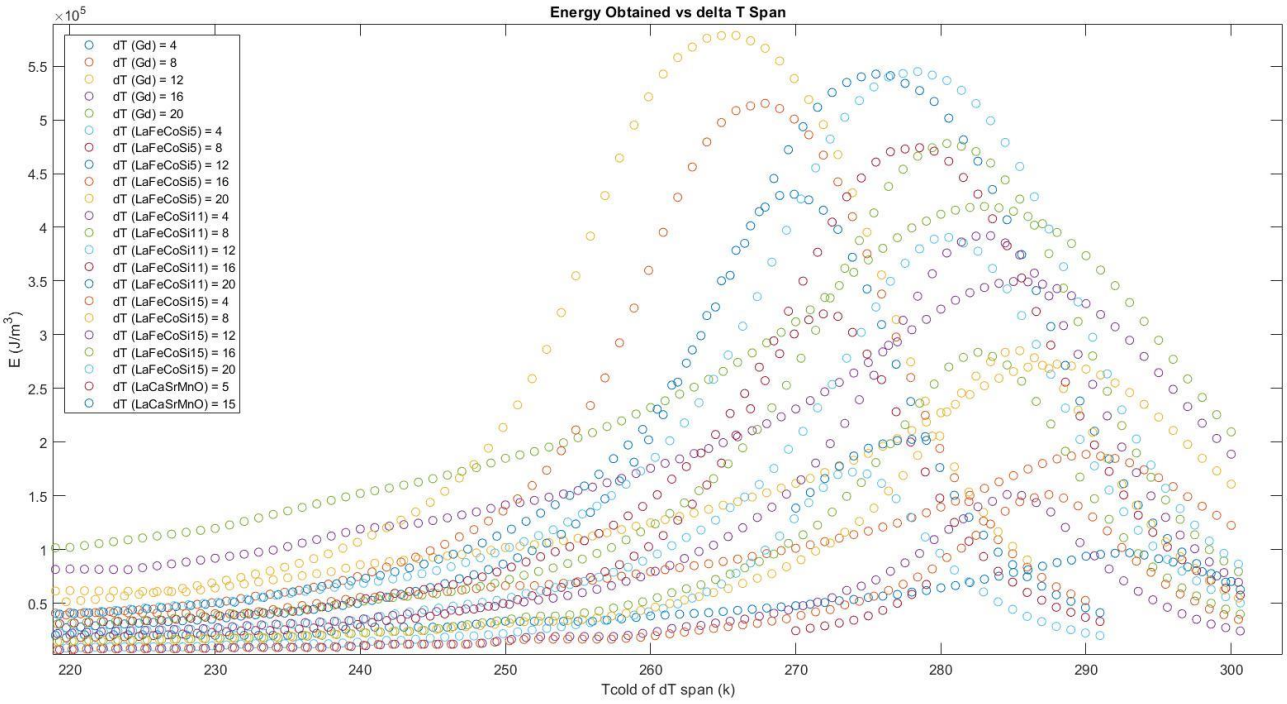


It is still possible to see a peak of 2.026*10^5 J/m^3 for dT = 15 k at a cold temperature of 278k. For larger ranges, the temperatures represented are too low:



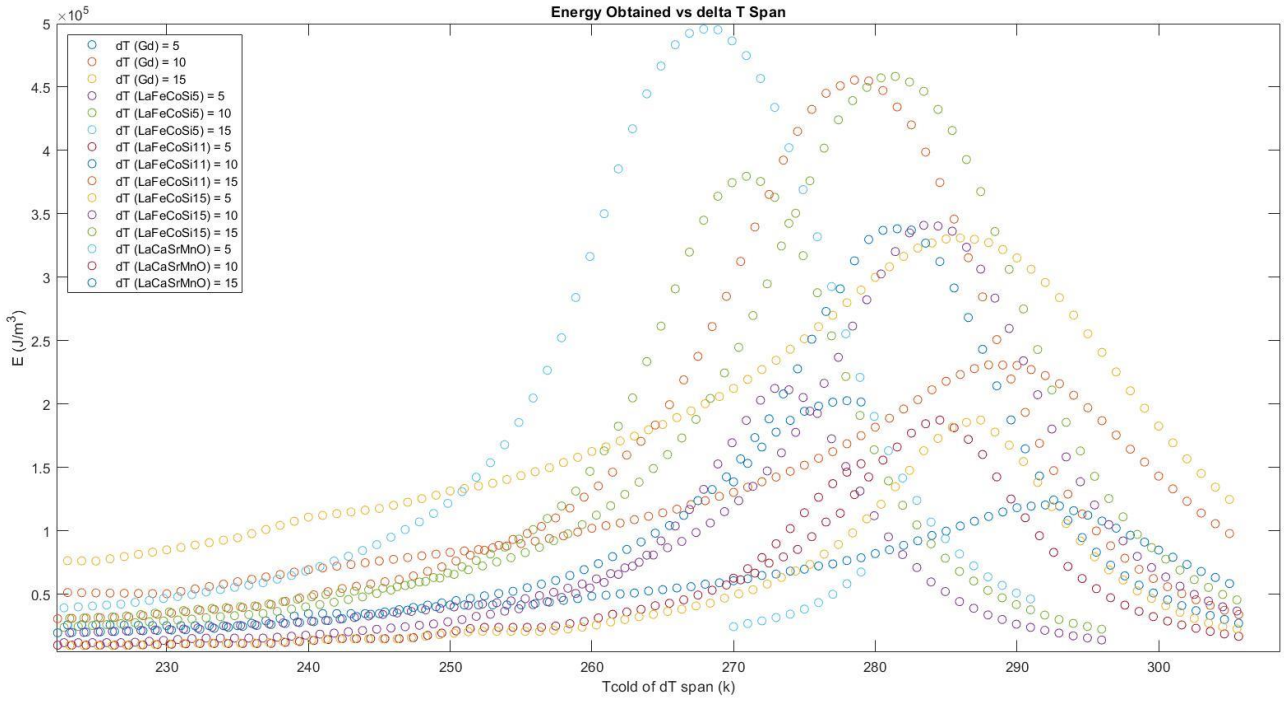
To better compare how the materials behave, they are all plotted together:

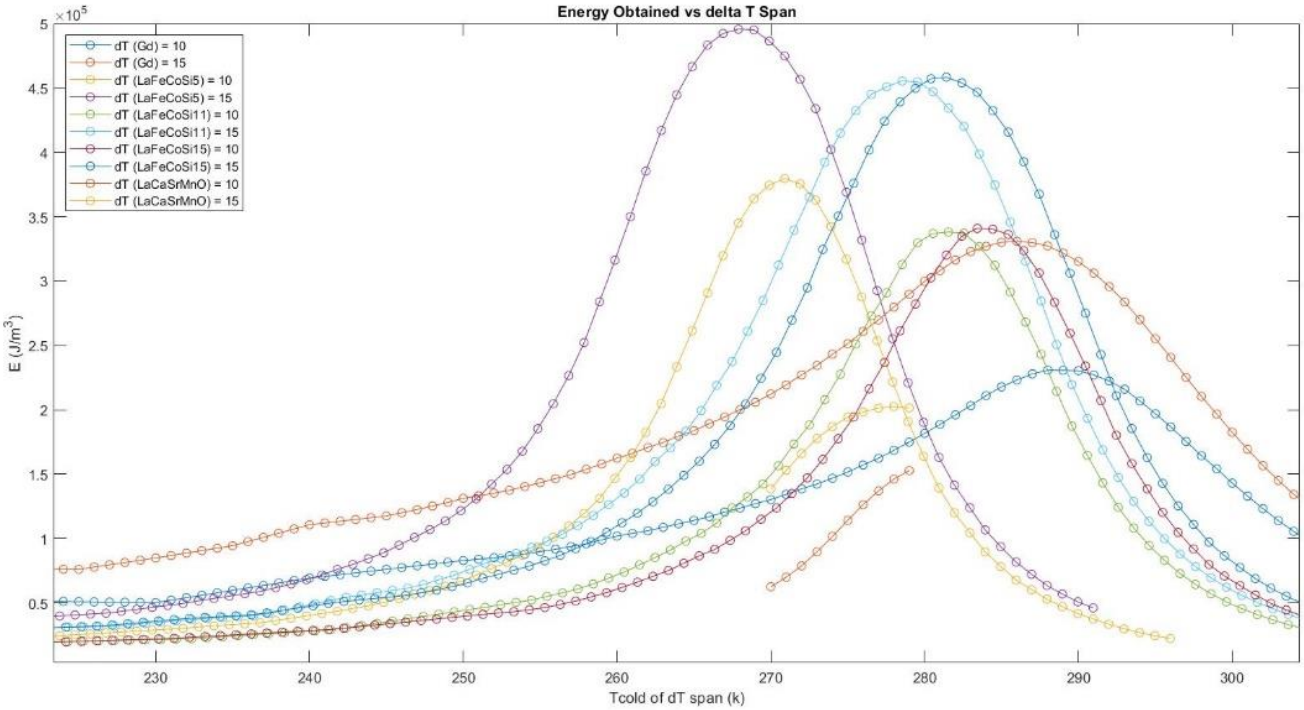




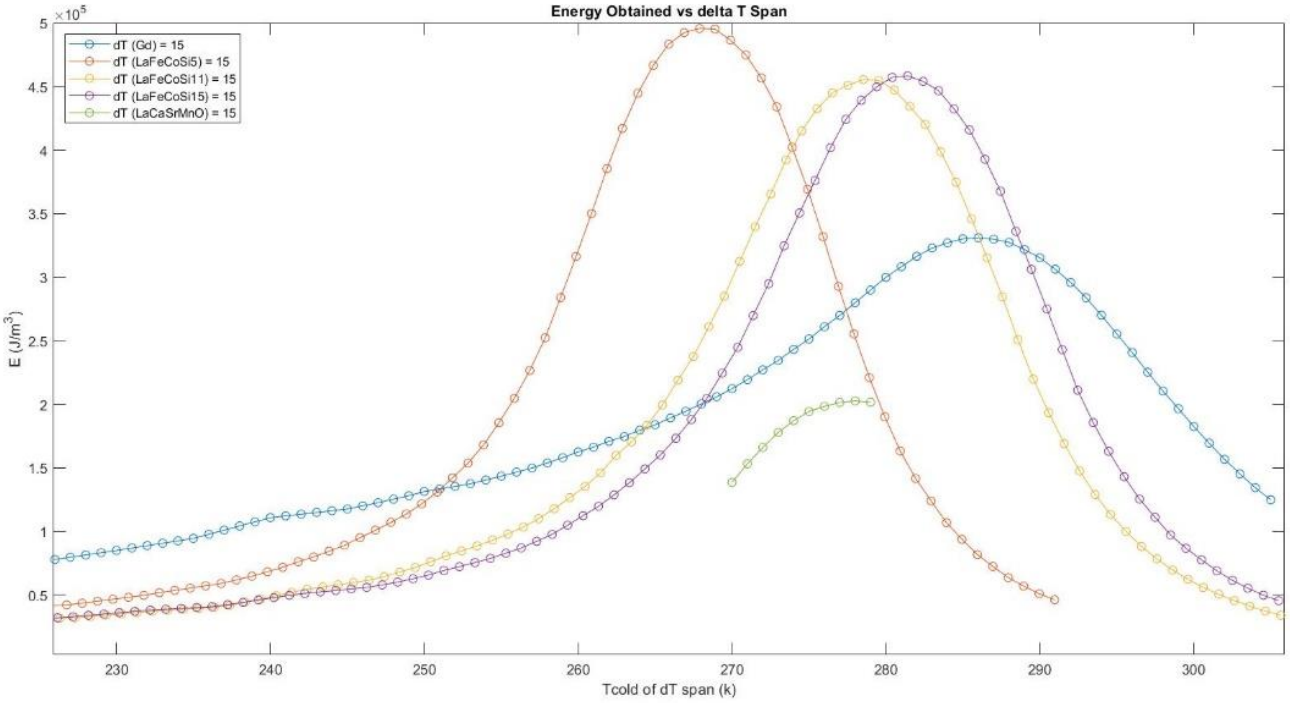
With the same ranges {5,10,15}k:

Considering ranges {10,15}, curves have been joint using lines for a better view:





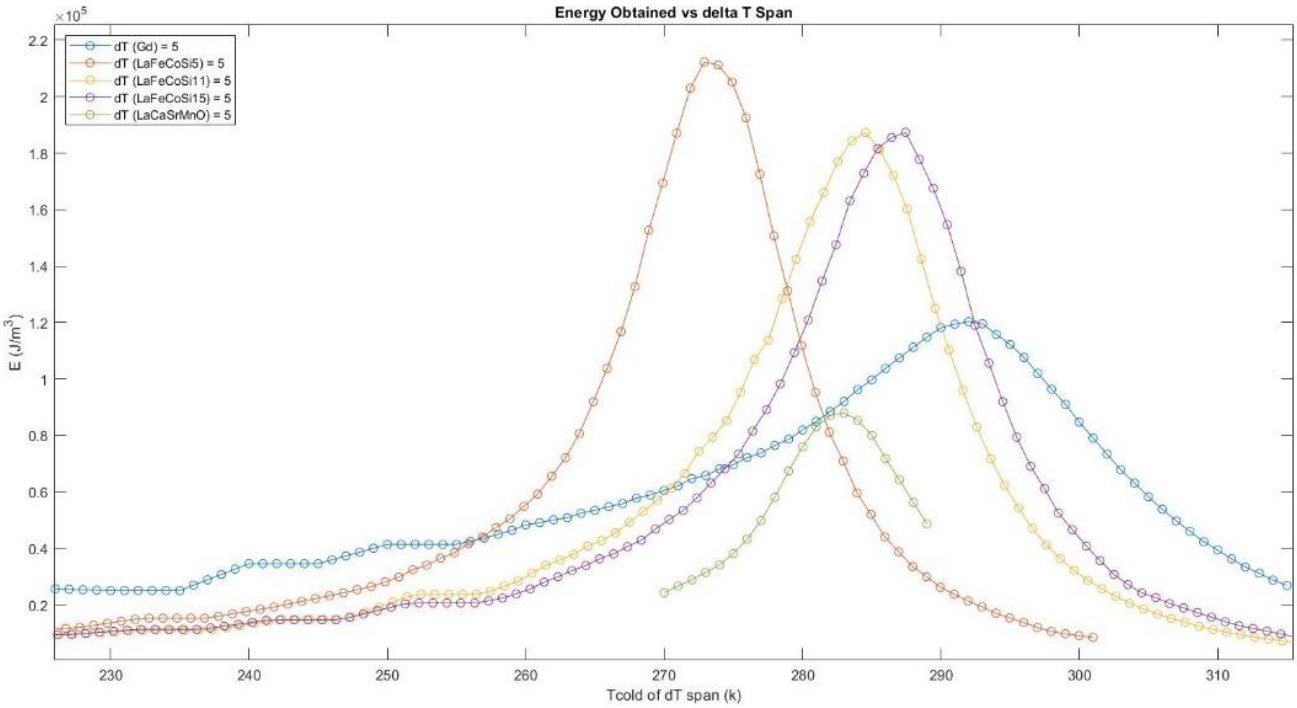
Some noticeable difference can already be seen. Plotting the material curves for a sole span of 15k the differences shall be more evident:

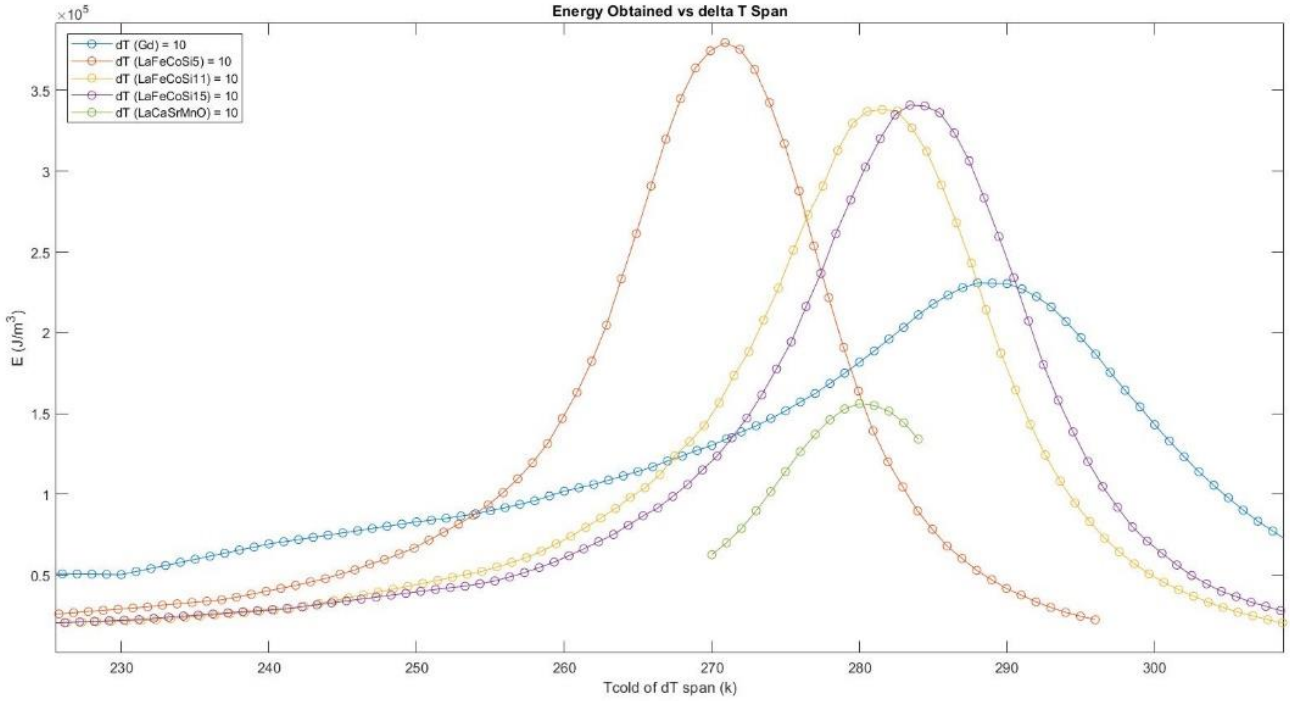


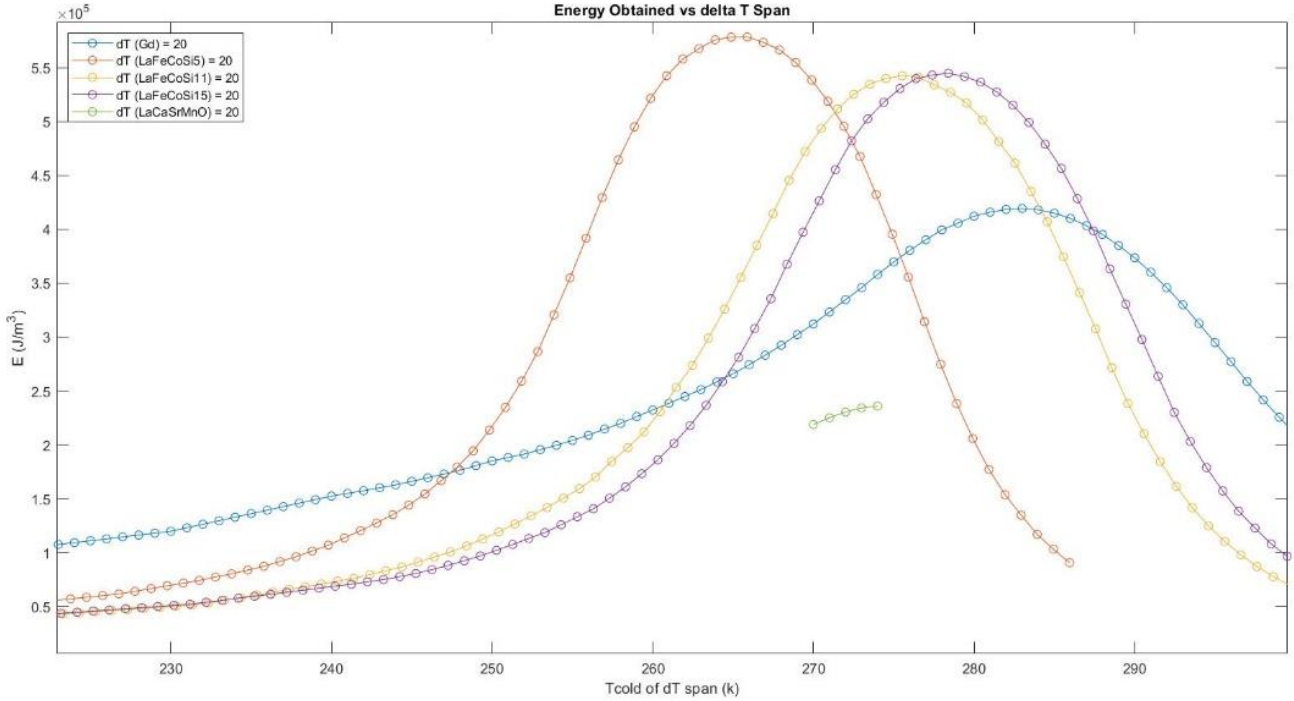
Several conclusions can summarize this results:

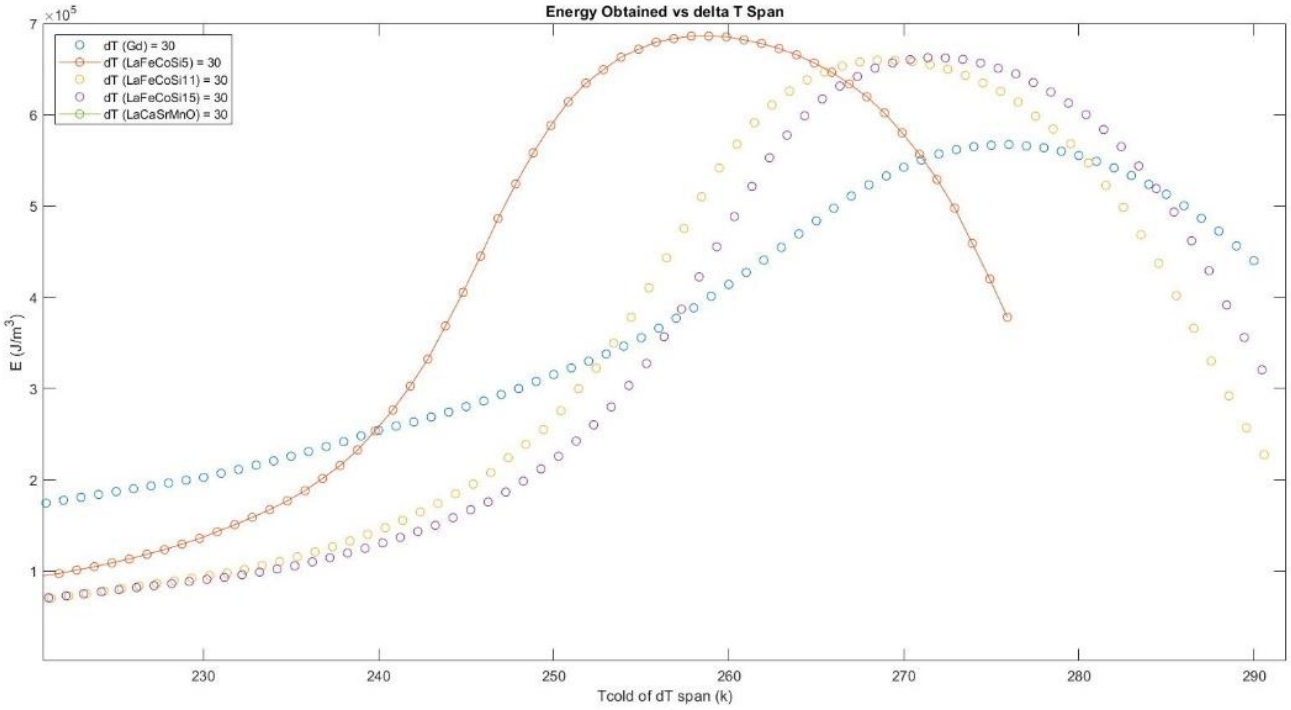
- LaCaSrMnO material clearly performs worst than the rest of materials when it come to energy per volume output considering temperature span. Therefore, and as its graphics are less representative due to a shorter temperature data, it may be taken out of further considerations.
- The higher energy per volume output is obtained for LaFeCoSi5.
- The material that achieves its energy peak at lower working temperature, thus more easy and cheap to implement, is LaFeCoSi5 (the temperatures are very low, though).

Thus, the best working material when it comes to energy per volume outcome in regards for a temperature span of 15k is without doubt LaFeCoSi5. Now further plots will be considered to study if this material is also better at other temperature spans:

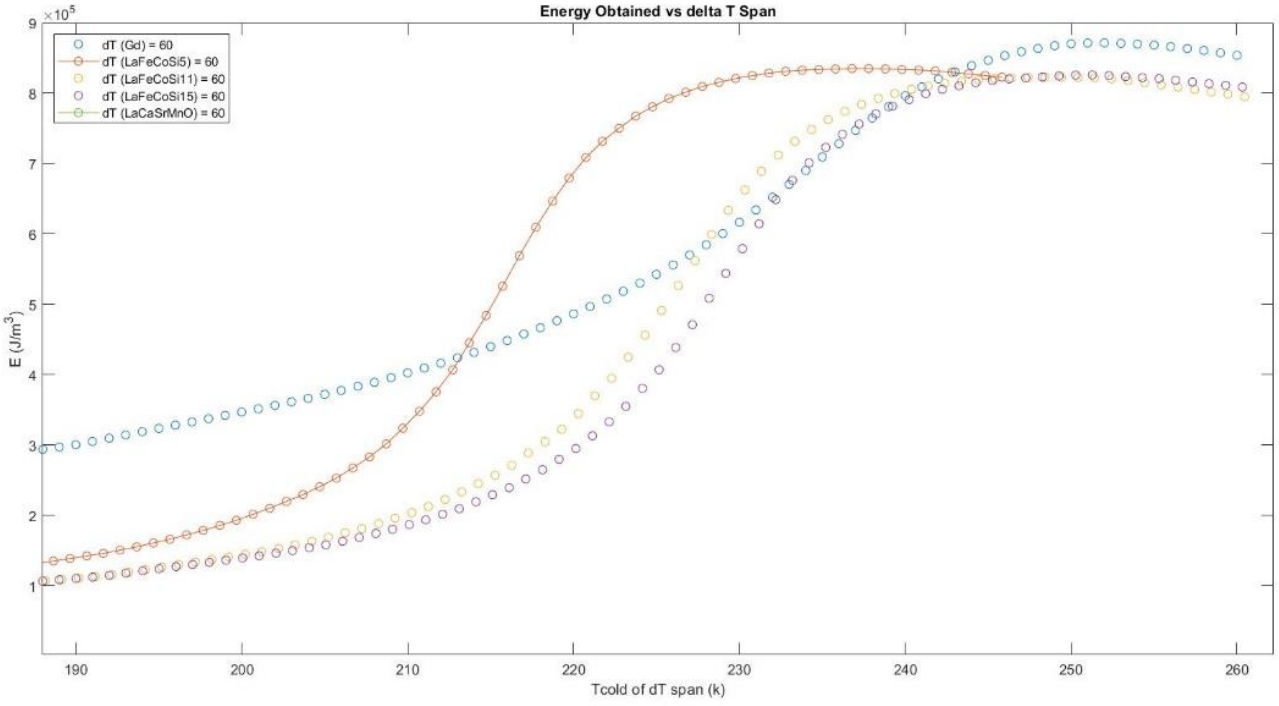



Energy Obtained vs delta T Span
×10^5
dT (Gd) = 10
dT (LaFeCoSi5) = 10
dT (LaFeCoSi11) = 10
dT (LaFeCoSi15) = 10
dT (LaCaSrMnO) = 10
E (J/m^3)
Tcold of dT span (k)

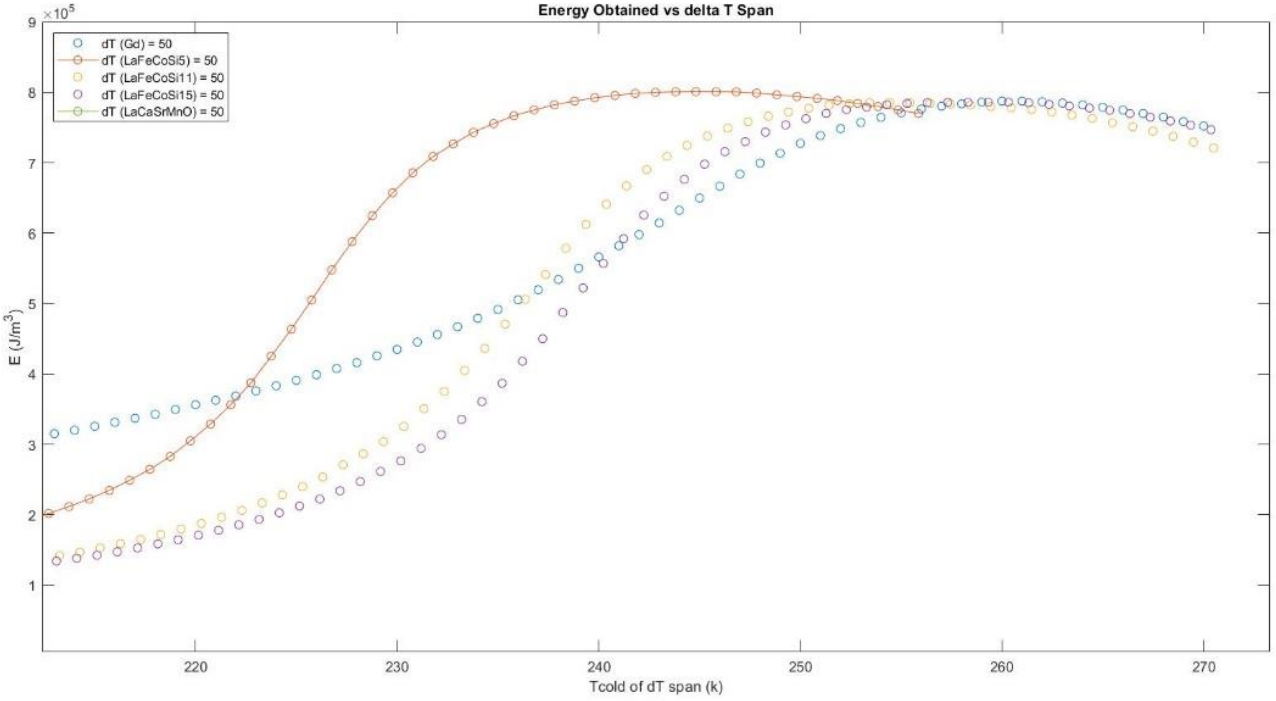

Energy Obtained vs delta T Span
×10^5
dT (Gd) = 20
dT (LaFeCoSi5) = 20
dT (LaFeCoSi11) = 20
dT (LaFeCoSi15) = 20
dT (LaCaSrMnO) = 20
E (J/m^3)
Tcold of dT span (k)



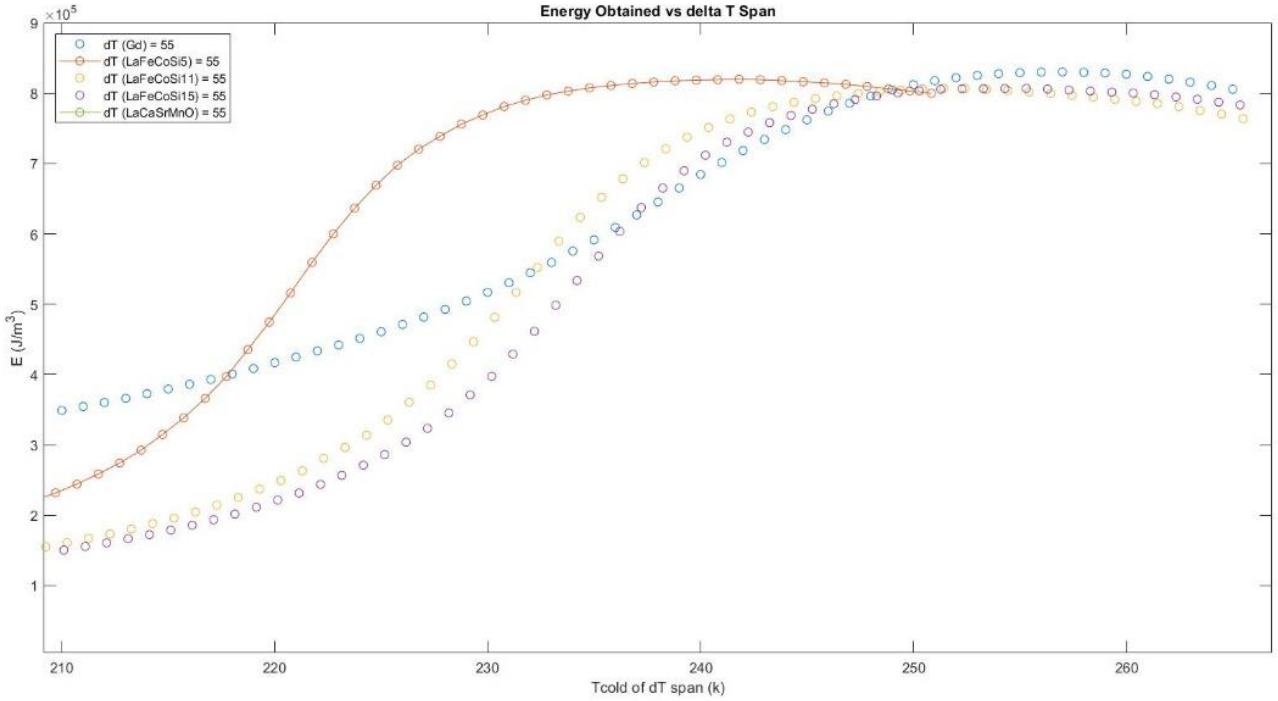
As it can be seen, for all this ranges, LaFeCoSi5 is the best material to work with. However, the relative improvement of this material to that if the others slightly decreases with the increase of temperature span. Further span are considered so as to check it LaFeCoSi5 can be outperformed by other materials at a certain temperature span.



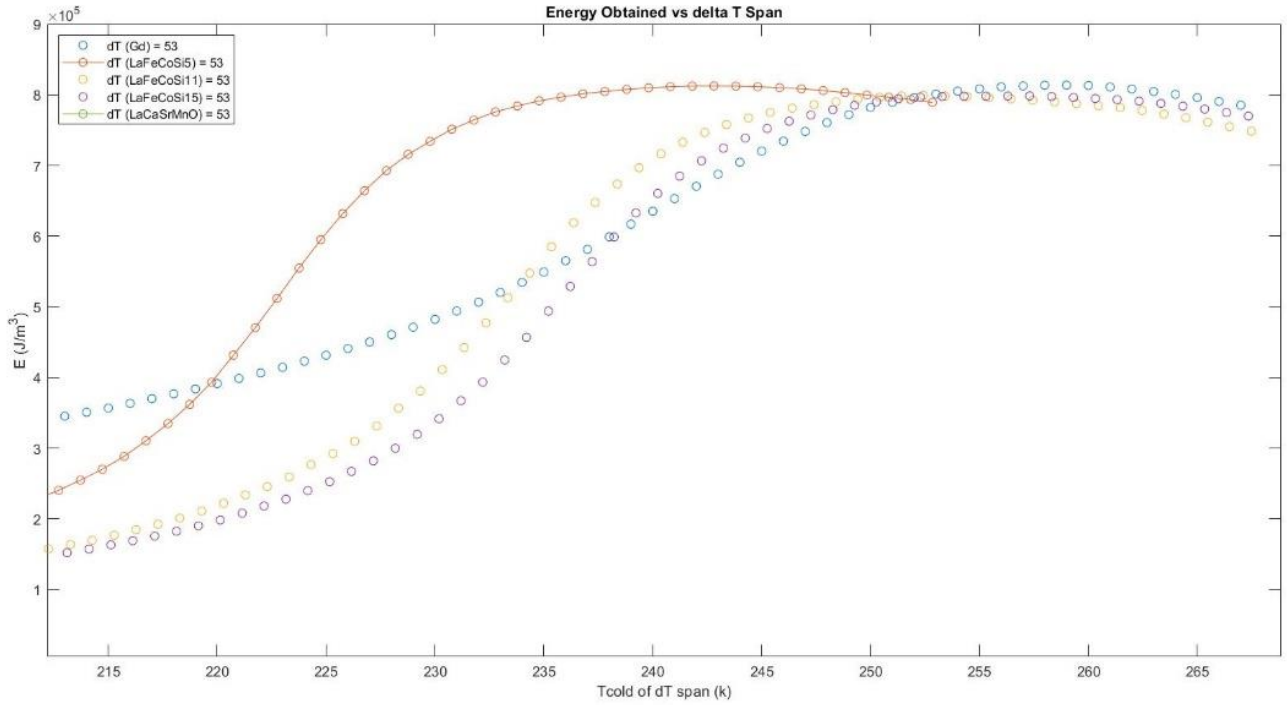
As it can be seen, for a temperature span of 60k, Gadolinium outperforms LaFeCosi5. Note that the working temperatures are still lower for LaFeCoSi5. It is then interesting to establish the range at which LaFeCoSi5 is a better material in terms of energy per volume production.



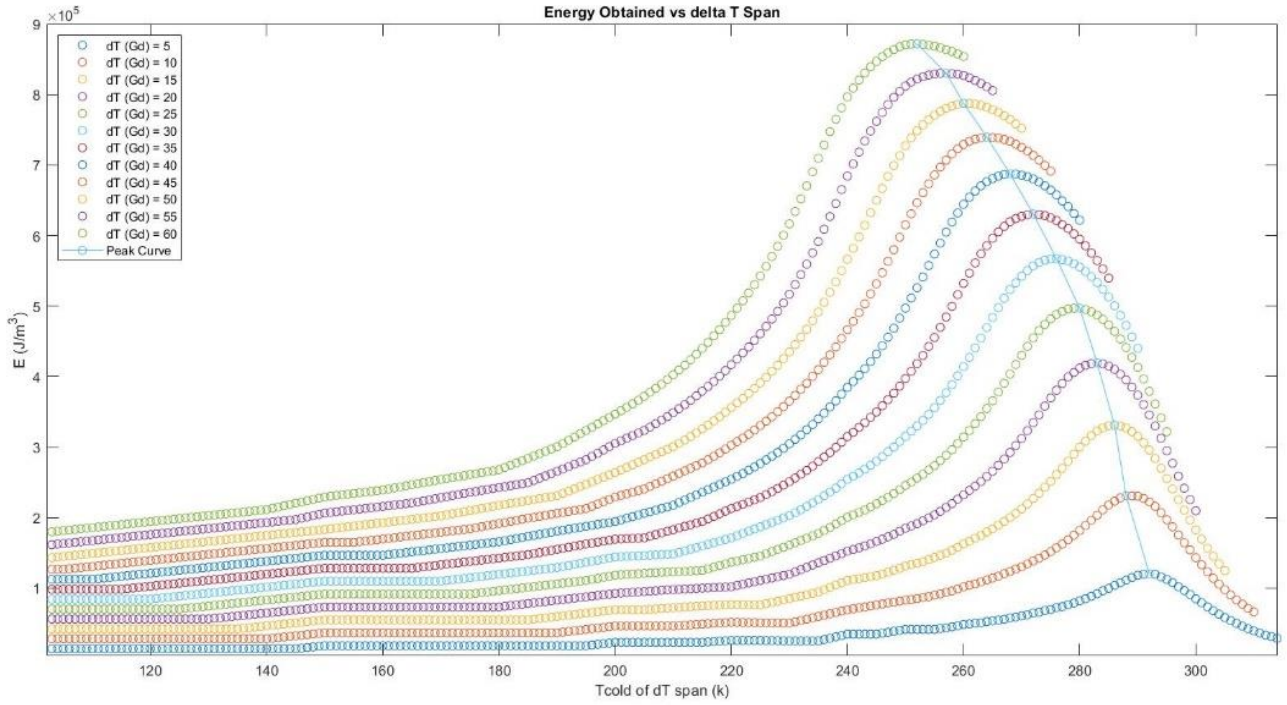
For a span of 50k, LaFeCoSi5 is still the best material, but for 55k, Gadolinium is just a little better:

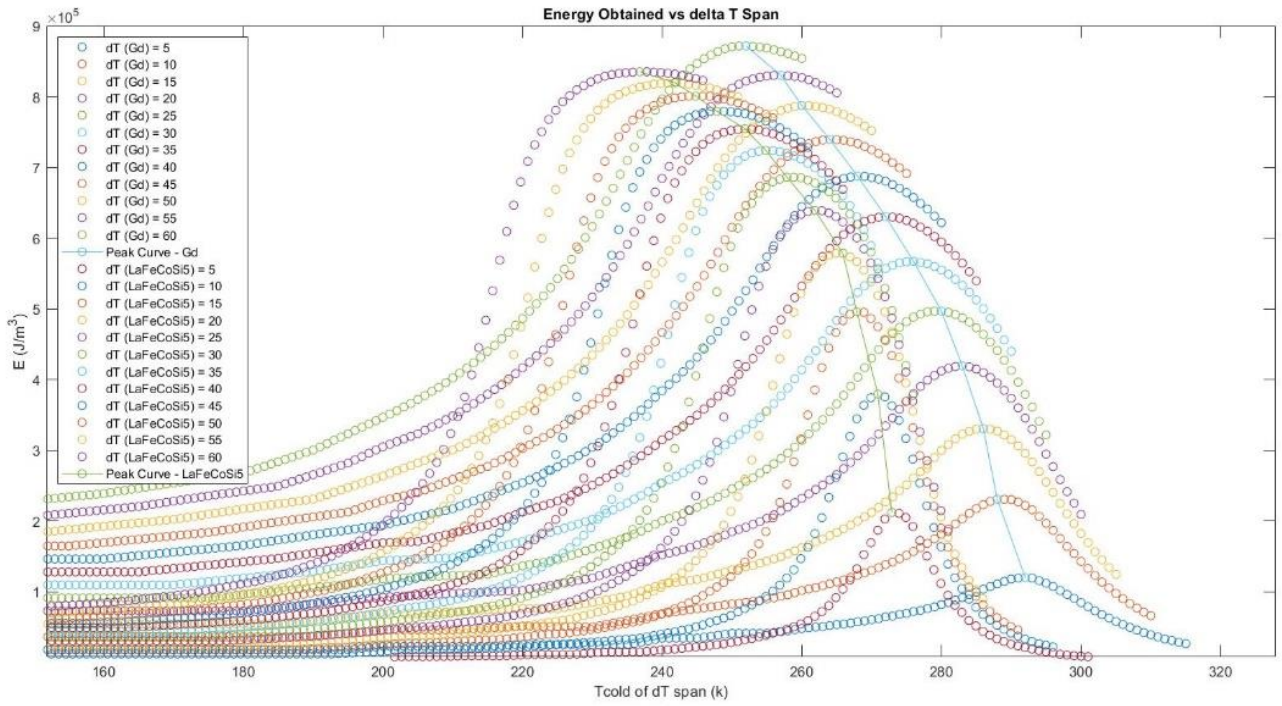


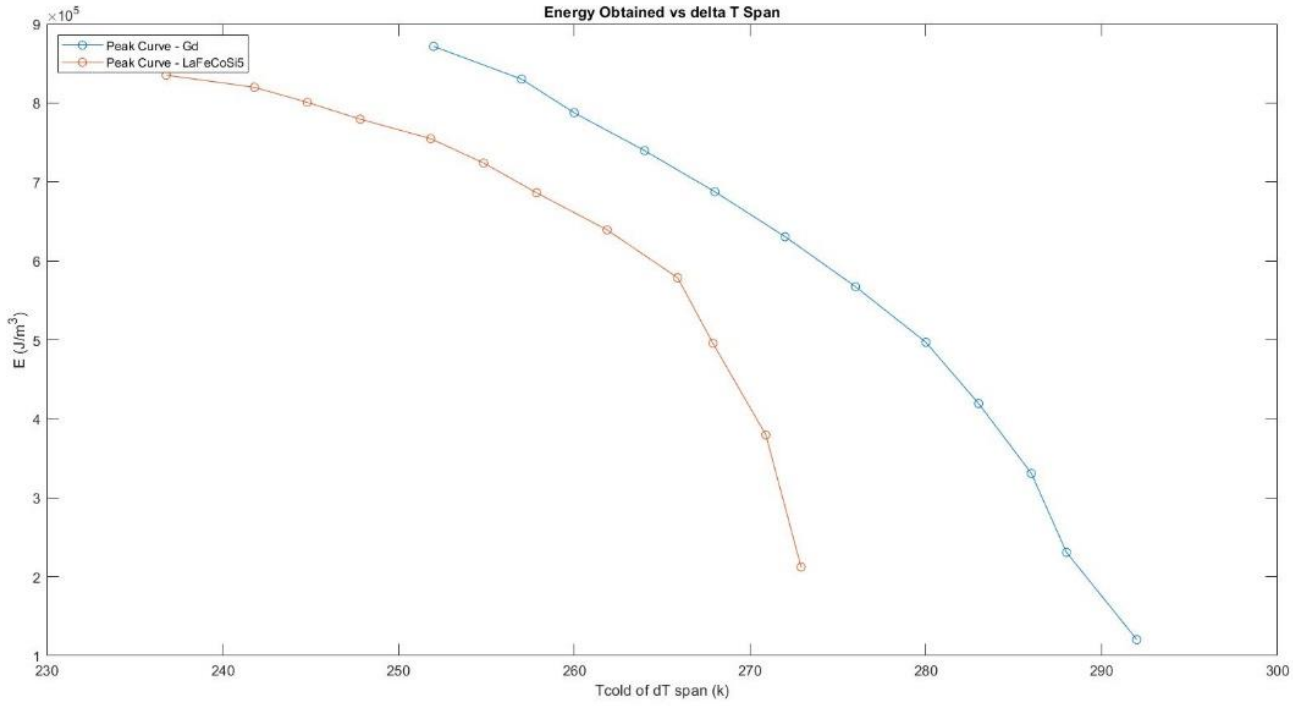
The turning point is found at a temperature span of 53k, where the peak for LaFeCoSi5 is 8.124*10^5 J/m^3, while Gadolinium has a peak of 8.135*10^5 J/m^3.

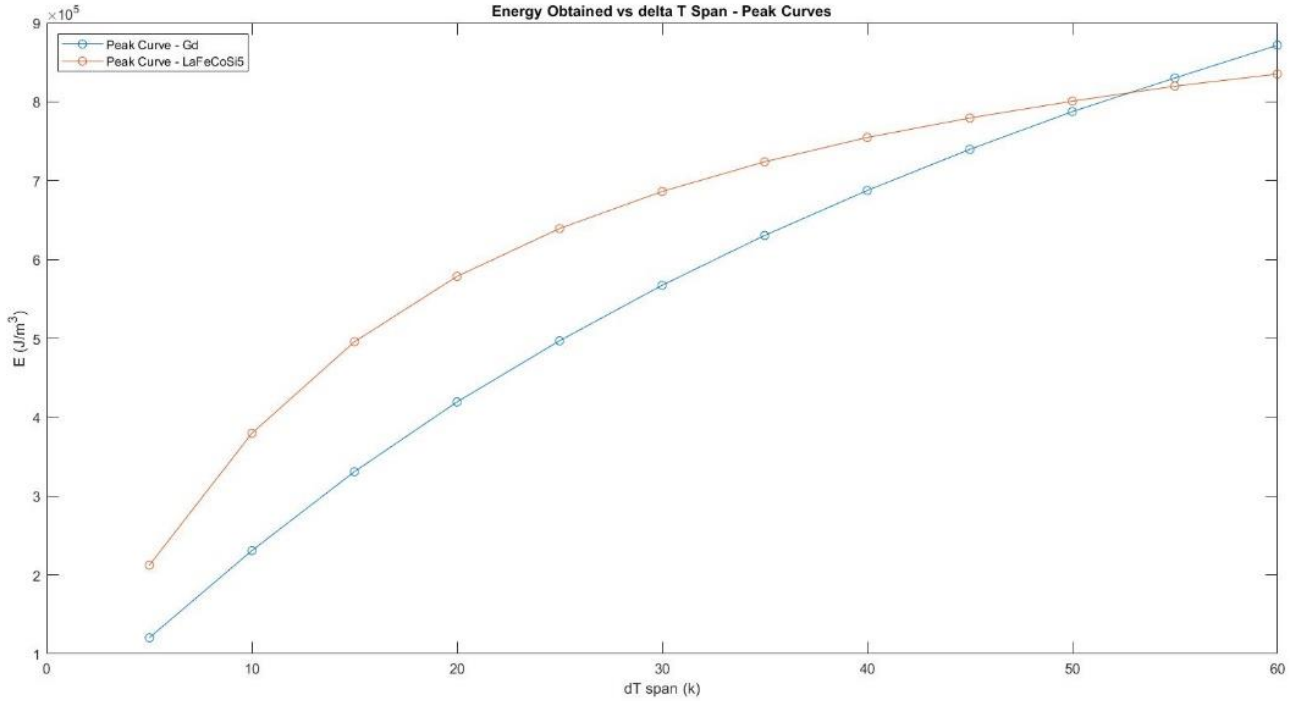


To obtain the temperature at which Gd outperforms LaFeCoSi5 two curves are plotted representing the energy per volume output peak for each temperature span:



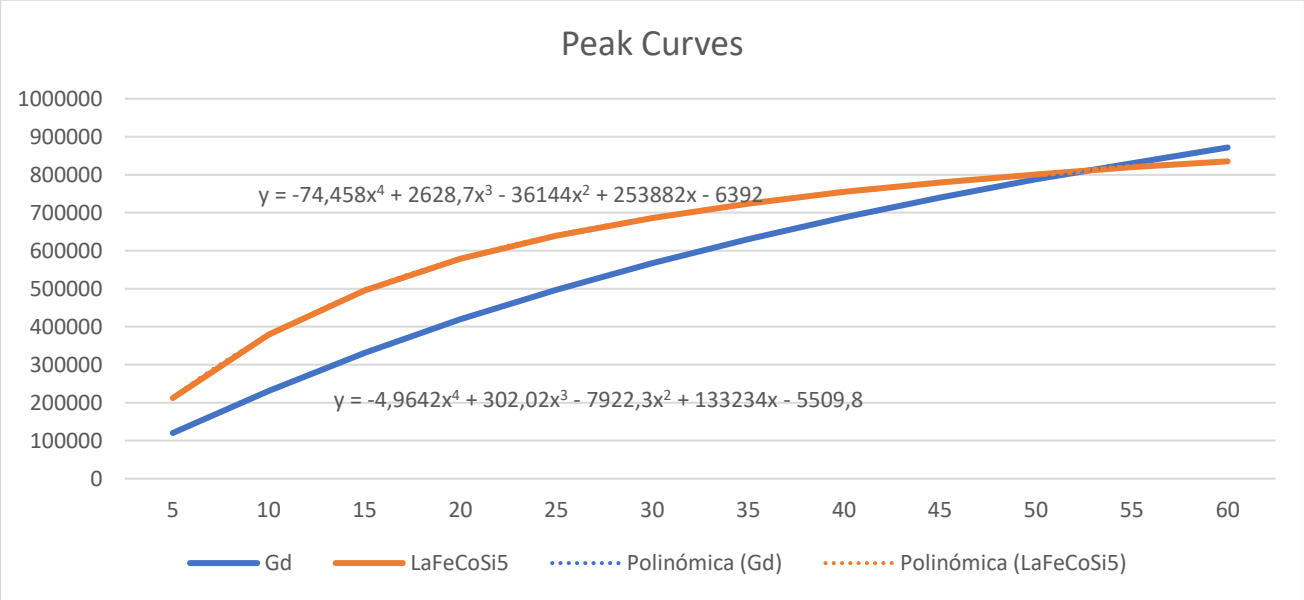

Energy Obtained vs delta T Span
×10^5
E (J/m^3)
Tcold of dT span (k)
dT (Gd) = 5
dT (Gd) = 10
dT (Gd) = 15
dT (Gd) = 20
dT (Gd) = 25
dT (Gd) = 30
dT (Gd) = 35
dT (Gd) = 40
dT (Gd) = 45
dT (Gd) = 50
dT (Gd) = 55
dT (Gd) = 60
Peak Curve - Gd
dT (LaFeCoSi5) = 5
dT (LaFeCoSi5) = 10
dT (LaFeCoSi5) = 15
dT (LaFeCoSi5) = 20
dT (LaFeCoSi5) = 25
dT (LaFeCoSi5) = 30
dT (LaFeCoSi5) = 35
dT (LaFeCoSi5) = 40
dT (LaFeCoSi5) = 45
dT (LaFeCoSi5) = 50
dT (LaFeCoSi5) = 55
dT (LaFeCoSi5) = 60
Peak Curve - LaFeCoSi5


Energy Obtained vs delta T Span
×10^5
E (J/m^3)
Tcold of dT span (k)
Peak Curve - Gd
Peak Curve - LaFeCoSi5



To find the exact temperature at which the meeting happens a polynomic equation can be attached to each peak curve. The equation for Gd is found to be: y = -4.9642x4 + 302.02x3 - 7922.3x2 + 133234x - 5509.8, being x the value of the span normalized by the size of the span vector dT, x = dT/12.

The equation for LaFeCoSi5 is y = -74.458x4 + 2628.7x3 - 36144x2 + 253882x - 6392



The intersection of both functions is x = 10,6313. This means the span where LaFeCoSi5 is outperformed by Gd is x * 60 / 12 = 53.16 K

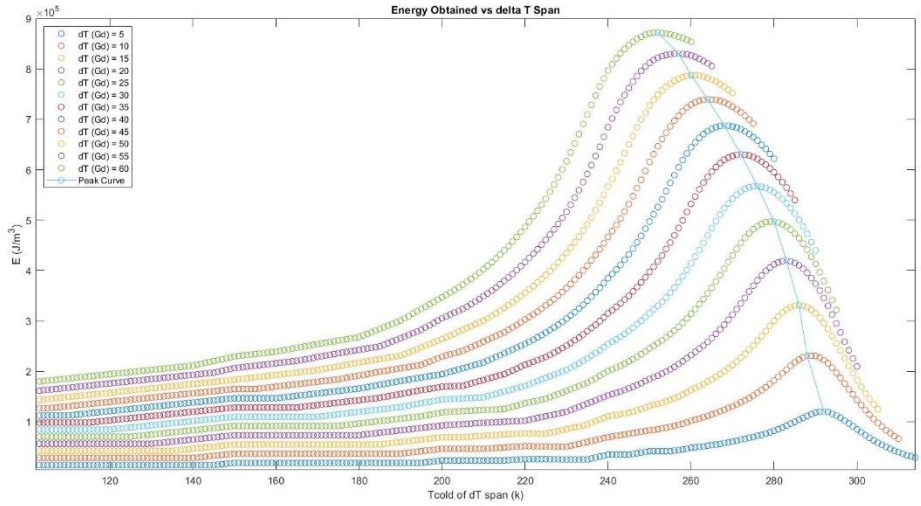
Tcurie and Tcenter:

It would be interesting to know the behavior of the peak temperature. What relation does it have with Tcurie? When does Tcurie happen to meet the temperature of the dT span? Does this imply a higher efficiency?

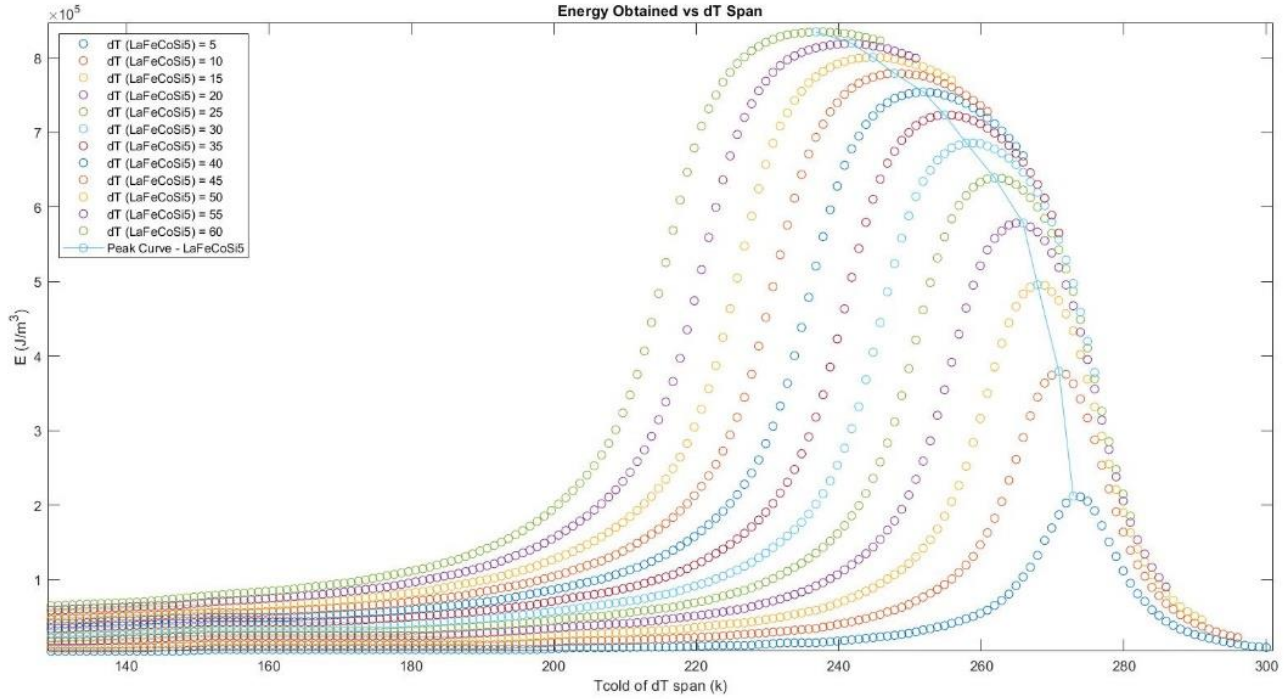
In order to answer these questions first the Curie temperatures must be known. According to [7] Curie temperature of pure Gadolinium is around Tc = 293 k. However, for commercial non-pure Gadolinium the Curie temperature is found to be Tc = 297 k by the inverse susceptibility method of the mean field Curie–Weiss law [5] and 295 k by fitting the susceptibility to an expression with a critical exponent $\chi \alpha (T\text{-}Tc)^{-\gamma}$. In order to work with a precise Curie value of commercial Gadolinium, the Curie temperature will be assumed to be Tc_{Gd} = 296 k.

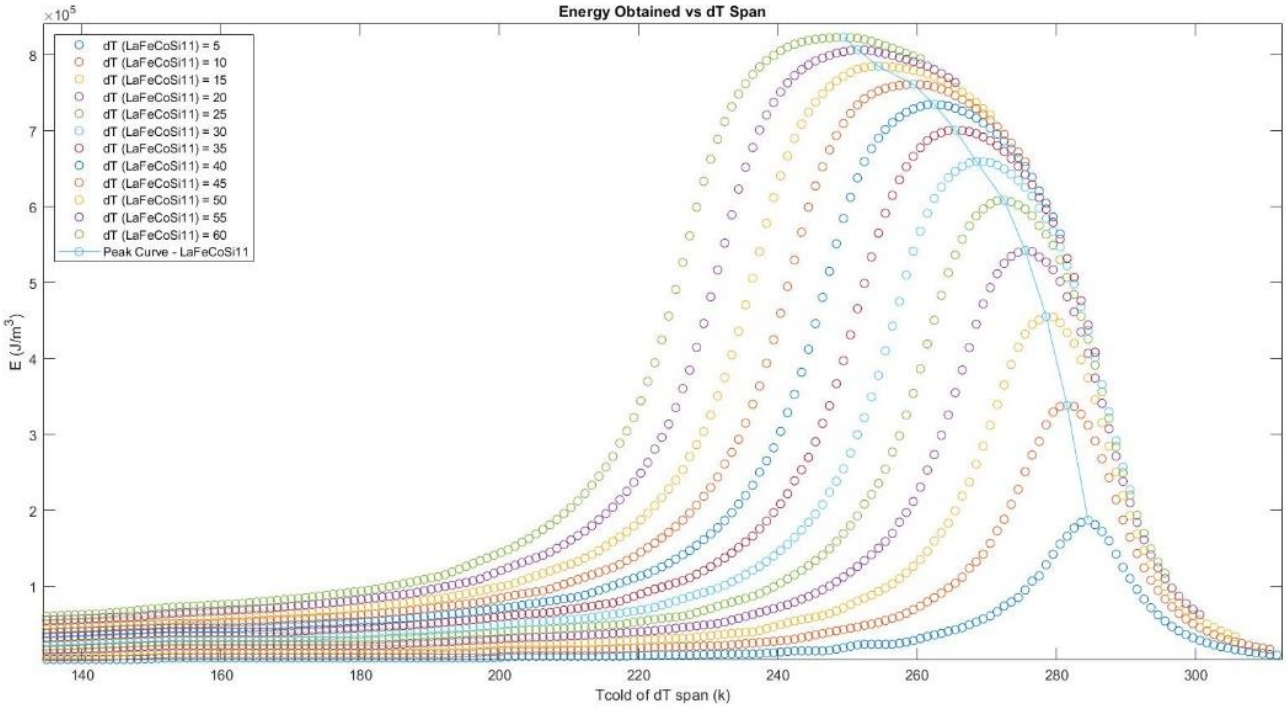
For LaFeCoSin materials the Curie temperature is not known, but is in the range of 276 k – 288 k according to [7]. For $La_{0.67}Ca_{0.33-x}Sr_xMnO_3$, the Curie temperature spans from 267 k for $La_{0.67}Ca_{0.33}MnO_3$ to 370 k for $La_{0.67}Sr_{33}MnO_3$ [6]. As the Curie temperature is only certainly known for Gd, this material will be used to study the behaviors described earlier.

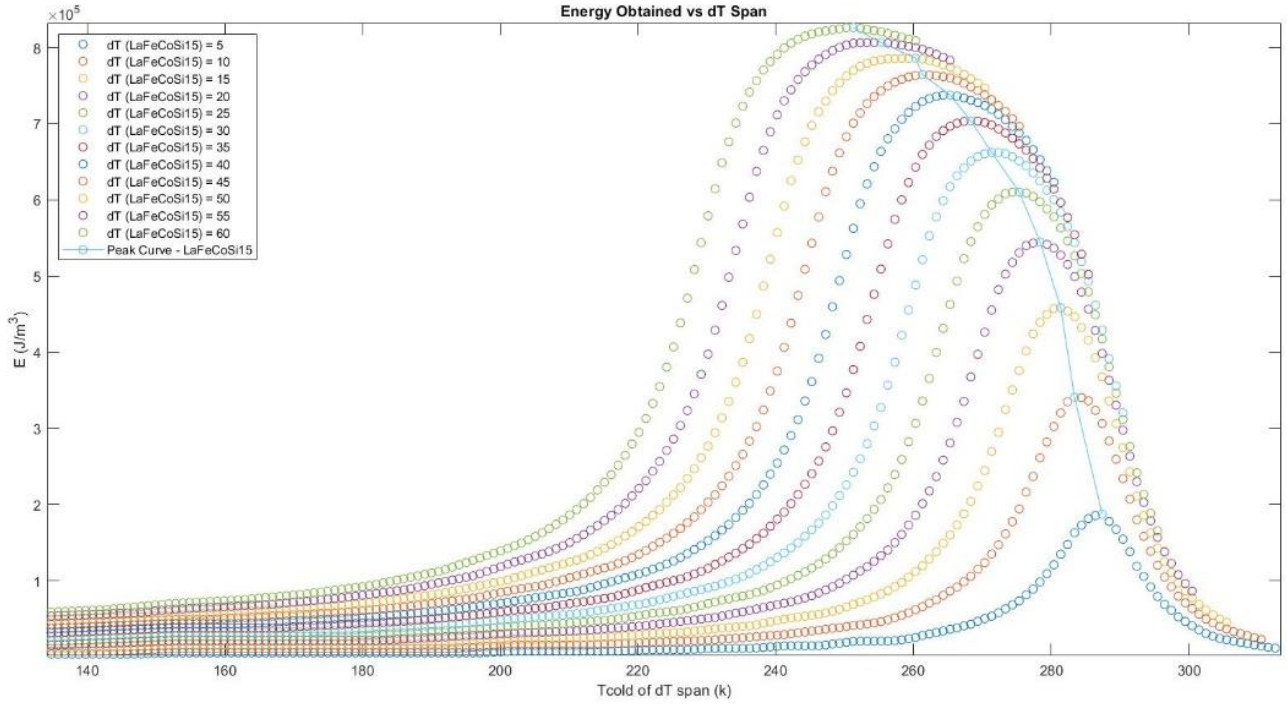
As it was seen in figure

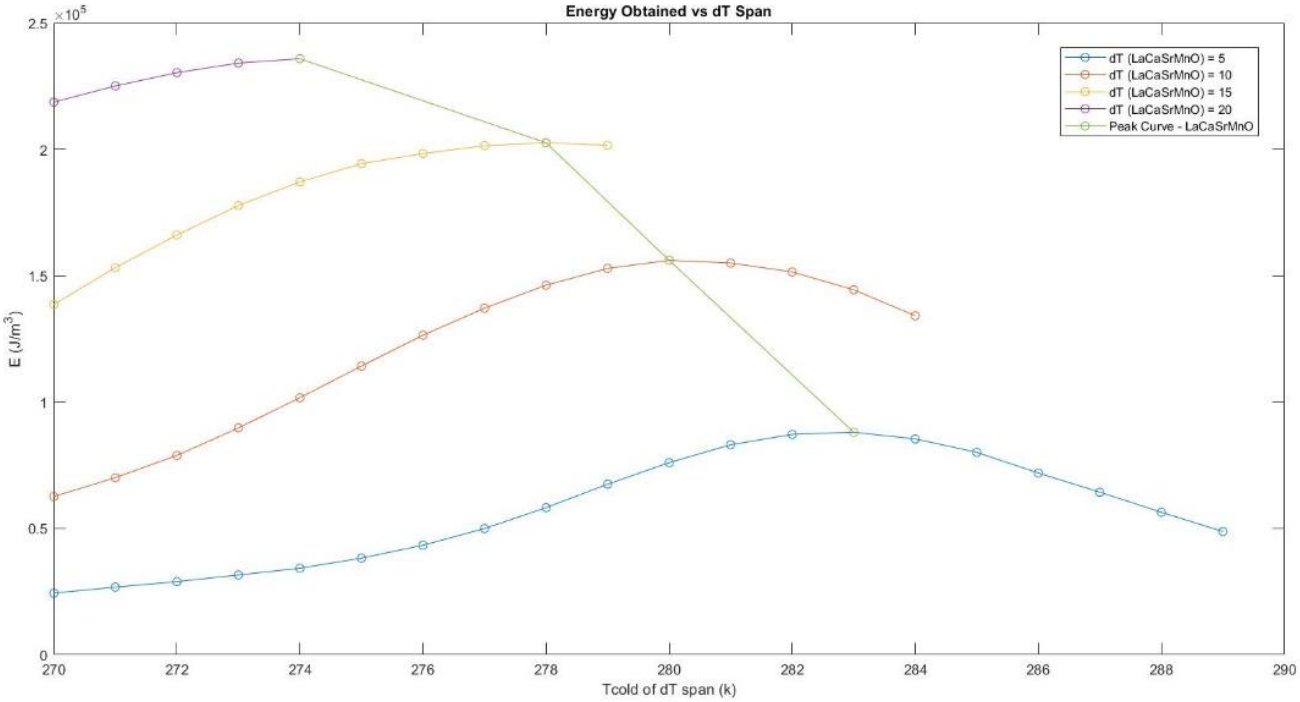


the peak in energy output versus Tspan gives a higher energy per volume and moves in the direction of lower temperatures the larger the temperature span. This is true also for the rest of the materials as it can be seen in the following figures:


Energy Obtained vs dT Span
×10^5
dT (LaFeCoSi5) = 5
dT (LaFeCoSi5) = 10
dT (LaFeCoSi5) = 15
dT (LaFeCoSi5) = 20
dT (LaFeCoSi5) = 25
dT (LaFeCoSi5) = 30
dT (LaFeCoSi5) = 35
dT (LaFeCoSi5) = 40
dT (LaFeCoSi5) = 45
dT (LaFeCoSi5) = 50
dT (LaFeCoSi5) = 55
dT (LaFeCoSi5) = 60
Peak Curve - LaFeCoSi5
E (J/m^3)
Tcold of dT span (k)

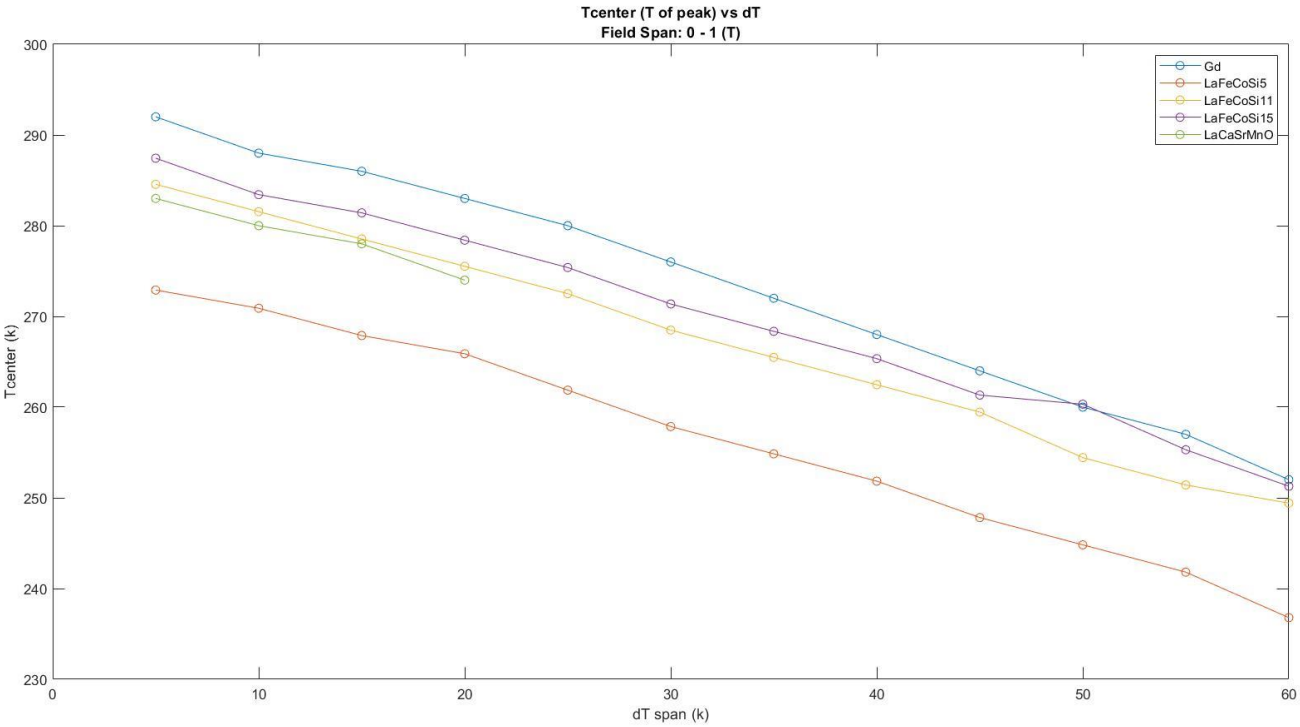

Energy Obtained vs dT Span
×10^5
dT (LaFeCoSi11) = 5
dT (LaFeCoSi11) = 10
dT (LaFeCoSi11) = 15
dT (LaFeCoSi11) = 20
dT (LaFeCoSi11) = 25
dT (LaFeCoSi11) = 30
dT (LaFeCoSi11) = 35
dT (LaFeCoSi11) = 40
dT (LaFeCoSi11) = 45
dT (LaFeCoSi11) = 50
dT (LaFeCoSi11) = 55
dT (LaFeCoSi11) = 60
Peak Curve - LaFeCoSi11
E (J/m^3)
Tcold of dT span (k)





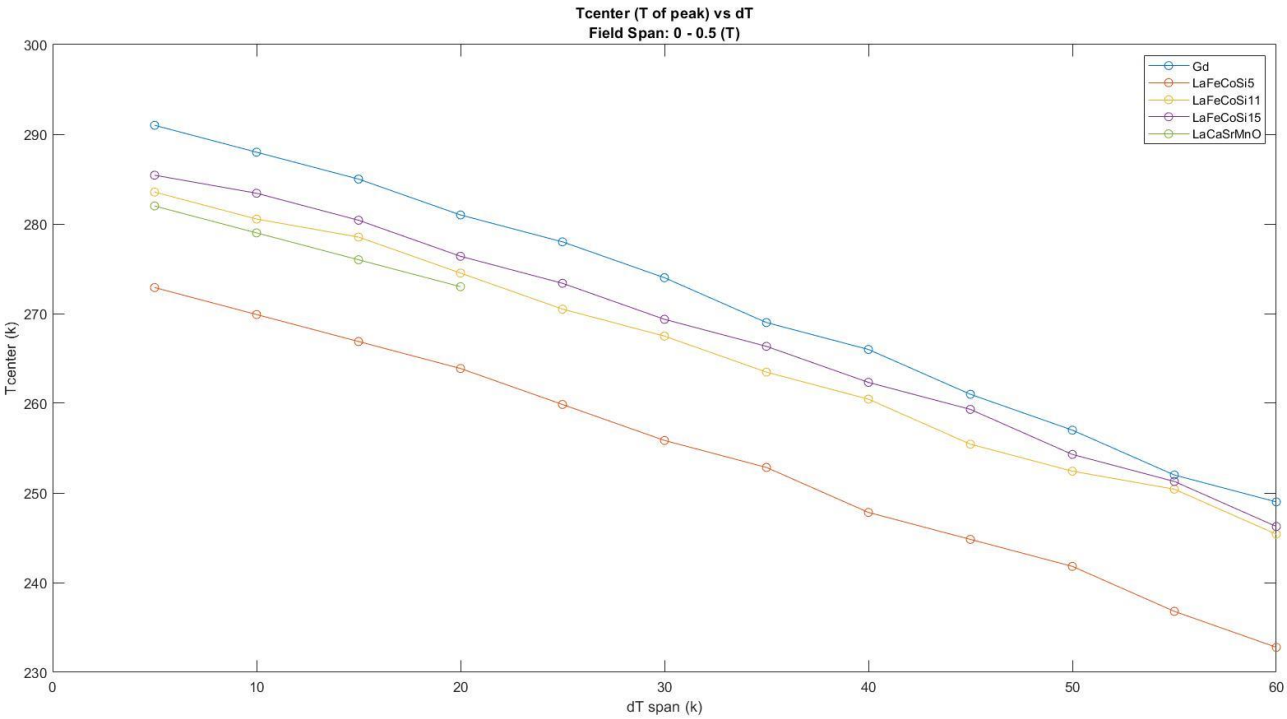
Note that the plot for LaCaSrMnO shows only up to 20k span, due to the data limitations mentioned before. It is confirmed that for all materials the higher the temperature span the higher the energy per volume output and the more the peak is moved towards lower temperatures.

To give a better view of how the peak moves with the increase in temperature span, a plot showing the temperature of the center of span (peak temperature) versus the span is presented:

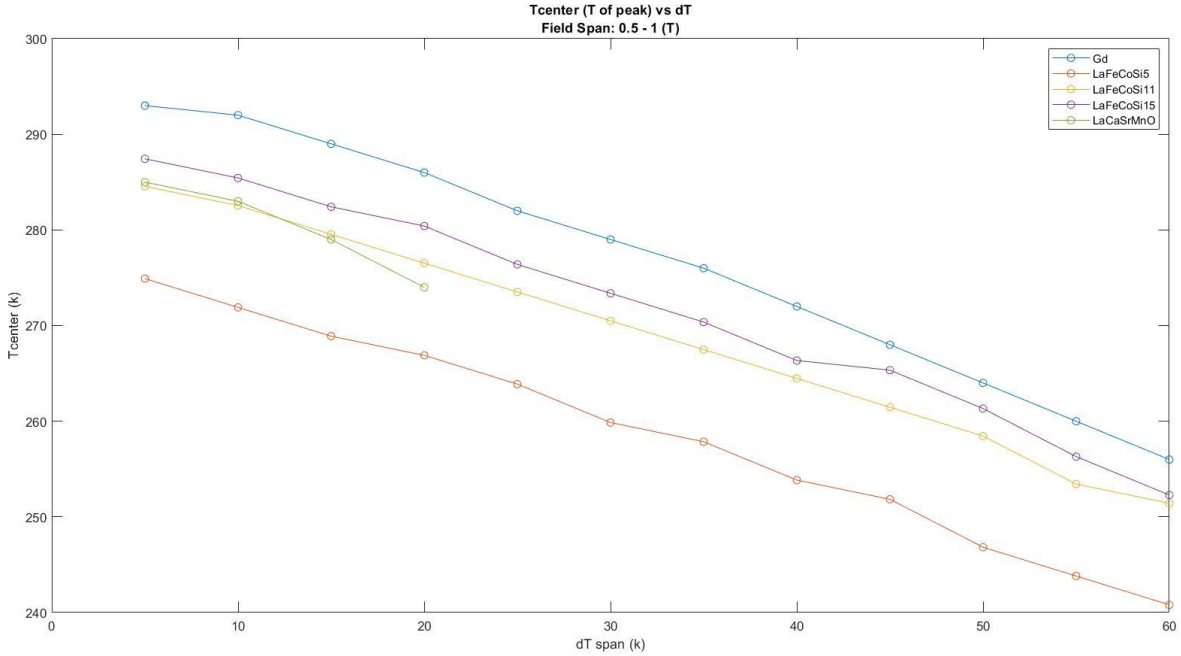


This figure confirms the fact that for all materials the peaks will move towards lower temperatures when the spans are increased. It is also appreciable that LaFeCoSi5 peaks are achieved for fairly lower temperatures than those for the rest of the thermomagnetic materials. The higher temperatures spans require low working temperatures that may be difficult to reach, specially in the case of LaFeCoSi5 (temperatures can be lower than -30ºC for dT = 60 k).

This result was obtained for a field span of 1 Tesla. It is now interesting to study the effect of changing the field span. From 0 to 0.5 T:

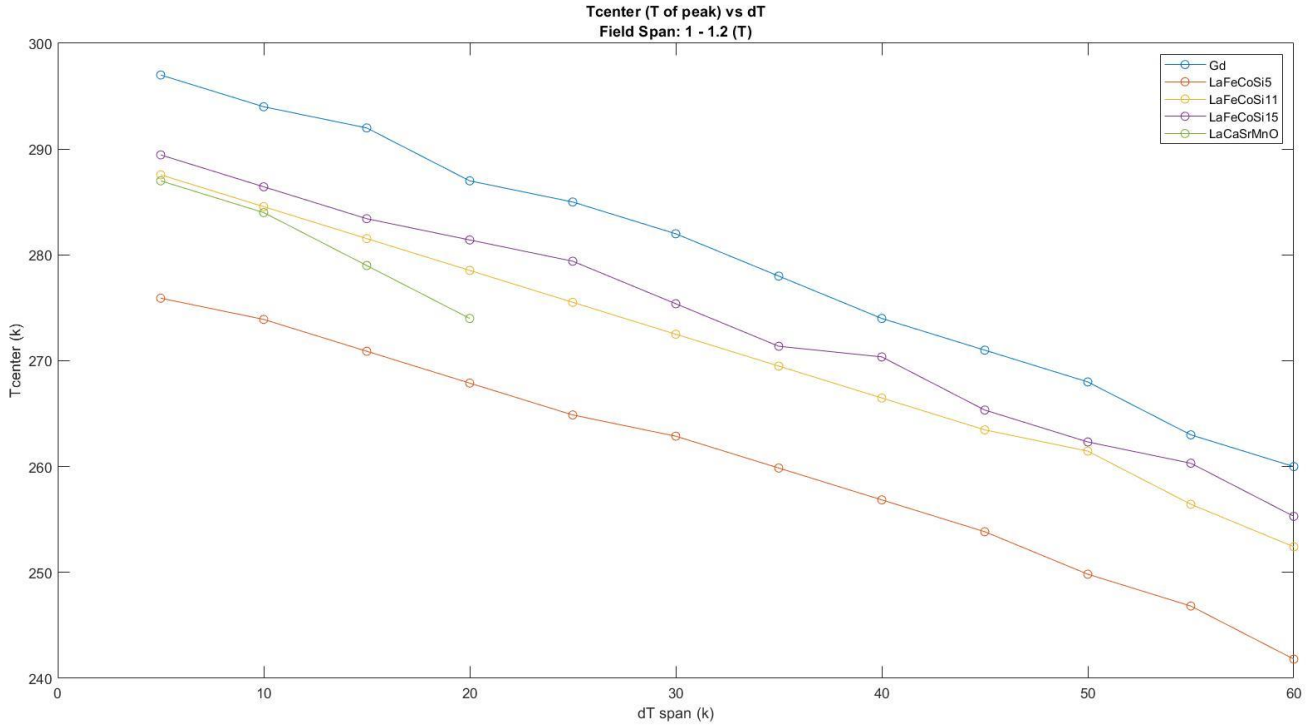


The behavior of materials seem to be very similar to the range of 0 T to 1 T. For large spans close to 60 k the temperature of the peak is slightly lower now. Also, for this span LaFeCoSi11 seems to do a bit better relatively to the other materials when comparing with the previous case. Now, same range at a starting field of 0.5 T:

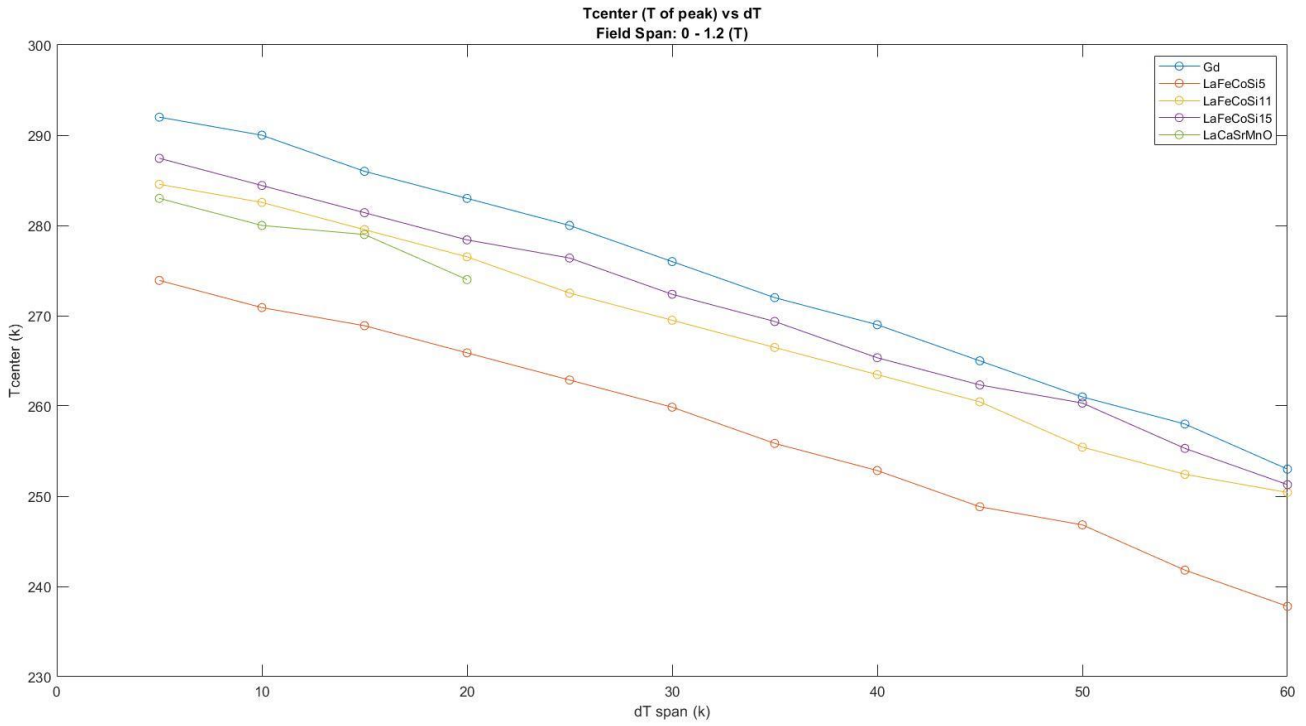


The general trend is maintained. LaCaSrMnO slightly surpasses LaFeCoSi11 for small spans, could be due to data aberrations. Peak temperatures for smaller spans are slightly bigger. For larger spans, the temperatures are lower.

Due to the limitation of material datasets, the maximum field that can be fully presented is 1.2 T. The behavior from 1 to 1.2 T follows the previous trend:

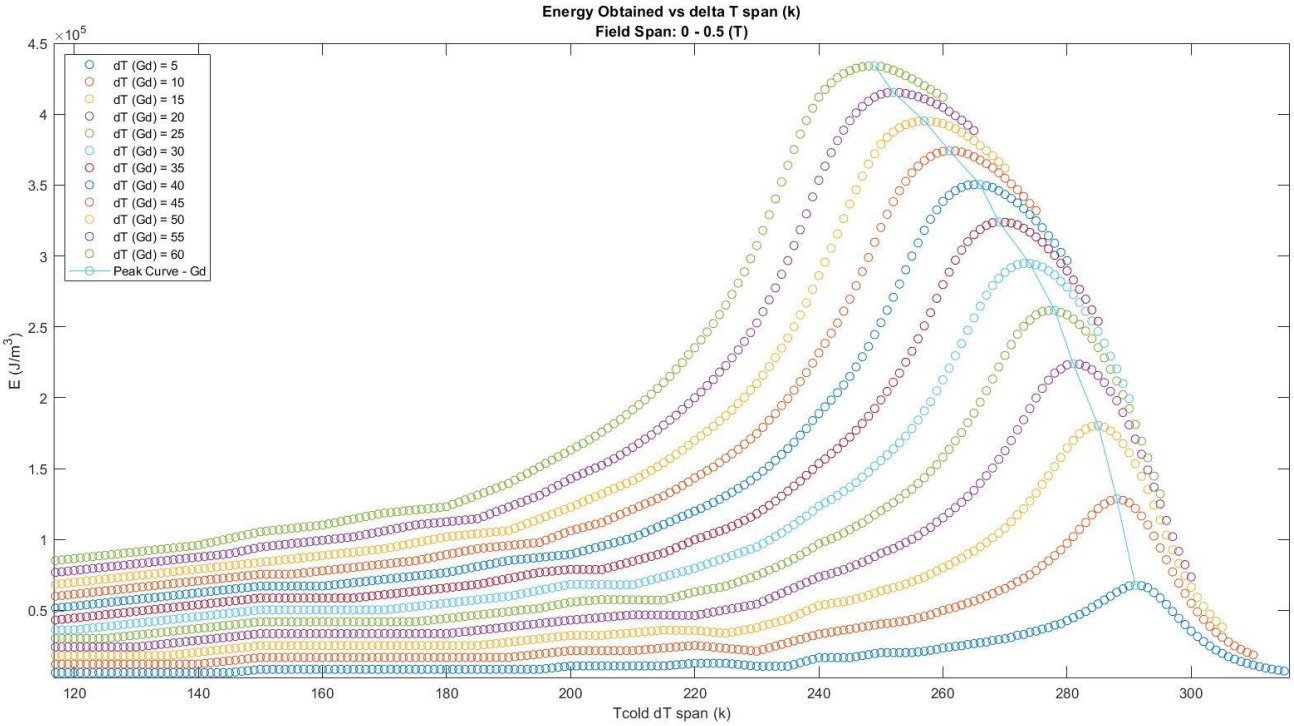


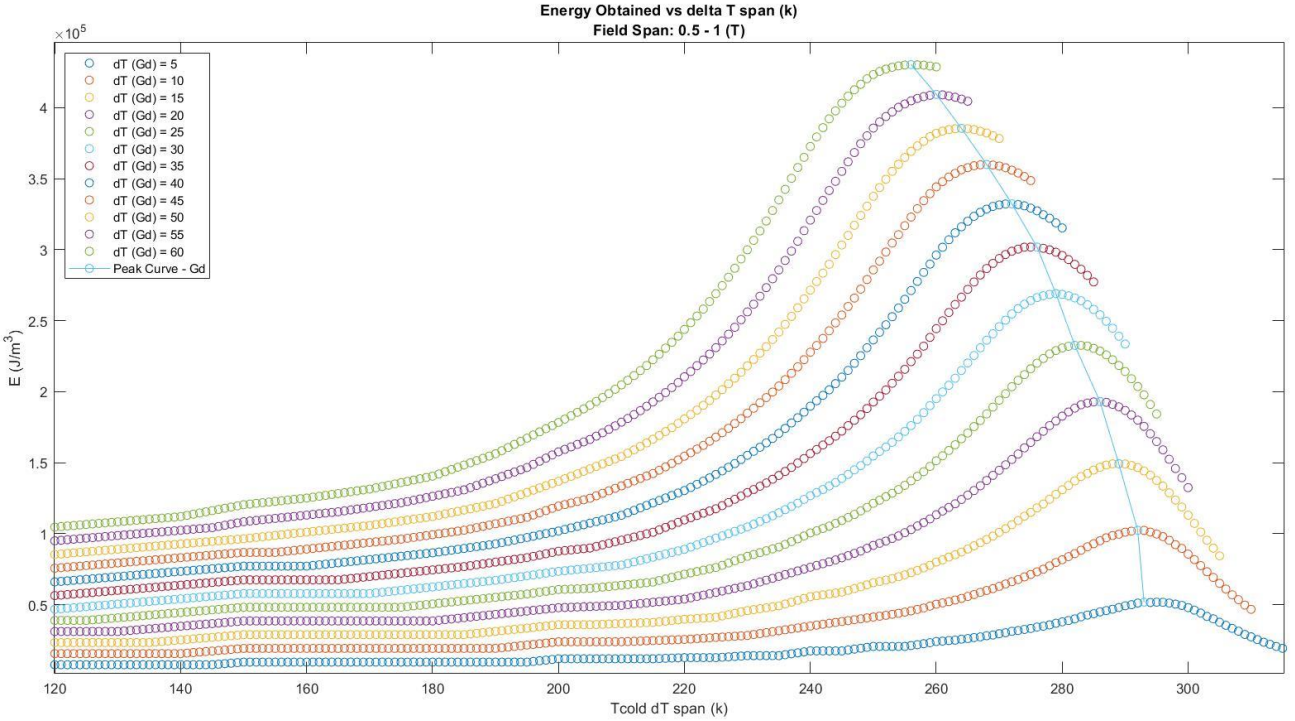
Other field spans can be considered:

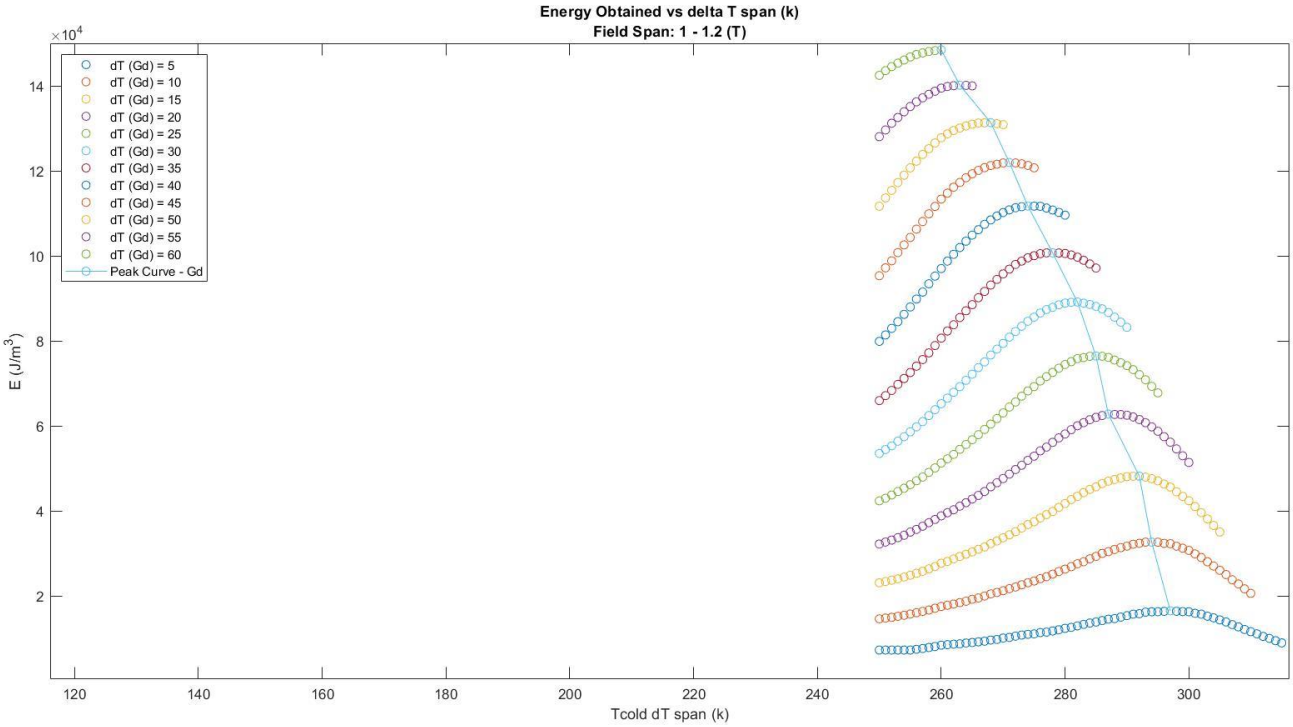


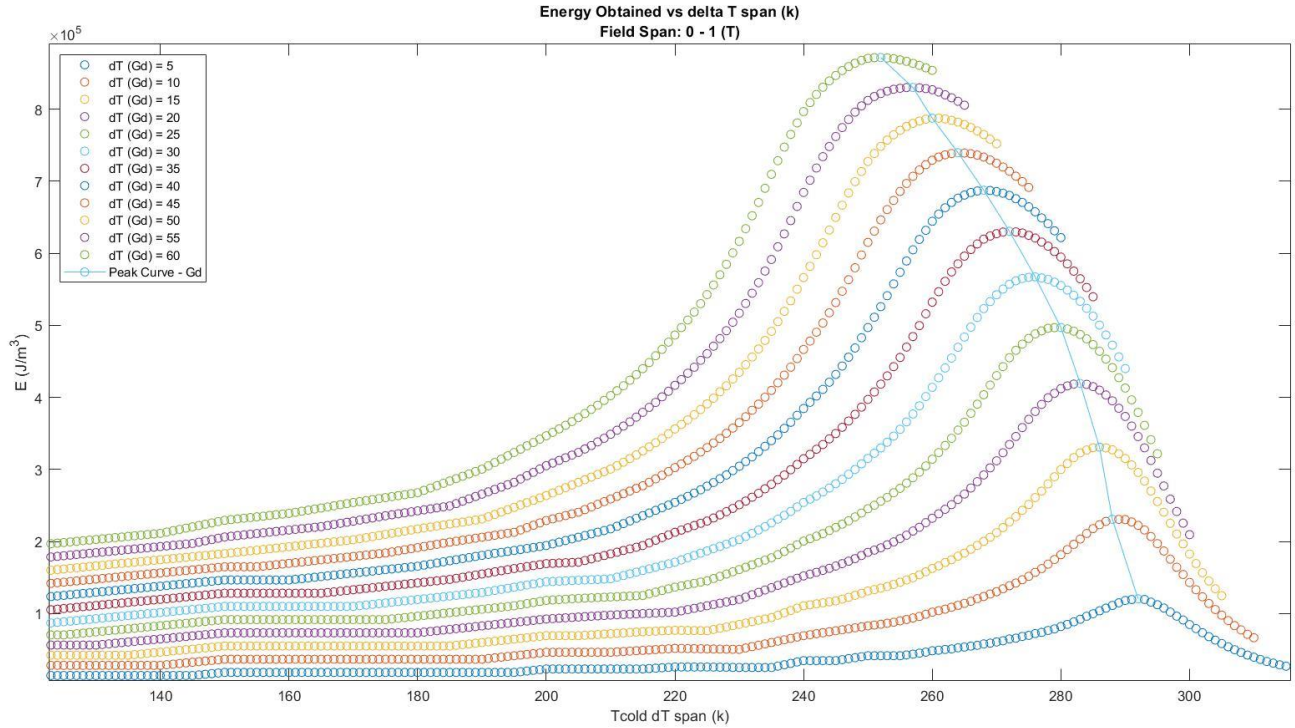
Note the drop in peak temperature for large temperature spans, when compared to the previous cases.

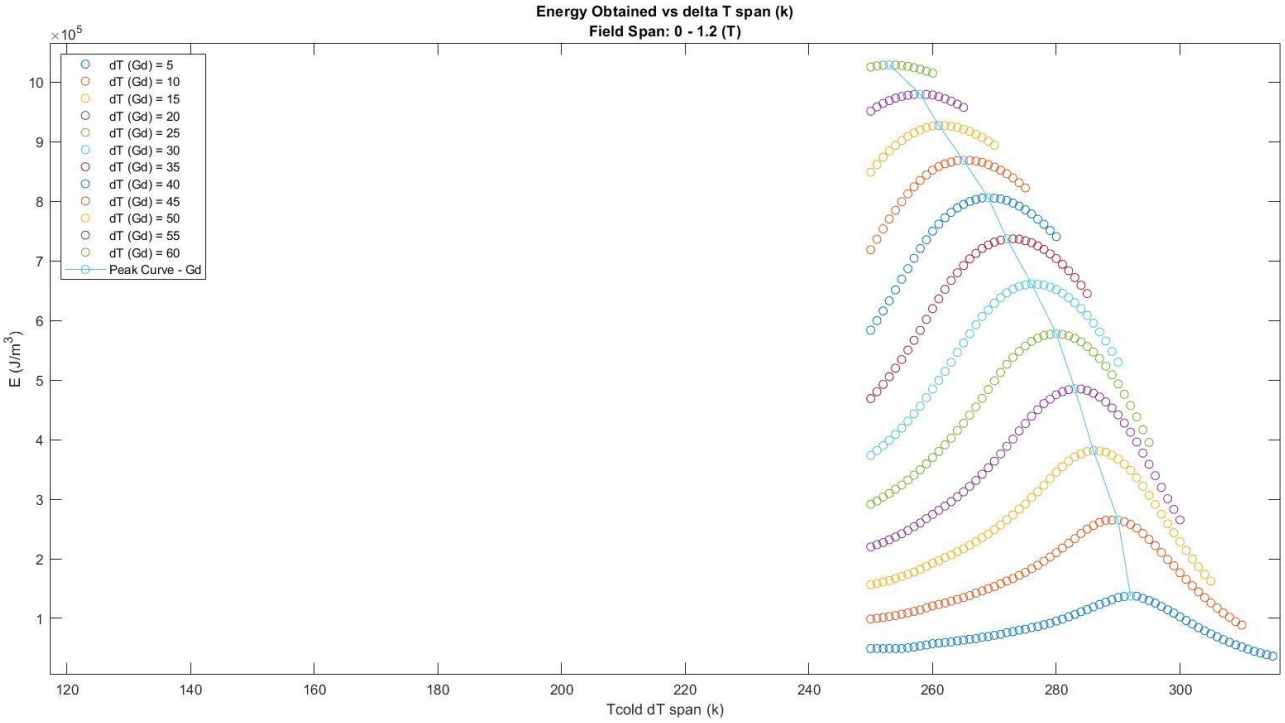
It is worth to consider also how the field span affects the energy per volume output for the various materials:



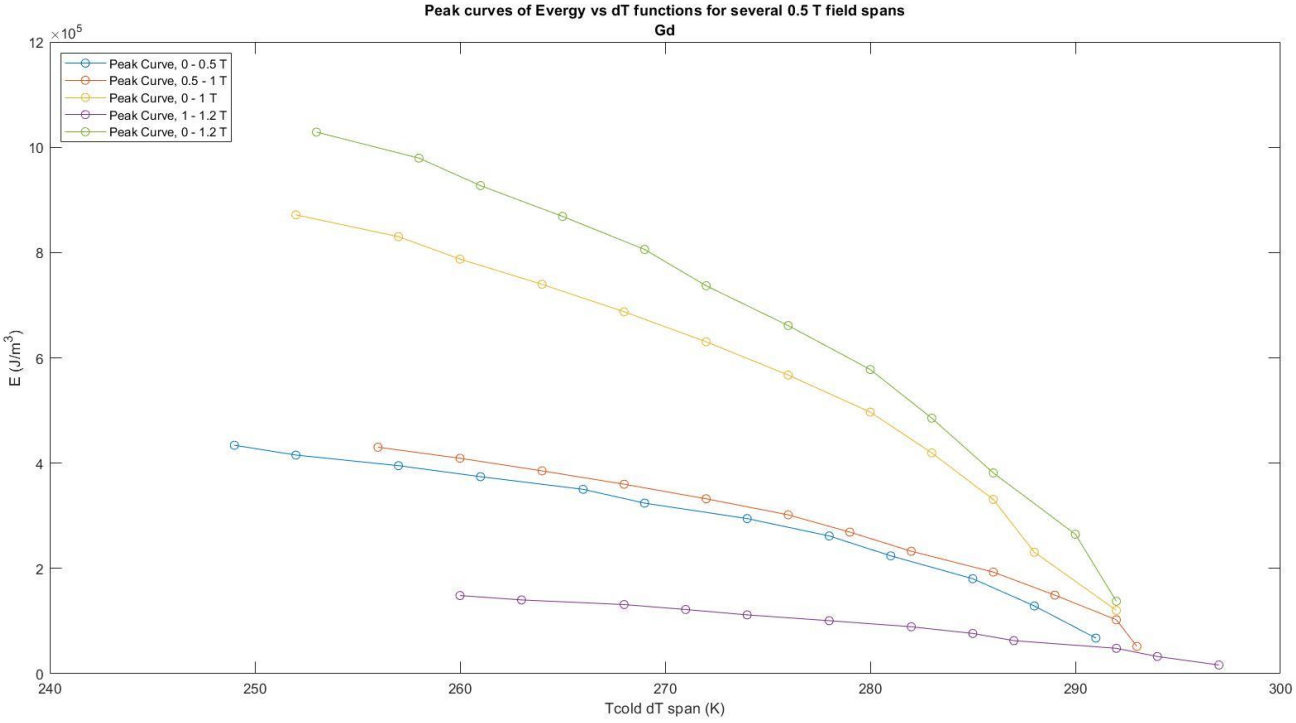



Energy Obtained vs delta T span (k)
Field Span: 1 - 1.2 (T)
×10^4
dT (Gd) = 5
dT (Gd) = 10
dT (Gd) = 15
dT (Gd) = 20
dT (Gd) = 25
dT (Gd) = 30
dT (Gd) = 35
dT (Gd) = 40
dT (Gd) = 45
dT (Gd) = 50
dT (Gd) = 55
dT (Gd) = 60
Peak Curve - Gd
E (J/m^3)
Tcold dT span (k)

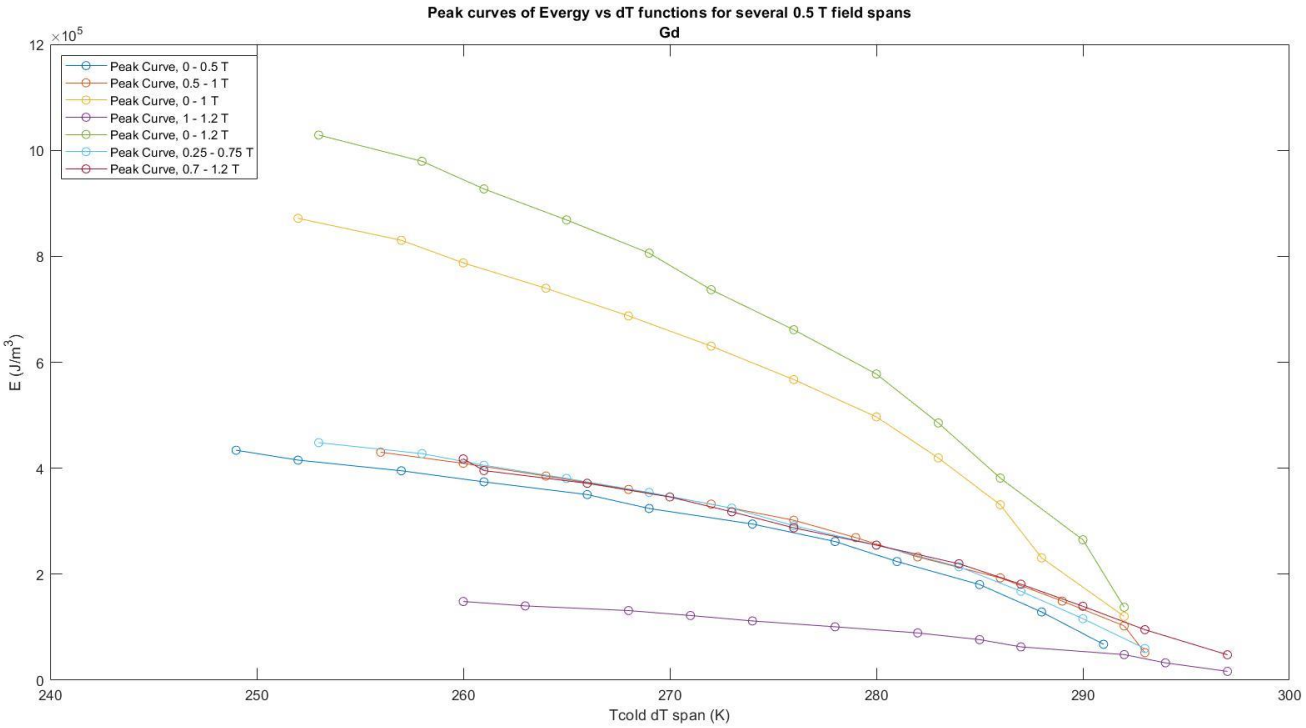

Energy Obtained vs delta T span (k)
Field Span: 0 - 1 (T)
×10^5
dT (Gd) = 5
dT (Gd) = 10
dT (Gd) = 15
dT (Gd) = 20
dT (Gd) = 25
dT (Gd) = 30
dT (Gd) = 35
dT (Gd) = 40
dT (Gd) = 45
dT (Gd) = 50
dT (Gd) = 55
dT (Gd) = 60
Peak Curve - Gd
E (J/m^3)
Tcold dT span (k)

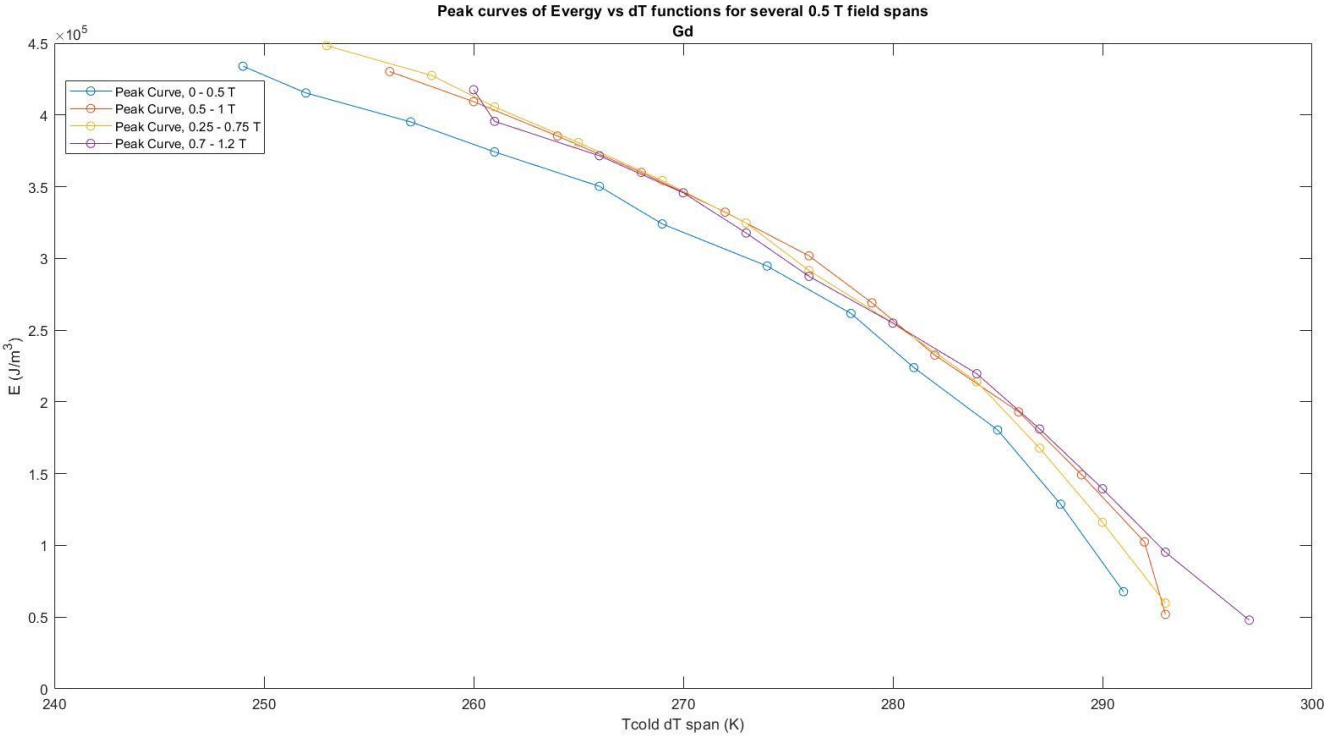


It is useful to plot all the peak curves together to show a better and concise way of comparing its behavior for the different field spans:

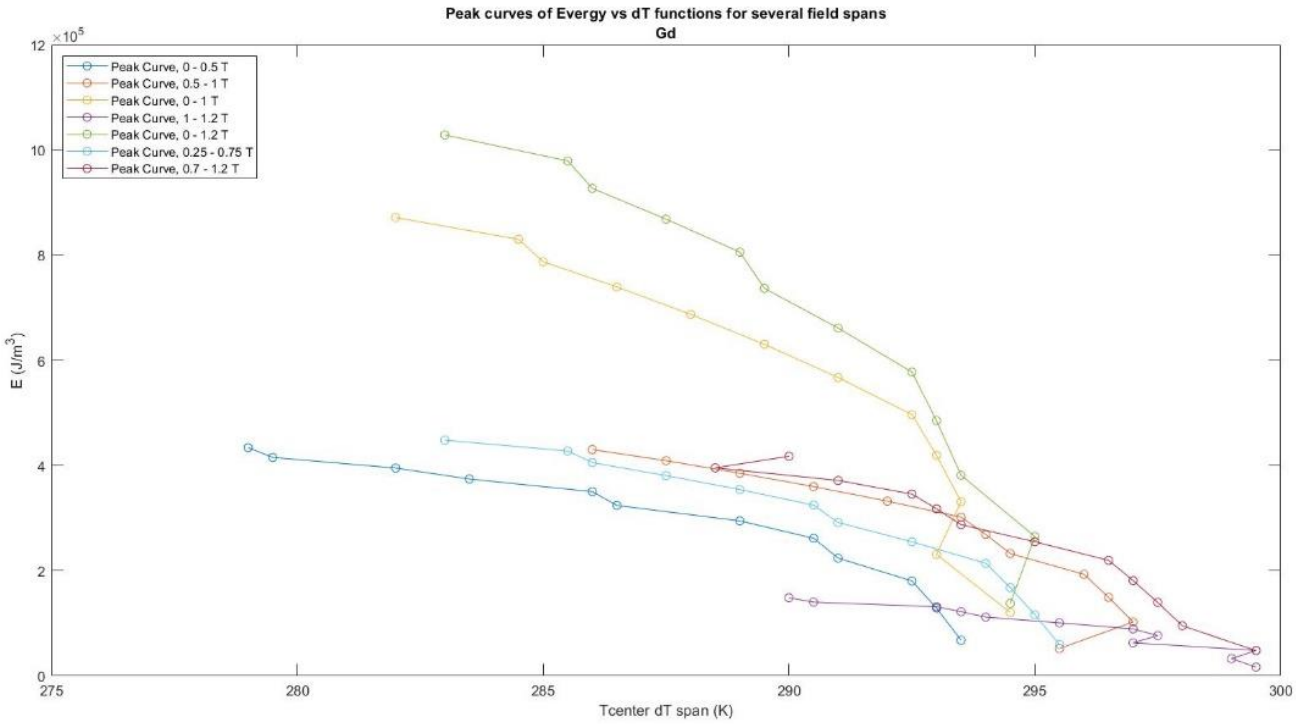


The direction towards lower temperatures (right to left) indicates increasing temperature span (dT); the last value to the left is the maximum in the ranges considered. It is worth noting that the curves are steeper the higher the field span, thus resulting in bigger energy per volume output. It is also noticeable the little difference between the peak curve for 0 – 0.5 T and 0.5 – 1 T. This is important because it could mean that the energy per volume outcome is nearly the same for a given range of field, despite of



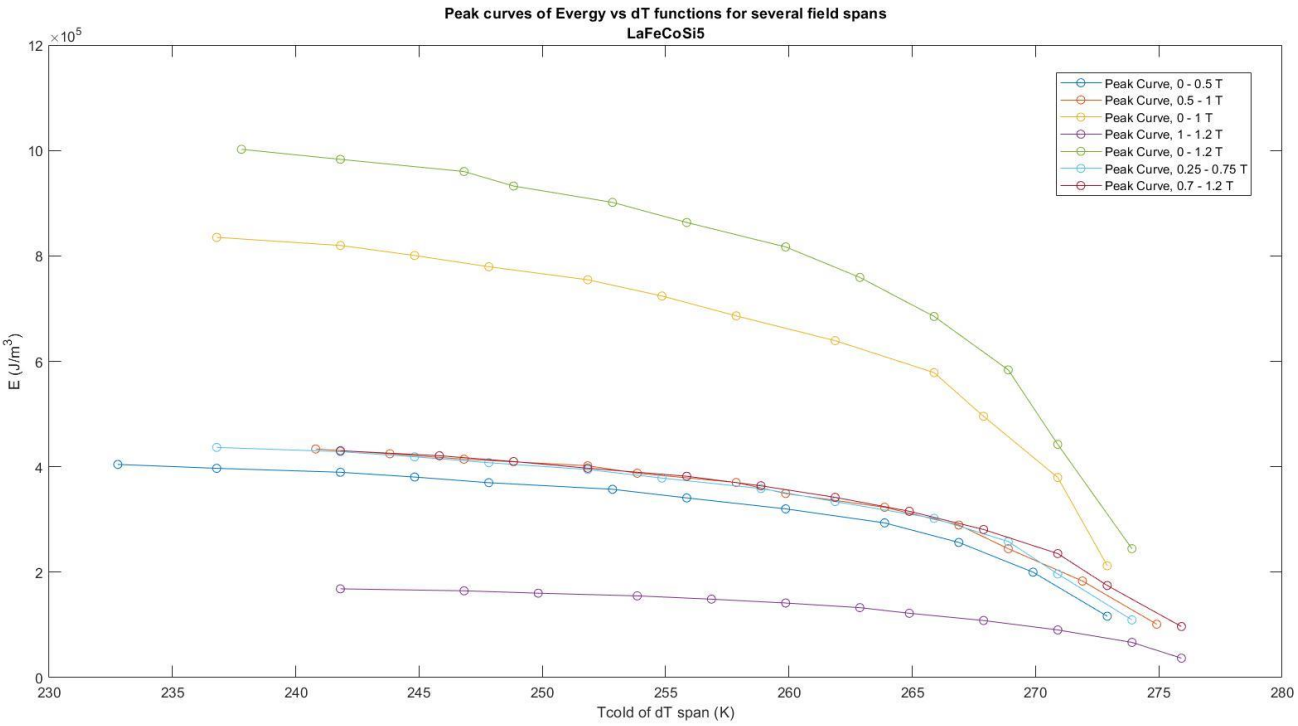


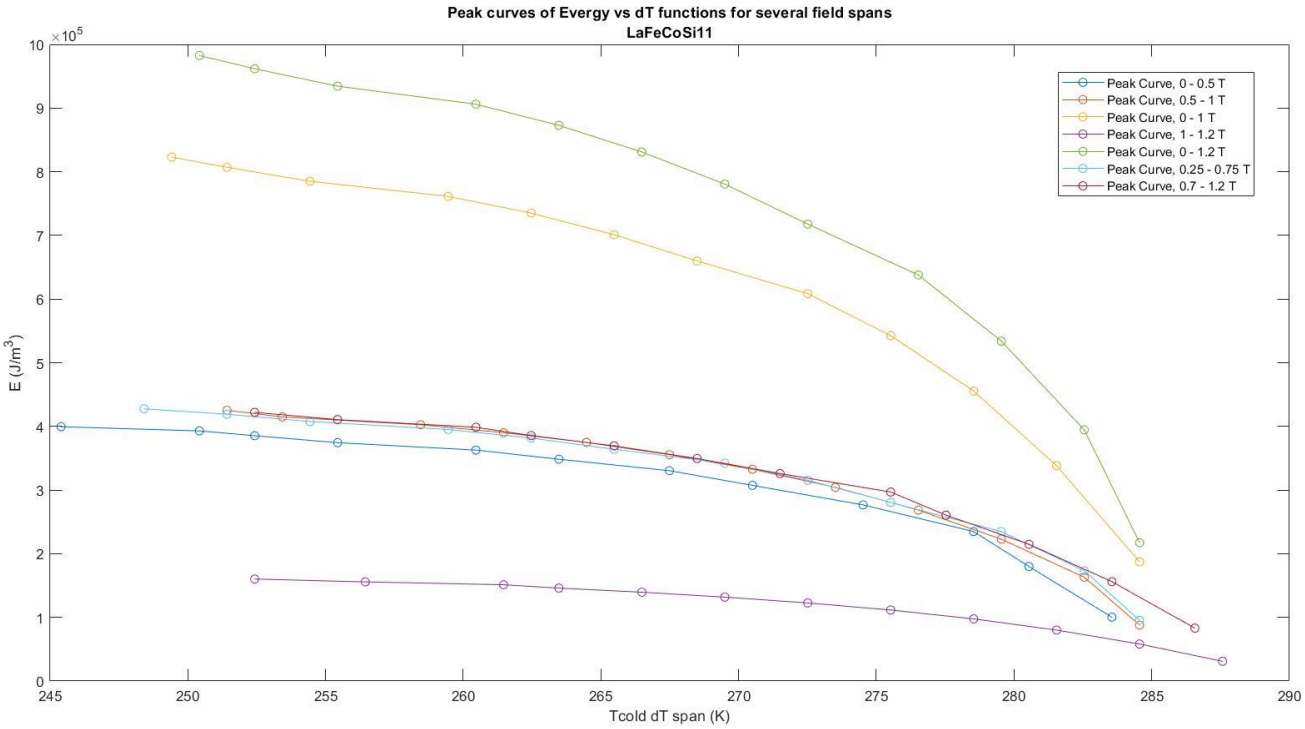
where the range is spanning from. Thus, similar results in energy production would be obtained without the need to worry about the field span location. To study this, more 0.5 T spans are considered:

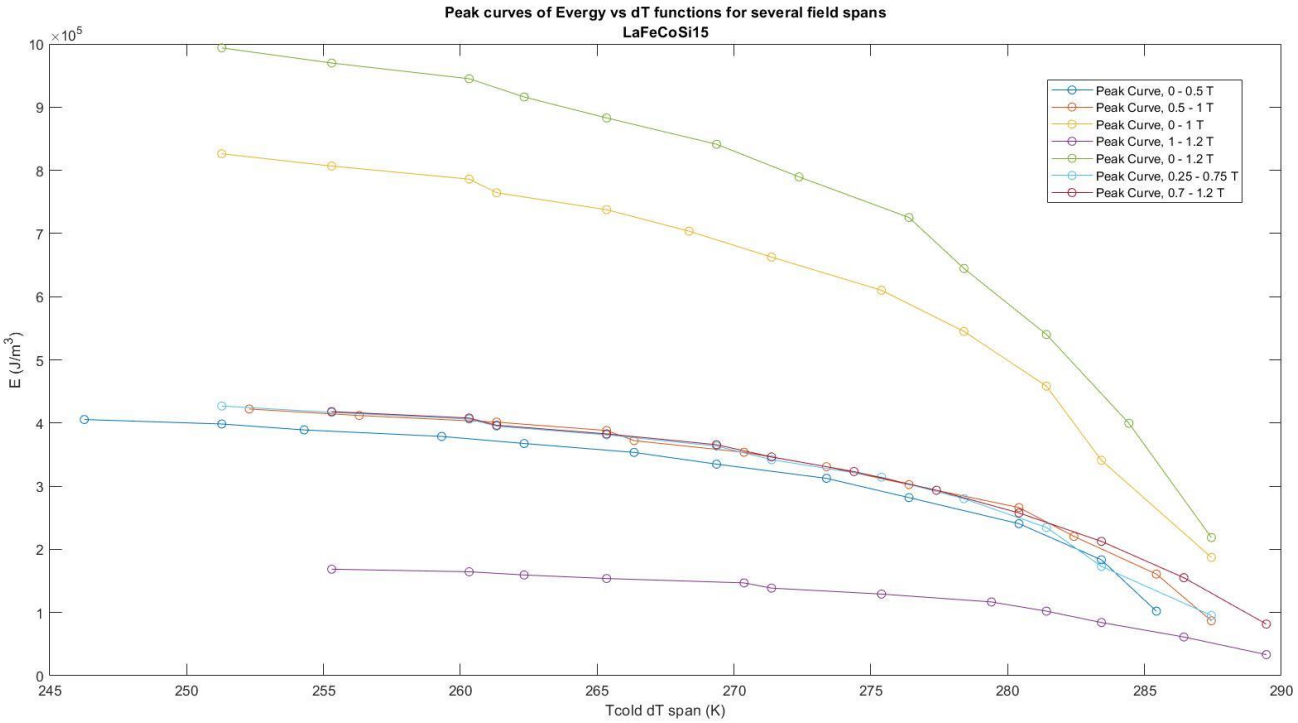


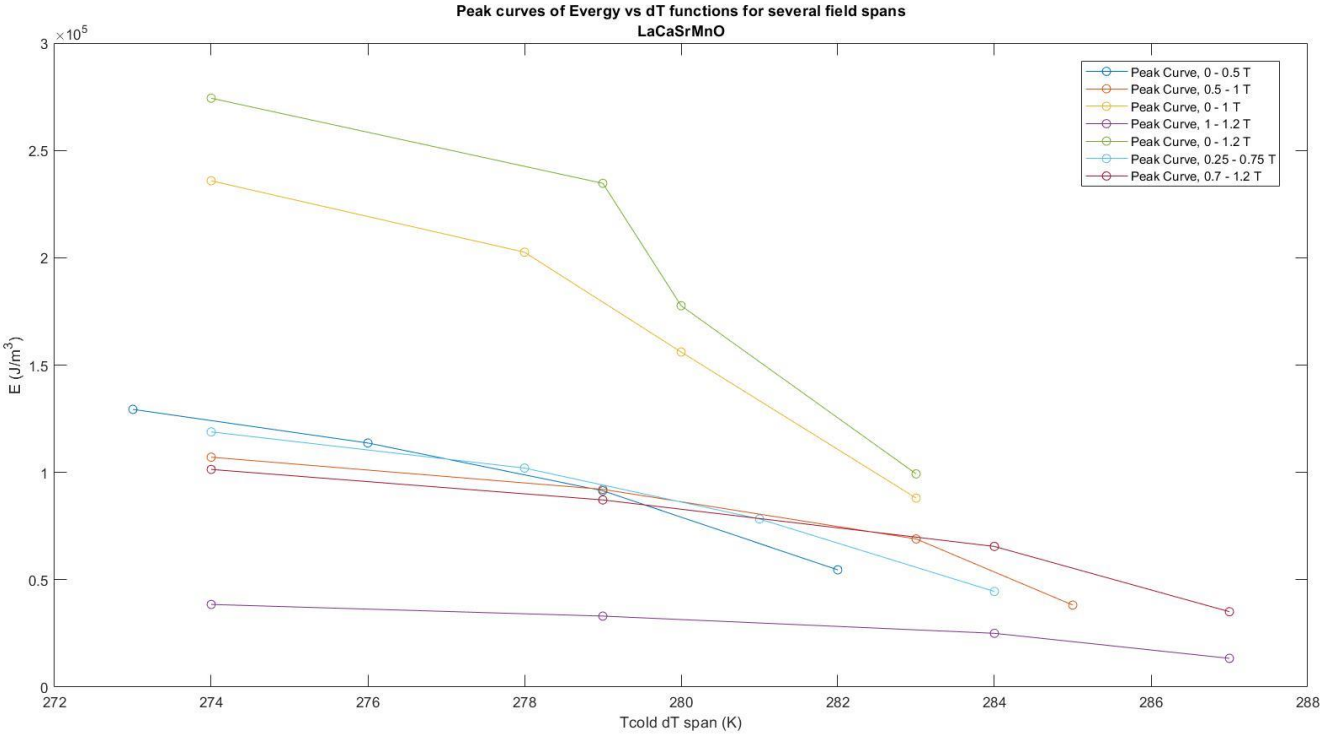
With Tcenter is more difficult to see the similarities but note that the horizontally, when coparing heights of peaks points, the energy obtained is similar for the same ranges of 0.5 T.

Now for the other materials:

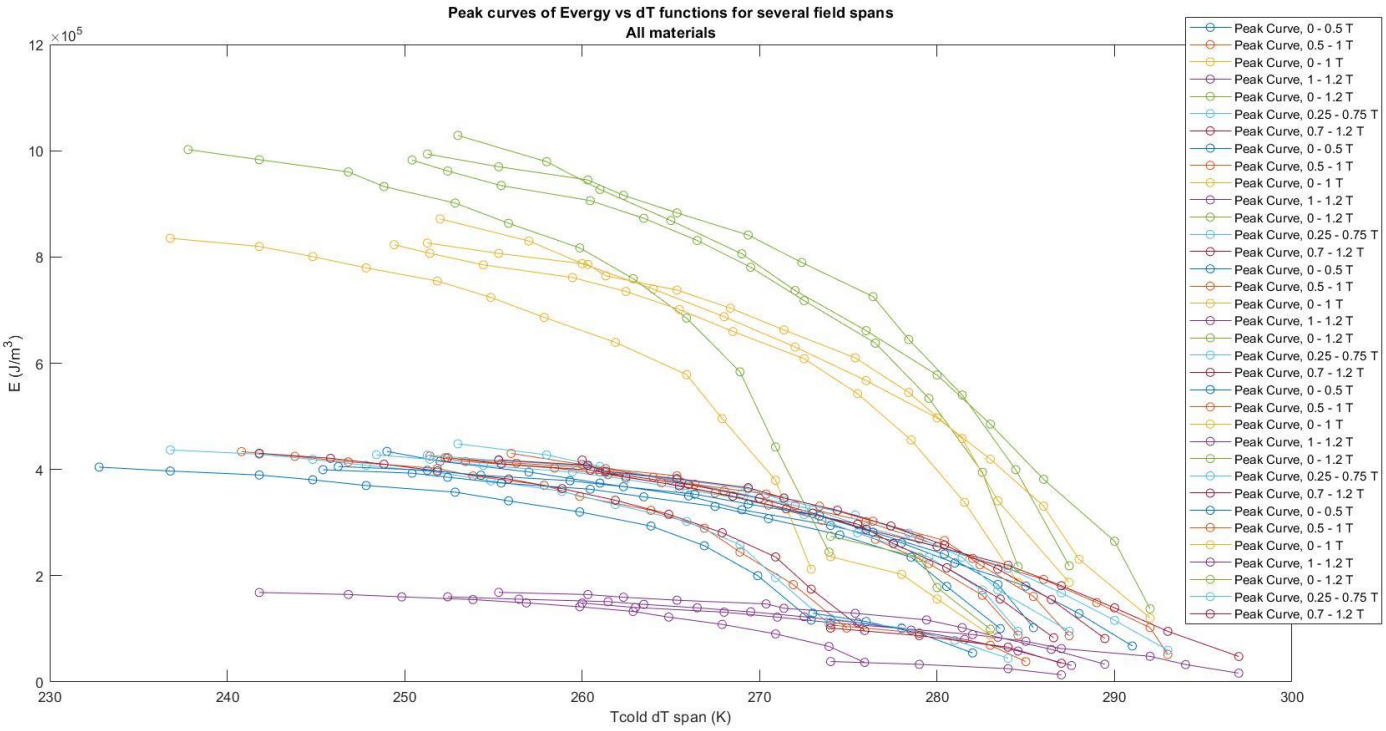

Peak curves of Evergy vs dT functions for several field spans
LaFeCoSi5
×10^5
E (J/m^3)
Tcold of dT span (K)
Peak Curve, 0 - 0.5 T
Peak Curve, 0.5 - 1 T
Peak Curve, 0 - 1 T
Peak Curve, 1 - 1.2 T
Peak Curve, 0 - 1.2 T
Peak Curve, 0.25 - 0.75 T
Peak Curve, 0.7 - 1.2 T


Peak curves of Evergy vs dT functions for several field spans
LaFeCoSi11
×10^5
E (J/m^3)
Tcold dT span (K)
Peak Curve, 0 - 0.5 T
Peak Curve, 0.5 - 1 T
Peak Curve, 0 - 1 T
Peak Curve, 1 - 1.2 T
Peak Curve, 0 - 1.2 T
Peak Curve, 0.25 - 0.75 T
Peak Curve, 0.7 - 1.2 T


Peak curves of Evergy vs dT functions for several field spans
LaFeCoSi15
×10^5
E (J/m^3)
Tcold dT span (K)
Peak Curve, 0 - 0.5 T
Peak Curve, 0.5 - 1 T
Peak Curve, 0 - 1 T
Peak Curve, 1 - 1.2 T
Peak Curve, 0 - 1.2 T
Peak Curve, 0.25 - 0.75 T
Peak Curve, 0.7 - 1.2 T

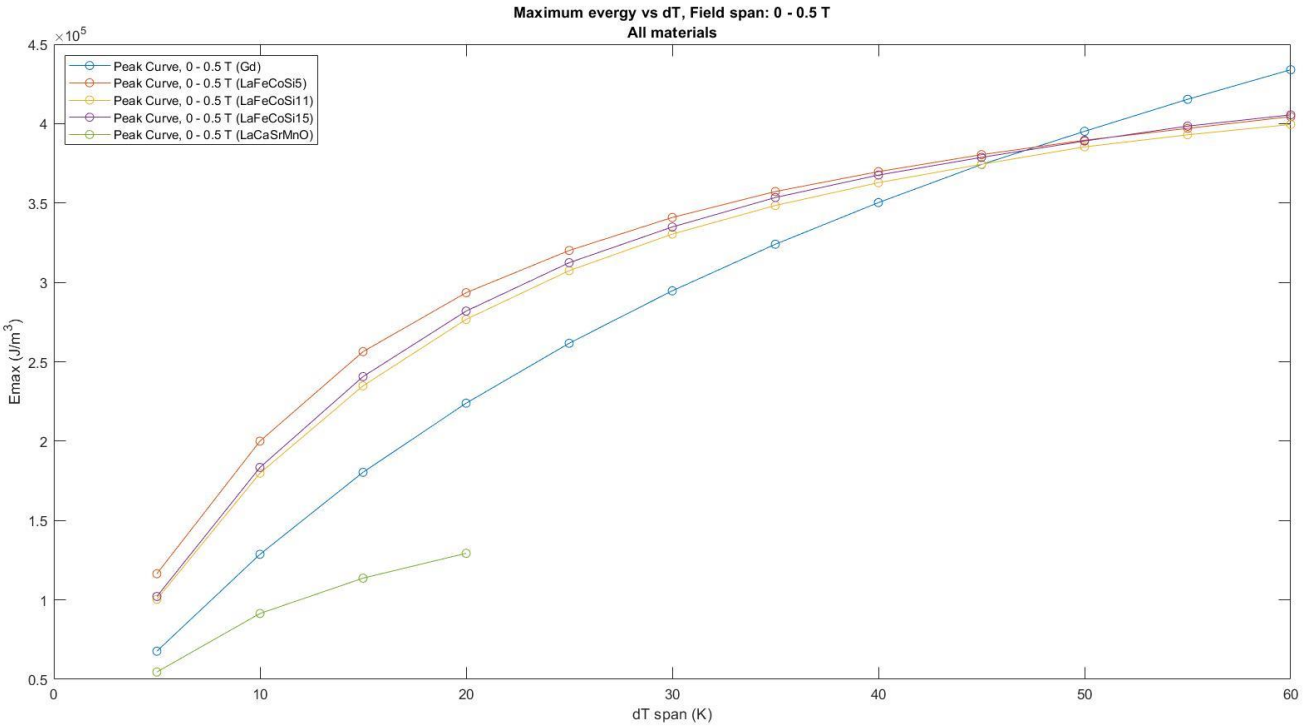

Peak curves of Evergy vs dT functions for several field spans
LaCaSrMnO
×10^5
E (J/m^3)
Tcold dT span (K)
Peak Curve, 0 - 0.5 T
Peak Curve, 0.5 - 1 T
Peak Curve, 0 - 1 T
Peak Curve, 1 - 1.2 T
Peak Curve, 0 - 1.2 T
Peak Curve, 0.25 - 0.75 T
Peak Curve, 0.7 - 1.2 T

All materials follow the same general trends. This can be seen when plotted all together:

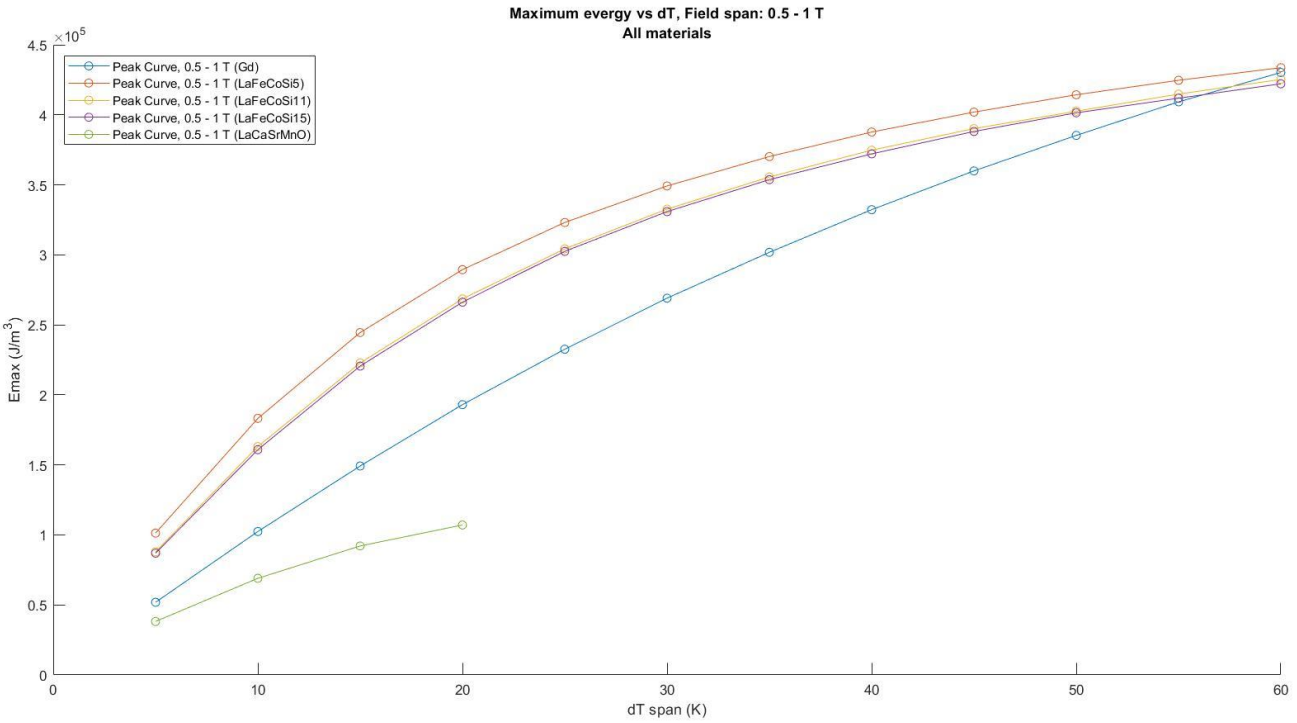


From the previous plot where all materials are included, it can be seen that **the span of field does not have a noticeable impact on the peaks of energy per volume production** (this is clearly seen when considering the 0.5 T field span studied before). It does have, however, an impact on the temperatures at which the peaks take place, as LaFeCoSi5 clearly demands lower peak temperatures.

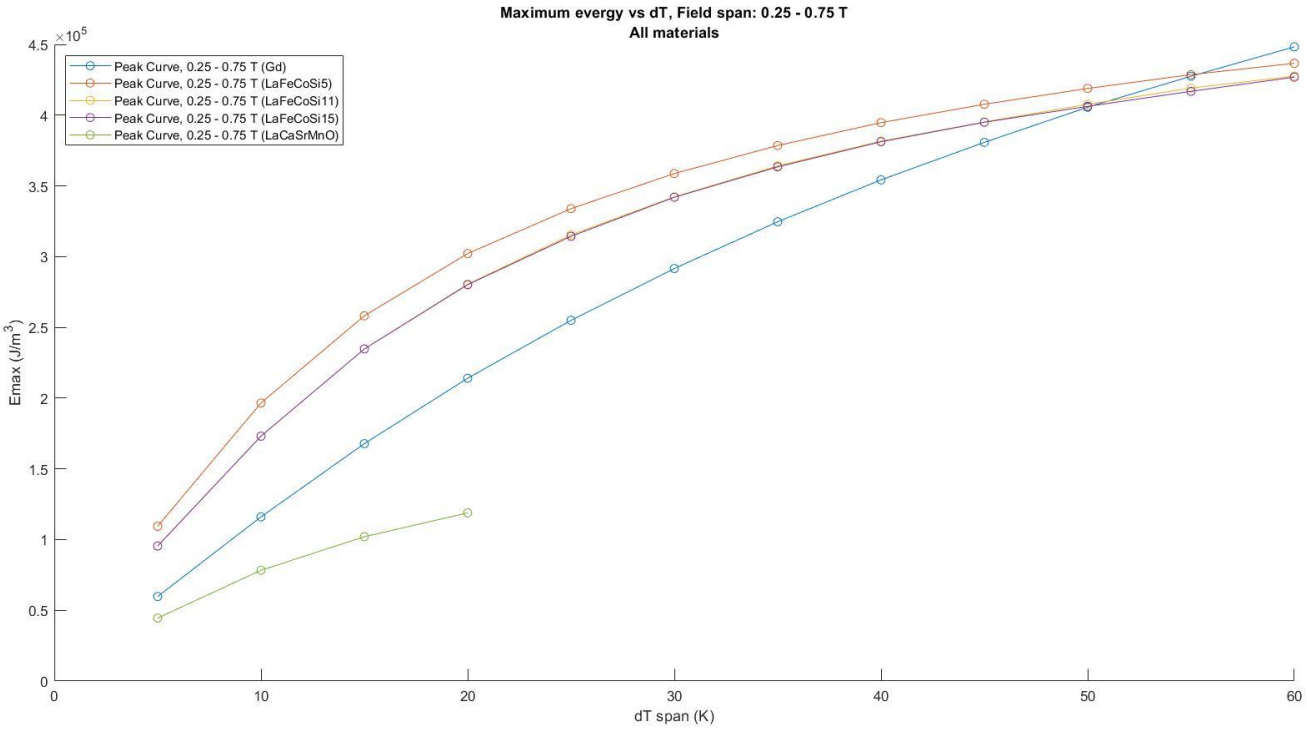
APPENDIX C – MAXIMUM ENERGY PER VOLUME OUTPUT FOR EVERY TEMPERATURE SPAN. ALL MATERIALS AND DIFFERENT FIELD CONSIDERED



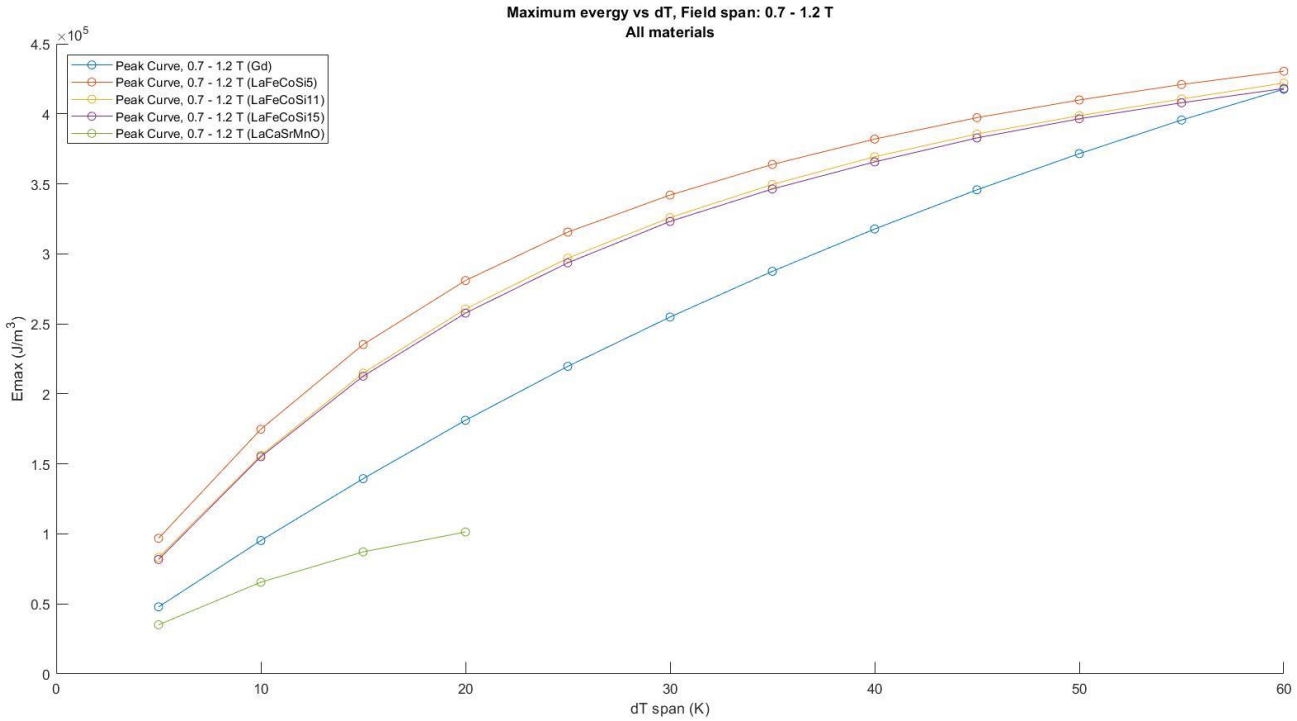
Notice Gadolinium outperforming LaFeCoSi5 for s temperature span lower than 50 K.



It can be seen that for a span starting from 0.5 T to 1 T Gadolinium has still not outmatched LaFeCoSi5 at a temperature span of 60 K, the surpass point moves with the localization of the filed span. **This indicates that the localization of the filed span does matter to choose the best performing material for large temperature spans**.

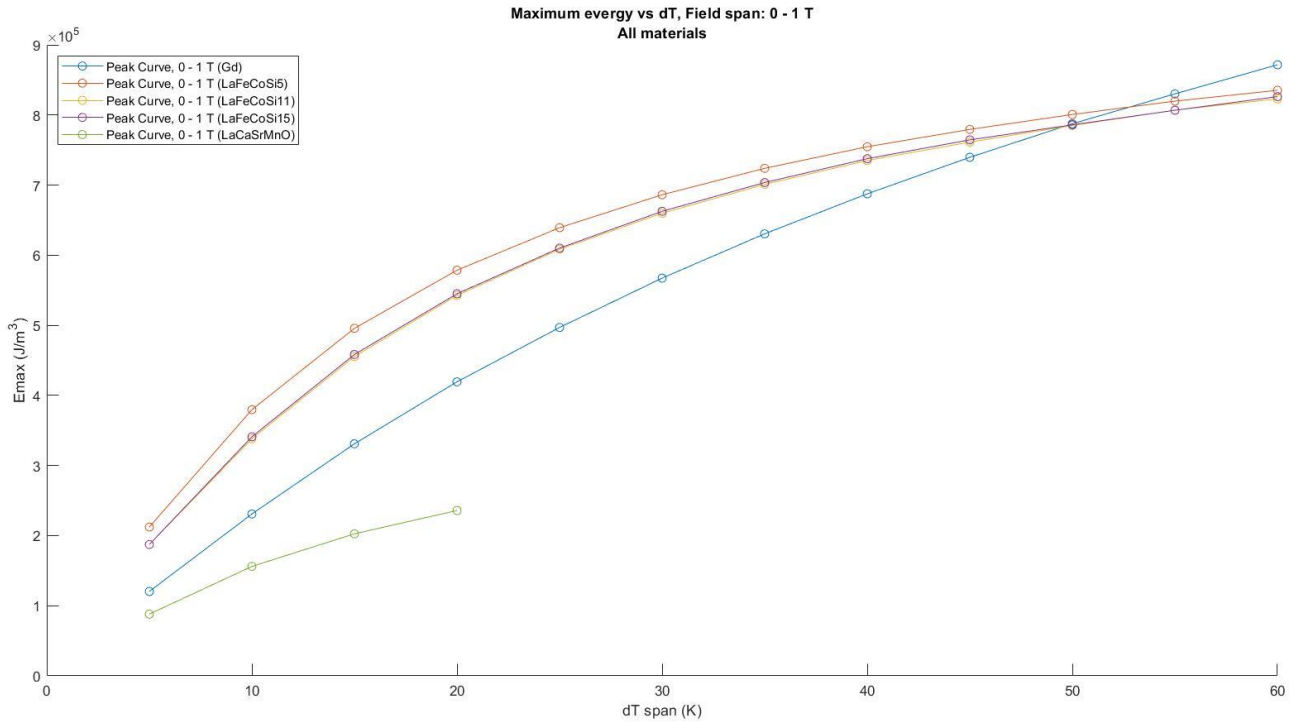


See that for a field spanning from 0.25 T to 0.75 T the surpass point of Gd over LaFeCoSi5 is past the 55 K temperature span. It has moved to the right, towards higher temperatures spans.

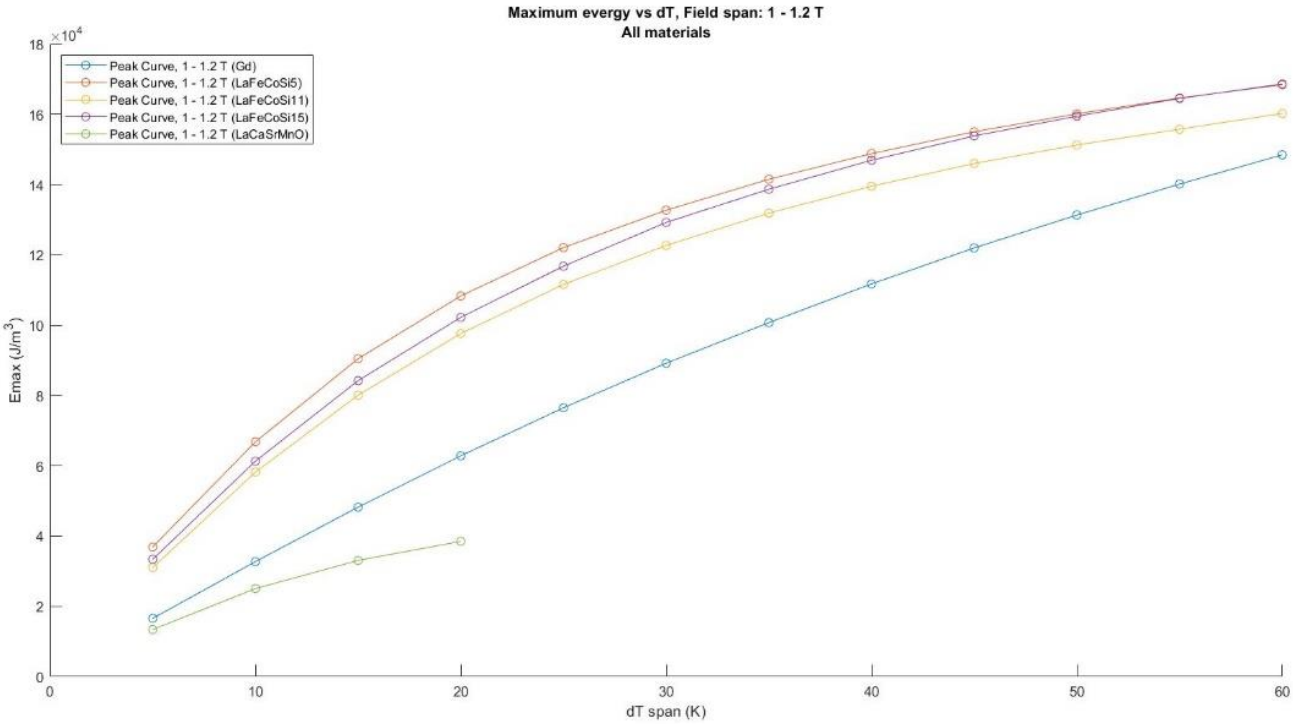


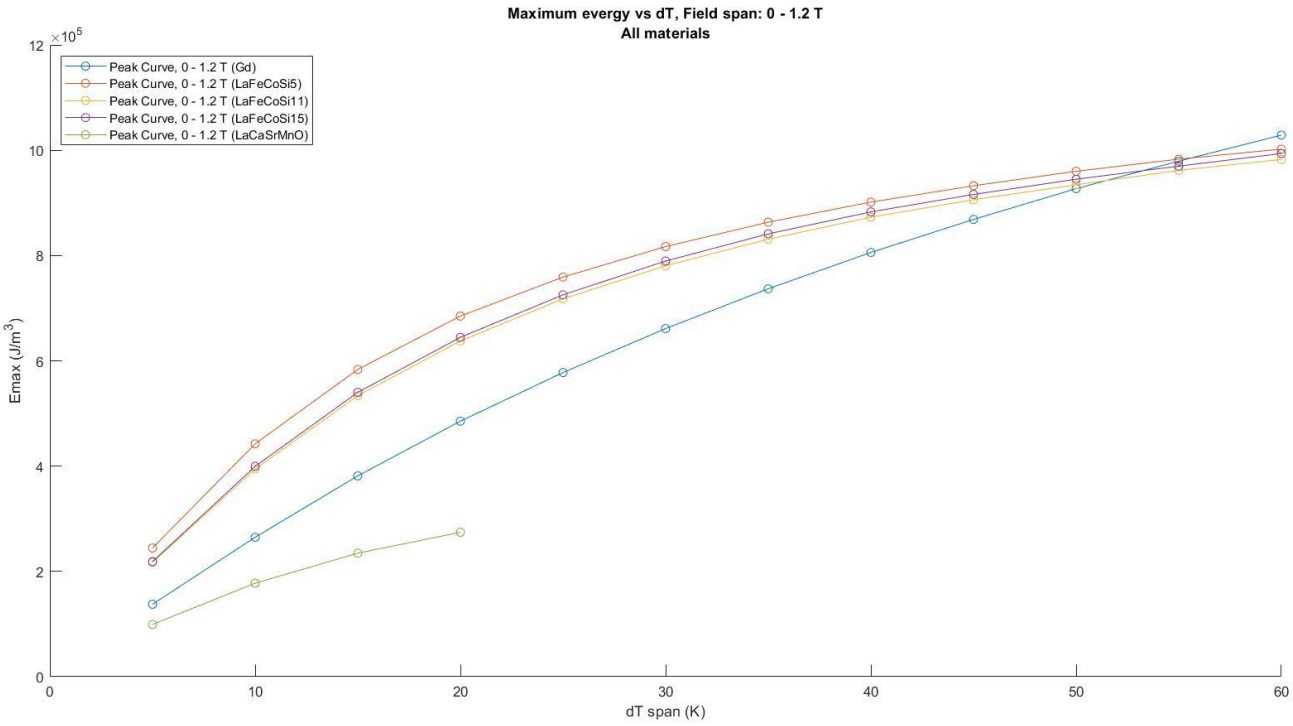
The surpass point keeps moving in the same direction when the fixed-length field span is moved on to higher field values. Thus, the tendency demanding bigger temperatures spans when the same field range is located at higher values is confirmed: **the lower a fixed-length field span is set, the lower the temperature span dT at which Gadolinium surpasses LaFeCoSi5 maximum energy per volume production will be.**

For the oher rages considered:



[above - studied before]




Maximum evergy vs dT, Field span: 0 - 1.2 T
All materials
×10^5
Peak Curve, 0 - 1.2 T (Gd)
Peak Curve, 0 - 1.2 T (LaFeCoSi5)
Peak Curve, 0 - 1.2 T (LaFeCoSi11)
Peak Curve, 0 - 1.2 T (LaFeCoSi15)
Peak Curve, 0 - 1.2 T (LaCaSrMnO)
Emax (J/m^3)
dT span (K)
0
2
4
6
8
10
12
0
10
20
30
40
50
60

APPENDIX D – MATLAB CODES

E_vs_H_Gd

```
load Gdcommmagnetizationtableconvertedhcorrected2.mat

Gd_m=table2array(Gdcommmagnetizationtableconvertedhcor
rected2);
M_280=Gd_m(:,22);
M_300=Gd_m(:,42);
H=Gd_m(:,1);
limit=129;
Integral=0; %Area under curve
square=0;
mu=4*pi*10^-7;

plot(H,M_300,'-b')
hold on
plot(H,M_280,'-r')
hold off
EE=zeros(129,1);
H_span=zeros(129,1); %vector od Hspan for plot

for i=1:129
    if i==1
         H_span(i)=H(1);
    else
         H_span(i)=H(i)-H(1);
    end
end

%varying limit=H:
figure()
for limit=1:129
    Integral=0; %if we don't initialize for every
Hspan (for every limit count in greaer or loop then we
would add integral of the previous H span to nect
Hspan
    square=0;

    %integral:
    %for idx=1:length(H) -> must not take whole range
as there are NAN
    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
```

```
        square=(M_280(idx)*(H(idx+1)-H(idx)));
        Integral=Integral+square;
    end

    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
        square=(M_300(idx)*(H(idx+1)-H(idx)));
        Integral=Integral-square;
    end

    %Integral is now the net area of A280-A300. We can
convert this net area to
    %energy to obtain the energy generated in the
cycle: E=mu*A

    E=mu*Integral; %E(J/m^3)
    EE(limit)=E;   %EE is a vector containing all
values of E for each time the script is calculated
according to limit=Hspan value
    plot(H_span(limit),EE(limit),'or')
    xlabel('H span (A/m)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs H Span (Gd)');
    hold on
end
```

E_vs_H_LaFeCoSi5

```
load LaFeCoSi5magnetizationtable.mat

Gd_m=table2array(LaFeCoSi5magnetizationtable);
M_280=Gd_m(:,33);
M_300=Gd_m(:,44);
H=Gd_m(:,1);
limit=129;
Integral=0; %Area under curve
square=0;
mu=4*pi*10^-7;

plot(H,M_300,'-b')
hold on
plot(H,M_280,'-r')
hold off
EE=zeros(129,1);
H_span=zeros(129,1); %vector od Hspan for plot

for i=1:129
```

```
    if i==1
         H_span(i)=H(1);
    else
         H_span(i)=H(i)-H(1);
    end
end

%varying limit=H:
figure()
for limit=1:129
    Integral=0; %if we don't initialize for every
Hspan (for every limit count in greaer or loop then we
would add integral of the previous H span to nect
Hspan
    square=0;

    %integral:
    %for idx=1:length(H) -> must not take whole range
as there are NAN
    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
       square=(M_280(idx)*(H(idx+1)-H(idx)));
       Integral=Integral+square;
    end

    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
       square=(M_300(idx)*(H(idx+1)-H(idx)));
       Integral=Integral-square;
    end

    %Integral is now the net area of A280-A300. We can
convert this net area to
    %energy to obtain the energy generated in the
cycle: E=mu*A

    E=mu*Integral; %E(J/m^3)
    EE(limit)=E;   %EE is a vector containing all
values of E for each time the script is calculated
according to limit=Hspan value
    plot(H_span(limit),EE(limit),'or')
    xlabel('H span (A/m)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs H Span (LaFeCoSi5)');
    hold on
end
```

E_vs_H_LaFeCoSi11

```
load LaFeCoSi11magnetizationtable.mat

Gd_m=table2array(LaFeCoSi11magnetizationtable);
M_280=Gd_m(:,27);
M_300=Gd_m(:,48);
H=Gd_m(:,1);
limit=129;
Integral=0; %Area under curve
square=0;
mu=4*pi*10^-7;

plot(H,M_300,'-b')
hold on
plot(H,M_280,'-r')
hold off
EE=zeros(129,1);
H_span=zeros(129,1); %vector od Hspan for plot

for i=1:129
    if i==1
         H_span(i)=H(1);
    else
         H_span(i)=H(i)-H(1);
    end
end

%varying limit=H:
figure()
for limit=1:129
    Integral=0; %if we don't initialize for every
Hspan (for every limit count in greaer or loop then we
would add integral of the previous H span to nect
Hspan
    square=0;

    %integral:
    %for idx=1:length(H) -> must not take whole range
as there are NAN
    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
       square=(M_280(idx)*(H(idx+1)-H(idx)));
       Integral=Integral+square;
    end

    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
```

```
        square=(M_300(idx)*(H(idx+1)-H(idx)));
        Integral=Integral-square;
    end

    %Integral is now the net area of A280-A300. We can
convert this net area to
    %energy to obtain the energy generated in the
cycle: E=mu*A

    E=mu*Integral; %E(J/m^3)
    EE(limit)=E;   %EE is a vector containing all
values of E for each time the script is calculated
according to limit=Hspan value
    plot(H_span(limit),EE(limit),'or')
    xlabel('H span (A/m)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs H Span (LaFeCoSi11)');
    hold on
end
```

E_vs_H_LaFeCoSi15

```
load LaFeCoSi15magnetizationtable.mat

Gd_m=table2array(LaFeCoSi15magnetizationtable);
M_280=Gd_m(:,21);
M_300=Gd_m(:,41);
H=Gd_m(:,1);
limit=129;
Integral=0; %Area under curve
square=0;
mu=4*pi*10^-7;

plot(H,M_300,'-b')
hold on
plot(H,M_280,'-r')
hold off
EE=zeros(129,1);
H_span=zeros(129,1); %vector od Hspan for plot

for i=1:129
    if i==1
         H_span(i)=H(1);
    else
         H_span(i)=H(i)-H(1);
    end
end
```

```
%varying limit=H:
figure()
for limit=1:129
    Integral=0; %if we don't initialize for every
Hspan (for every limit count in greaer or loop then we
would add integral of the previous H span to nect
Hspan
    square=0;

    %integral:
    %for idx=1:length(H) -> must not take whole range
as there are NAN
    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
       square=(M_280(idx)*(H(idx+1)-H(idx)));
       Integral=Integral+square;
    end

    for idx=1:limit  %limit is to avoid NAN, could be
different for every tmeperature span
       square=(M_300(idx)*(H(idx+1)-H(idx)));
       Integral=Integral-square;
    end

    %Integral is now the net area of A280-A300. We can
convert this net area to
    %energy to obtain the energy generated in the
cycle: E=mu*A

    E=mu*Integral; %E(J/m^3)
    EE(limit)=E;   %EE is a vector containing all
values of E for each time the script is calculated
according to limit=Hspan value
    plot(H_span(limit),EE(limit),'or')
    xlabel('H span (A/m)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs H Span');
    hold on
end
```

Harvest_integration_Gd_E_vs_H_V2

```
%V2 => span centered in Curie temperature, better
plots
clear all;
close all;
```

```
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

starttemp = 283; %280; %100;
tempIndStart = find(tempInt == starttemp);
endtemp = 303; %300;
tempIndEnd = find(tempInt == endtemp);
%startfield = 0.01;
startfield = 7957.74700000000;
fieldIndStart = find(fieldInt == startfield);
endfield = 799633.017000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
fieldIndEnd = find(fieldInt == endfield);
fieldIndEnd = fieldIndEnd-1; %This is to avoid taking
NAN numbers in trapz function later, which then gives
NAN in return

txt1 = ['Tcold: ',num2str(starttemp),' K'];
txt2 = ['Thot: ',num2str(endtemp),' K'];
plot(fieldInt,dataInt(:,tempIndStart),'-
b','DisplayName',txt1)
hold on
plot(fieldInt,dataInt(:,tempIndEnd),'-
r','DisplayName',txt2)
xlabel('H span (A/m)');
ylabel('M (A/m)');
title({'Magnetization vs H span (Gd)','H span = 0 - 1
T'});
legend('Location','NorthWest')
```

```
H_spacing=fieldInt(2)-fieldInt(1);
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

Energy = (LowInt_3-HighInt_3)*mu;

%In process vectors for plotting:
LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart:
fieldIndEnd,tempIndStart));
HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndEnd));

Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  % UNITS
NEED FIXING

figure()
plot(fieldInt(fieldIndStart:fieldIndEnd),Energy_cum,'o
r')
xlabel('H span (A/m)');
ylabel('E (J/m^3)');
title({'Energy Obtained vs H Span (Gd)','Temperature
span = 283 - 303 K','H span = 0 - 1 T'});
```

Harvest_integration_LaFeCoSi5_E_vs_H_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\varios_scripts\
LaFeCoSi5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);
```

```
dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

starttemp = 265.883; %279.927; %280 aprox
tempIndStart = find(tempInt == starttemp);
endtemp = 285.945; %299.988; %300 aprox
tempIndEnd = find(tempInt == endtemp);
%startfield = 0.01;
startfield = 7957.74700000000; %
fieldIndStart = find(fieldInt == startfield);
endfield = 800195.686000000; %we should take
1T=795774.7155 but we are taking 800195.686 which is
the closest value on new table fieldInt.
fieldIndEnd = find(fieldInt == endfield);
%fieldIndEnd = fieldIndEnd-1; %This is to avoid taking
NAN numbers in trapz function later, which then gives
NAN in return

txt1 = ['Tcold: 266 K']; %num2str(starttemp),' K'];
txt2 = ['Thot: 286 K']; %',num2str(endtemp),' K'];
plot(fieldInt,dataInt(:,tempIndStart),'-
b','DisplayName',txt1)
hold on
plot(fieldInt,dataInt(:,tempIndEnd),'-
r','DisplayName',txt2)
xlabel('H span (A/m)');
ylabel('M (A/m)');
title({'Magnetization vs H span (LaFeCoSi5)','H span =
0 - 1 T'});
legend('Location','NorthWest')

H_spacing=fieldInt(2)-fieldInt(1);
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

Energy = (LowInt_3-HighInt_3)*mu;

%In process vectors for plotting:
LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart:
fieldIndEnd,tempIndStart));
HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndEnd));
```

```
Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  % UNITS
NEED FIXING

figure()
plot(fieldInt(fieldIndStart:fieldIndEnd),Energy_cum,'o
r')
xlabel('H span (A/m)');
ylabel('E (J/m^3)');
title({'Energy Obtained vs H Span
(LaFeCoSi5)','Temperature span = 266 - 286 K','H span
= 0 - 1 T'});
```

Harvest_integration_LaFeCoSi11_E_vs_H_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\varios_scripts\
LaFeCoSi11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;  => table already in
A/m
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

starttemp = 271.504; %279.538; %280 aprox
tempIndStart = find(tempInt == starttemp);
endtemp = 291.588; %299.621; %300 aprox
tempIndEnd = find(tempInt == endtemp);
%startfield = 0.01;
startfield = 7957.74700000000; %
fieldIndStart = find(fieldInt == startfield);
```

```
endfield = 799633.017; %we should take 1T=795774.7155
but we are taking 799633.017 which is the closest
value on new table fieldInt.
fieldIndEnd = find(fieldInt == endfield);
%fieldIndEnd = fieldIndEnd-1; %This is to avoid taking
NAN numbers in trapz function later, which then gives
NAN in return

txt1 = ['Tcold: 272 K']; %num2str(starttemp),' K'];
txt2 = ['Thot: 292 K']; %',num2str(endtemp),' K'];
plot(fieldInt,dataInt(:,tempIndStart),'-
b','DisplayName',txt1)
hold on
plot(fieldInt,dataInt(:,tempIndEnd),'-
r','DisplayName',txt2)
xlabel('H span (A/m)');
ylabel('M (A/m)');
title({'Magnetization vs H span (LaFeCoSi11)','H span
= 0 - 1 T'});
legend('Location','NorthWest')

H_spacing=fieldInt(2)-fieldInt(1);
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

Energy = (LowInt_3-HighInt_3)*mu;
%In process vectors for plotting:
LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart:
fieldIndEnd,tempIndStart));
HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndEnd));

Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;
figure()
plot(fieldInt(fieldIndStart:fieldIndEnd),Energy_cum,'o
r')
xlabel('H span (A/m)');
ylabel('E (J/m^3)');
title({'Energy Obtained vs H Span
(LaFeCoSi11)','Temperature span = 272 - 292 K','H span
= 0 - 1 T'});
```

Harvest_integration_LaFeCoSi15_E_vs_H_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\varios_scripts\
LaFeCoSi15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

starttemp = 278.401; %280.41; %280 aprox
tempIndStart = find(tempInt == starttemp);
endtemp = 298.491; %300.5; %300 aprox
tempIndEnd = find(tempInt == endtemp);
%startfield = 0.01;
startfield = 7957.74700000000; %
fieldIndStart = find(fieldInt == startfield);
endfield = 792880.989; %we should take 1T=795774.7155
but we are taking 792880.989 which is the closest
value on new table fieldInt.
fieldIndEnd = find(fieldInt == endfield);
```

Harvest_integration_LaCaSrMnO_E_vs_H_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
```

```
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

starttemp = 273.999;
tempIndStart = find(tempInt == starttemp);
endtemp = 294;
tempIndEnd = find(tempInt == endtemp);
%startfield = 0.01;
startfield = -649.880000000000;%11965.7180000000;
%closest value to zero,this parameter can be changed
fieldIndStart = find(fieldInt == startfield);
endfield = 794132.790000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
fieldIndEnd = find(fieldInt == endfield);
fieldIndEnd = fieldIndEnd-1; %This is to avoid taking
NAN numbers in trapz function later, which then gives
NAN in return

txt1 = ['Tcold: 274 K']; %num2str(starttemp),' K'];
txt2 = ['Thot: 294 K']; %',num2str(endtemp),' K'];
plot(fieldInt,dataInt(:,tempIndStart),'-
b','DisplayName',txt1)
hold on
plot(fieldInt,dataInt(:,tempIndEnd),'-
r','DisplayName',txt2)
xlabel('H span (A/m)');
ylabel('M (A/m)');
title({'Magnetization vs H span (LaCaSrMnO)','H span =
0 - 1 T'});
legend('Location','NorthWest')

H_spacing=fieldInt(2)-fieldInt(1);
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
```

```
HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

Energy = (LowInt_3-HighInt_3)*mu;

%In process vectors for plotting:
LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart:
fieldIndEnd,tempIndStart));
HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndEnd));

Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;

figure()
plot(fieldInt(fieldIndStart:fieldIndEnd),Energy_cum,'o
r')
xlabel('H span (A/m)');
ylabel('E (J/m^3)');
title({'Energy Obtained vs H Span
(LaCaSrMnO)','Temperature span = 274 - 294 K','H span
= 0 - 1 T'});
```

Harvest_integration_Gd_E_vs_dT_V2

```
%V2 => better plot
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;
```

```
T_curie=293;%286;
tempIndCurie = find(tempInt == T_curie);
i=0;
n=length(tempInt)-tempIndCurie;
while i<n

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));
```

```
%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;


    plot(dT,Energy,'or')
    hold on
    xlabel('dT span (K)');
    ylabel('E (J/m^3)');
    txt=['Center of span = ',num2str(T_curie),' K'];
    title({'Energy Obtained vs delta T Span
(Gd)',txt});
i=i+1;
end
```

Harvest_integration_LaFeCoSi5_E_vs_dT_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=275.914; %281.933; %WE ARE ASSUMING Tcurie =
282 K -> no, is around 276K
tempIndCurie = find(tempInt == T_curie);
i=0;
n=length(tempInt)-tempIndCurie;
```

```
while i<n

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 800195.686000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  %

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %
```

```
    plot(dT,Energy,'or')
    hold on
    xlabel('dT span (K)');
    ylabel('E (J/m^3)');
    txt=['Center of span = ',num2str(T_curie),' K'];
    title({'Energy Obtained vs delta T Span
(LaFeCoSi5)',txt});
i=i+1;
end
```

Harvest_integration_LaFeCoSi11_E_vs_dT_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
% data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=281.546; %WE ARE ASSUMING Tcurie = 282 K
tempIndCurie = find(tempInt == T_curie);
i=0;
n=length(tempInt)-tempIndCurie;
while i<n

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
```

```
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;
    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;

    plot(dT,Energy,'or')
    hold on
    xlabel('dT span (K)');
    ylabel('E (J/m^3)');
    txt=['Center of span = ',num2str(T_curie),' K'];
```

```
    title({'Energy Obtained vs delta T Span
(LaFeCoSi11)',txt});
i=i+1;
end
```

Harvest_integration_LaFeCoSi15_E_vs_dT_V2

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=288.446; %282.419; %WE ARE ASSUMING Tcurie =
282 K -> No, its 288K
tempIndCurie = find(tempInt == T_curie);
i=0;
n=length(tempInt)-tempIndCurie;
while i<n

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
```

```
    fieldIndStart = find(fieldInt == startfield);
    endfield = 792880.989; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu

    plot(dT,Energy,'or')
    hold on
    xlabel('dT span (K)');
    ylabel('E (J/m^3)');
    txt=['Center of span = ',num2str(T_curie),' K'];
    title({'Energy Obtained vs delta T Span
(LaFeCoSi15)',txt});
i=i+1;
```

```
end
```

Harvest_integration_LaCaSrMnO_E_vs_dT_V2

```
%Note that the short curve is due to the short range
of temperatures of the data
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=283.999; %280.999; %WE ARE ASSUMING Tcurie =
284 K
tempIndCurie = find(tempInt == T_curie);
i=0;
n=length(tempInt)-tempIndCurie;
while i<n

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = -649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
```

```
    endfield = 794132.790000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;
    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %


    plot(dT,Energy,'or')
    hold on
    xlabel('dT span (K)');
    ylabel('E (J/m^3)');
    txt=['Center of span = ',num2str(T_curie),' K'];
    title({'Energy Obtained vs delta T Span
(LaCaSrMnO)',txt});
i=i+1;
end
```

Harvest_ALL_E_vs_dT_V2all_together

```
% Gd
##########################################################
###########
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=293;%286;
tempIndCurie = find(tempInt == T_curie);
i=1; %0;
n=length(tempInt)-tempIndCurie;
while i<n+1 %<n, chnaged because i changed from 0 to 1

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
```

```
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    dT_v_gd(i)=dT;
    Energy_v_gd(i)=Energy;
i=i+1;
end
txt=['Gd, Center of span = ',num2str(T_curie),' K'];
plot(dT_v_gd,Energy_v_gd,'o-','DisplayName',txt)
hold on
% LaFeCoSi5
########################################################
######

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=275.914; %281.933; %WE ARE ASSUMING Tcurie =
282 K -> no, is around 276K
```

```
tempIndCurie = find(tempInt == T_curie);
i=1;
n=length(tempInt)-tempIndCurie;
while i<n+1

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 800195.686000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    dT_v_5(i)=dT;
    Energy_v_5(i)=Energy;
i=i+1;
end
txt=['LaFeCoSi5, Center of span = ',num2str(T_curie),'
K'];
plot(dT_v_5,Energy_v_5,'or-','DisplayName',txt)
hold on

% LaFeCoSi11
##########################################################
#####
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
% data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;
```

```
tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=281.546; %WE ARE ASSUMING Tcurie = 282 K
tempIndCurie = find(tempInt == T_curie);
i=1;
n=length(tempInt)-tempIndCurie;
while i<n+1

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    dT_v_11(i)=dT;
    Energy_v_11(i)=Energy;
i=i+1;
end
```

```
txt=['LaFeCoSi11, Center of span =
',num2str(T_curie),' K'];
plot(dT_v_11,Energy_v_11,'om-','DisplayName',txt)
hold on

% LaFeCoSi15
##########################################################
###
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=288.446; %282.419; %WE ARE ASSUMING Tcurie =
282 K -> No, its 288K
tempIndCurie = find(tempInt == T_curie);
i=1;
n=length(tempInt)-tempIndCurie;
while i<n+1

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 792880.989; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
```

```
    H_spacing=fieldInt(2)-fieldInt(1);

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    dT_v_15(i)=dT;
    Energy_v_15(i)=Energy;
i=i+1;
end
txt=['LaFeCoSi15, Center of span =
',num2str(T_curie),' K'];
plot(dT_v_15,Energy_v_15,'o-','DisplayName',txt)
hold on

% LaCaSrMnO
##############################################
%Note that the short curve is due to the short range
of temperatures of the data
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,3);
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

T_curie=283.999; %280.999; %WE ARE ASSUMING Tcurie =
284 K
tempIndCurie = find(tempInt == T_curie);
```

```
i=1;
n=length(tempInt)-tempIndCurie;
while i<n+1

    starttemp = tempInt(tempIndCurie-i); %100;
    tempIndStart = find(tempInt == starttemp);
    endtemp =  tempInt(tempIndCurie+i);
    tempIndEnd = find(tempInt == endtemp);
    dT=endtemp-starttemp;

    startfield = -649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 794132.790000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;

    dT_v_laca(i)=dT;
    Energy_v_laca(i)=Energy;
i=i+1;
end
txt=['LaCaSrMnO, Center of span = ',num2str(T_curie),'
K'];
plot(dT_v_laca,Energy_v_laca,'og-','DisplayName',txt)
hold on
title({'Energy Obtained vs delta T Span for the
different materials','Field span = 0 - 1 T'});
xlabel('dT span (K)');
ylabel('E (J/m^3)');
legend('Location','NorthWest')
```

Harvest_integration_Gd_E_vs_20dT

```
clear all;
close all;
```

```
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

% T_curie=286;
% tempIndCurie = find(tempInt == T_curie);
dT=4:4:20;
cllist=lines(length(dT))
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-20

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));


    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
```

```
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %
UNITS NEED FIXING


%
plot(starttemp(ii),Energy(ii),'o','Color',cllist(ii,:)
)
%     hold on
%     xlabel('Tcold of dT=20 span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span (Gd -
span centered on T curie)');
    starttemp_v(i)=starttemp(ii);
    Energy_v(i)=Energy(ii);
```

```
i=i+1;
end
    txt = ['dT = ',num2str(dT(ii))];

plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
    hold on
    xlabel('Tcold of dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span (Gd)');
   % plot('DisplayName',txt)
end
%legend('dT=5','dT=10','dT=15','dT=20','Location','Nor
thWest')
%legend('dT=5','Location','NorthWest')
%legend('dT=10','Location','NorthWest')
legend('Location','NorthWest')
%legend(['dT=' num2str(dT(1))],['dT='
num2str(dT(2))],['dT=' num2str(dT(3))],['dT='
num2str(dT(4))]);
%legend('dT=5','dT=10','dT=15','dT=20')
% for ii=1:length(dT)
%        legend(['dT=' num2str(dT(ii))]);
%     end
```

Harvest_integration_LaFeCoSi5_E_vs_20dT

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);
```

```
dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

% T_curie=281.933;
% tempIndCurie = find(tempInt == T_curie);
dT=4:4:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT-1)%19 %for i=n-18 or i=n-any_number_<_19
the tempIndStart is so that when summing dT to get
tempIndEnd, tempIndEnd will be > 209, causing eeror as
tempInt has only 209 values (it would look for the 210
value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);


    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 800195.686000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
```

```
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %
UNITS NEED FIXING


%     plot(starttemp,Energy,'or')
%     hold on
%     xlabel('Tcold of dT=20 span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span
(LaFeCoSi5)');
    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT = ',num2str(dT(ii))];

plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
    hold on
    xlabel('Tcold of dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span
(LaFeCosi5)');
```

```
   % plot('DisplayName',txt)
end
legend('Location','NorthWest')
```

Harvest_integration_LaFeCoSi11_E_vs_20dT

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

% T_curie=281.933;
% tempIndCurie = find(tempInt == T_curie);
dT=4:4:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).
```

```
    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);


    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %
UNITS NEED FIXING
```

```
%     plot(starttemp,Energy,'or')
%     hold on
%     xlabel('Tcold of dT=20 span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span
(LaFeCoSi5)');
    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT = ',num2str(dT(ii))];

plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
    hold on
    xlabel('Tcold of dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span
(LaFeCosi11)');
   % plot('DisplayName',txt)
end
legend('Location','NorthWest')
```

Harvest_integration_LaFeCoSi15_E_vs_20dT

```
clear all;
close all;

data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);
```

```
dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

% T_curie=281.933;
% tempIndCurie = find(tempInt == T_curie);
dT=4:4:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);


    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 792880.989; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
   % fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return
%
%     plot(fieldInt,dataInt(:,tempIndStart),'-b')
%     hold on
%     plot(fieldInt,dataInt(:,tempIndEnd),'-r')

    H_spacing=fieldInt(2)-fieldInt(1);
```

```
%
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
%     HighInt_3
=H_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    %In process vectors for plotting:

%LowInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStart
:fieldIndEnd,tempIndStart));

%HighInt_3_cum=H_spacing*cumtrapz(dataInt(fieldIndStar
t:fieldIndEnd,tempIndEnd));

    %Energy_cum = (LowInt_3_cum-HighInt_3_cum)*mu;  %
UNITS NEED FIXING


%     plot(starttemp,Energy,'or')
%     hold on
%     xlabel('Tcold of dT=20 span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span
(LaFeCoSi5)');
    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT = ',num2str(dT(ii))];

plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
    hold on
    xlabel('Tcold of dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span
(LaFeCosi15)');
```

```
   % plot('DisplayName',txt)
end
legend('Location','NorthWest')
```

Harvest_integration_ALL_E_vs_20dT_V2

```
%V2 => field searcher implemented

clear all;
close all;


global_field_1=0.5;  %in teslas
global_field_2=1;

% GD
%#########################################################
#####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%Field finder
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
```

```
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=20:4:20;
cllist=lines(length(dT))
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);%zeros(n-(dT(ii)+1),1);
%zeros(n-21,1);
centertemp_v=starttemp_v;
Energy_v=zeros(n-21,1);%zeros(n-(dT(ii)+1),1);
%zeros(n-21,1);
while i<n-(dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));


    %startfield = 0.01;
    startfield =closest_field1;% 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield =closest_field2;% 799633.017000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);
```

```
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED


    starttemp_v(i)=starttemp(ii);
    centertemp_v(i)=starttemp_v(i)+dT(ii)/2;
    Energy_v(i)=Energy(ii);

i=i+1;
end
    txt = ['dT (Gd) = ',num2str(dT(ii))];
    plot(centertemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
end
legend('Location','NorthWest')

% LaFeCoSi5
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%Field finder
```

```
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=20:5:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-(dT),1); %zeros(n-21,1);
centertemp_v=starttemp_v;
Energy_v=zeros(n-(dT),1); %zeros(n-21,1);
while i<n-(dT-1)%19 %for i=n-18 or i=n-any_number_<_19
the tempIndStart is so that when summing dT to get
tempIndEnd, tempIndEnd will be > 209, causing error as
tempInt has only 209 values (it would look for the 210
value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %800195.686000000; %we
should take 1T=795774.7155 but we are taking
```

```
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    centertemp_v(i)=starttemp_v(i)+dT(ii)/2;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT (LaFeCoSi5) = ',num2str(dT(ii))];
    plot(centertemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
end
legend('Location','NorthWest')

% LaFeCoSi11
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);
```

```
dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%Field finder
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=20:5:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-(dT),1); %zeros(n-21,1);
centertemp_v=starttemp_v;
Energy_v=zeros(n-(dT),1); %zeros(n-21,1);
while i<n-(dT-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);
```

```
    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    centertemp_v(i)=starttemp_v(i)+dT(ii)/2;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT (LaFeCoSi11) = ',num2str(dT(ii))];
    plot(centertemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
end
legend('Location','NorthWest')

% LaFeCoSi15
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);
```

```
[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%Field finder
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=20:5:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-(dT),1); %zeros(n-21,1);
centertemp_v=starttemp_v;
Energy_v=zeros(n-(dT),1); %zeros(n-21,1);
while i<n-(dT-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
```

```
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %792880.989; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    centertemp_v(i)=starttemp_v(i)+dT(ii)/2;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT (LaFeCoSi15) = ',num2str(dT(ii))];
    plot(centertemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
end
legend('Location','NorthWest')

% LaCaSrMnO
%##########################################################
#########################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,2);
```

```
fieldInt=linspace(field(1,1),field(end,1),100);%100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%Field finder
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=20:5:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-(dT),1); %zeros(n-21,1);
centertemp_v=starttemp_v;
Energy_v=zeros(n-(dT),1); %zeros(n-21,1);
while i<n-(dT-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).
```

```
    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %-649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %794132.790000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    centertemp_v(i)=starttemp_v(i)+dT(ii)/2;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT (LaCaSrMnO) = ',num2str(dT(ii))];
    plot(centertemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
    xlabel('Tcenter of dT span (K)');
    ylabel('E (J/m^3)');
    title({'Energy Obtained vs delta T Span','Field
span = 0 - 1 T'});
   % plot('DisplayName',txt)
end
legend('Location','NorthWest')
```

Harvest_integration_ALL_peaks

```
clear all;
close all;
```

```
% GD
%#####################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

dT=5:5:60;
cllist=lines(length(dT))
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<(n-dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));


    %startfield = 0.01;
    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
```

```
    fieldIndEnd = find(fieldInt == endfield);
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED


    starttemp_v(i)=starttemp(ii);
    Energy_v(i)=Energy(ii);

i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));

    txt = ['dT (Gd) = ',num2str(dT(ii))];

plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
    hold on
end
legend('Location','NorthWest')
txt = ['Peak Curve - Gd'];
plot(dT,Max_E_v,'o-','DisplayName',txt);
    xlabel('dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span - Peak
Curves');
    hold on

% LaFeCoSi5
%##################################################
###################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
```

```
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

dT=5:5:60;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19 %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 800195.686000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);
```

```
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
%     start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
%      txt = ['dT (LaFeCoSi5) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi5'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');

% LaFeCoSi11
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);
```

```
[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

dT=5:5:60;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 799633.017; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
```

```
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi11) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
%      [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
%     start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi11'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');


% LaFeCoSi15
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

dT=5:5:60;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
```

```
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = 7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 792880.989; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi15) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
%     [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
%     start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
```

```
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi15'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');

% LaCaSrMnO
%##########################################################
#########################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,2);
fieldInt=linspace(field(1,1),field(end,1),100);%100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

dT=5:5:20;
cllist=lines(length(dT))
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
```

```
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = -649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = 794132.790000000; %we should take
1T=795774.7155 but we are taking 799633.017 which is
the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
 txt = ['dT (LaCaSrMnO) = ',num2str(dT(ii))];
    plot(starttemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
    hold on
    xlabel('Tcold of dT span (k)');
    ylabel('E (J/m^3)');
    title('Energy Obtained vs delta T Span');
   % plot('DisplayName',txt)
   [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
legend('Location','NorthWest')
txt = ['Peak Curve - LaCaSrMnO'];
plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
xlabel('Tcold of dT span (k)');
```

```
ylabel('E (J/m^3)');
title('Energy Obtained vs dT Span');
```

Harvest_integration_ALL_field_varied_peak_curves

```
clear all;
close all;


% global_field_1 = 0.5; %input in Tesla
% global_field_2 = 1; %input in Tesla

% GD
%#########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=1:7
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
```

```
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
```

```
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<(n-dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));

    %startfield = 0.01;
    startfield =closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED

    starttemp_v(i)=starttemp(ii);
    Energy_v(i)=Energy(ii);
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));

%     txt = ['dT (Gd) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
legend('Location','NorthWest')
```

```
 %txt = ['Peak Curve - Gd'];
% % plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
%     txt3 = ['Field Span: ',num2str(global_field_1),'
- ',num2str(global_field_2),' (T)' ];
%     title({'Energy Obtained vs delta T span
(k)',txt3});
%     hold on
% txt = ['Gd'];
% plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  xlabel('dT span (k)');
% ylabel('Tcenter (k)');
% txt2 = ['Field Span: ',num2str(global_field_1),' -
',num2str(global_field_2),' (T)' ];
% title({'Tcenter (T of peak) vs dT',txt2});
% legend('Location','NorthEast')
% hold on
if j==1
    Gd_st_temp_max_E_0_05=start_temp_max_E_v;
    Gd_max_E_0_05=Max_E_v;
    legend('Location','NorthWest')
    txt = ['Peak Curve, 0 - 0.5 T'];
    xlabel('Tcold dT span (k)');
    ylabel('E (J/m^3)');
    %title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 0.5 (T)','Gd'});
    plot(Gd_st_temp_max_E_0_05,Gd_max_E_0_05,'o-
','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','Gd'});
    hold on
end
if j==2
    Gd_st_temp_max_E_05_1=start_temp_max_E_v;
Gd_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0.5 - 1 (T)'});
plot(Gd_st_temp_max_E_05_1,Gd_max_E_05_1,'o-
','DisplayName',txt);

end
if j==3
    Gd_st_temp_max_E_0_1=start_temp_max_E_v;
Gd_max_E_0_1=Max_E_v;
```

```
txt = ['Peak Curve, 0 - 1 T'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 1 (T)'});
plot(Gd_st_temp_max_E_0_1,Gd_max_E_0_1,'o-
','DisplayName',txt);

end
if j==4
    Gd_st_temp_max_E_1_12=start_temp_max_E_v;
Gd_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 1 - 1.2 (T)'});
plot(Gd_st_temp_max_E_1_12,Gd_max_E_1_12,'o-
','DisplayName',txt);

end
if j==5
    Gd_st_temp_max_E_0_12=start_temp_max_E_v;
Gd_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 0.5 (T)'});
plot(Gd_st_temp_max_E_0_12,Gd_max_E_0_12,'o-
','DisplayName',txt);
end
if j==6
    Gd_st_temp_max_E_025_075=start_temp_max_E_v;
Gd_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T'];
plot(Gd_st_temp_max_E_025_075,Gd_max_E_025_075,'o-
','DisplayName',txt);
end
if j==7
    Gd_st_temp_max_E_07_12=start_temp_max_E_v;
Gd_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T'];
plot(Gd_st_temp_max_E_07_12,Gd_max_E_07_12,'o-
','DisplayName',txt);
end
end

% LaFeCoSi5
%#####################################################
####################
```

```
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=1:7
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
```

```
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19 %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);
```

```
    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %800195.686000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
%      txt = ['dT (LaFeCoSi5) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
 legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi5'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
 xlabel('Tcold of dT span (K)');
 ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi5'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  legend('Location','NorthEast')
%  hold on

 if j==1
    L5_st_temp_max_E_0_05=start_temp_max_E_v;
```

```
    L5_max_E_0_05=Max_E_v;
    legend('Location','NorthWest')
    txt = ['Peak Curve, 0 - 0.5 T'];
    xlabel('Tcold dT span (k)');
    ylabel('E (J/m^3)');
    plot(L5_st_temp_max_E_0_05,L5_max_E_0_05,'o-
','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi5'});
    hold on
end
if j==2
    L5_st_temp_max_E_05_1=start_temp_max_E_v;
L5_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T'];
plot(L5_st_temp_max_E_05_1,L5_max_E_05_1,'o-
','DisplayName',txt);

end
if j==3
    L5_st_temp_max_E_0_1=start_temp_max_E_v;
L5_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T'];
plot(L5_st_temp_max_E_0_1,L5_max_E_0_1,'o-
','DisplayName',txt);

end
if j==4
    L5_st_temp_max_E_1_12=start_temp_max_E_v;
L5_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T'];
plot(L5_st_temp_max_E_1_12,L5_max_E_1_12,'o-
','DisplayName',txt);

end
if j==5
    L5_st_temp_max_E_0_12=start_temp_max_E_v;
L5_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T'];
plot(L5_st_temp_max_E_0_12,L5_max_E_0_12,'o-
','DisplayName',txt);
end
if j==6
    L5_st_temp_max_E_025_075=start_temp_max_E_v;
L5_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T'];
```

```
plot(L5_st_temp_max_E_025_075,L5_max_E_025_075,'o-
','DisplayName',txt);
end
if j==7
    L5_st_temp_max_E_07_12=start_temp_max_E_v;
L5_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T'];
plot(L5_st_temp_max_E_07_12,L5_max_E_07_12,'o-
','DisplayName',txt);
end
end

% LaFeCoSi11
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=1:7
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
```

```
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
```

```
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi11) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
     [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
```

```
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi11'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (K)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi11'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
% hold on
if j==1
    L11_st_temp_max_E_0_05=start_temp_max_E_v;
    L11_max_E_0_05=Max_E_v;
    txt = ['Peak Curve, 0 - 0.5 T'];
    plot(L11_st_temp_max_E_0_05,L11_max_E_0_05,'o-
','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi11'});
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2
    L11_st_temp_max_E_05_1=start_temp_max_E_v;
L11_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T'];
plot(L11_st_temp_max_E_05_1,L11_max_E_05_1,'o-
','DisplayName',txt);

end
if j==3
    L11_st_temp_max_E_0_1=start_temp_max_E_v;
L11_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T'];
plot(L11_st_temp_max_E_0_1,L11_max_E_0_1,'o-
','DisplayName',txt);

end
if j==4
    L11_st_temp_max_E_1_12=start_temp_max_E_v;
L11_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T'];
plot(L11_st_temp_max_E_1_12,L11_max_E_1_12,'o-
','DisplayName',txt);

end
```

```
if j==5
    L11_st_temp_max_E_0_12=start_temp_max_E_v;
L11_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T'];
plot(L11_st_temp_max_E_0_12,L11_max_E_0_12,'o-
','DisplayName',txt);
end
if j==6
    L11_st_temp_max_E_025_075=start_temp_max_E_v;
L11_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T'];
plot(L11_st_temp_max_E_025_075,L11_max_E_025_075,'o-
','DisplayName',txt);
end
if j==7
    L11_st_temp_max_E_07_12=start_temp_max_E_v;
L11_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T'];
plot(L11_st_temp_max_E_07_12,L11_max_E_07_12,'o-
','DisplayName',txt);
end
end
% LaFeCoSi15
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=1:7
%SEARCH OF DESIRED FIELDS:
```

```
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
```

```
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %792880.989; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
```

```
i=i+1;
end
%     txt = ['dT (LaFeCoSi15) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi15'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi15'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  legend('Location','NorthEast')
% hold on
if j==1
    L15_st_temp_max_E_0_05=start_temp_max_E_v;
    L15_max_E_0_05=Max_E_v;
    txt = ['Peak Curve, 0 - 0.5 T'];
    plot(L15_st_temp_max_E_0_05,L15_max_E_0_05,'o-
','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi15'});
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2
    L15_st_temp_max_E_05_1=start_temp_max_E_v;
L15_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T'];
plot(L15_st_temp_max_E_05_1,L15_max_E_05_1,'o-
','DisplayName',txt);

end
if j==3
    L15_st_temp_max_E_0_1=start_temp_max_E_v;
L15_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T'];
```

```
plot(L15_st_temp_max_E_0_1,L15_max_E_0_1,'o-
','DisplayName',txt);

end
if j==4
    L15_st_temp_max_E_1_12=start_temp_max_E_v;
L15_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T'];
plot(L15_st_temp_max_E_1_12,L15_max_E_1_12,'o-
','DisplayName',txt);

end
if j==5
    L15_st_temp_max_E_0_12=start_temp_max_E_v;
L15_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T'];
plot(L15_st_temp_max_E_0_12,L15_max_E_0_12,'o-
','DisplayName',txt);
end
if j==6
    L15_st_temp_max_E_025_075=start_temp_max_E_v;
L15_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T'];
plot(L15_st_temp_max_E_025_075,L15_max_E_025_075,'o-
','DisplayName',txt);
end
if j==7
    L15_st_temp_max_E_07_12=start_temp_max_E_v;
L15_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T'];
plot(L15_st_temp_max_E_07_12,L15_max_E_07_12,'o-
','DisplayName',txt);
end
end
% LaCaSrMnO
%#####################################################
#########################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
```

```
tempInt=round(tempInt,2);
fieldInt=linspace(field(1,1),field(end,1),100);%100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=1:7
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
```

```
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:20;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %-649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %794132.790000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
```

```
    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%  txt = ['dT (LaCaSrMnO) = ',num2str(dT(ii))];
%     plot(starttemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
%     hold on
%     xlabel('Tcold of dT span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span');
%    % plot('DisplayName',txt)
   [Max_E_LaCaSrMnO_v(ii),Idx_Max_E_v(ii)] =
max(Energy_v);
    start_temp_max_E_LaCaSrMnO_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaCaSrMnO'];
% plot(start_temp_max_E_LaCaSrMnO_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaCaSrMnO'];
%  plot(dT,start_temp_max_E_LaCaSrMnO_v,'o-
','DisplayName',txt);
%  legend('Location','NorthEast')
 if j==1

LaCa_st_temp_max_E_0_05=start_temp_max_E_LaCaSrMnO_v;
    LaCa_max_E_0_05=Max_E_LaCaSrMnO_v;
    txt = ['Peak Curve, 0 - 0.5 T'];
    plot(LaCa_st_temp_max_E_0_05,LaCa_max_E_0_05,'o-
','DisplayName',txt);
```

```
    title({'Peak curves of Evergy vs dT functions for
several field spans','All materials'});
     xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2

LaCa_st_temp_max_E_05_1=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_05_1=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0.5 - 1 T'];
plot(LaCa_st_temp_max_E_05_1,LaCa_max_E_05_1,'o-
','DisplayName',txt);

end
if j==3

LaCa_st_temp_max_E_0_1=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_0_1=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0 - 1 T'];
plot(LaCa_st_temp_max_E_0_1,LaCa_max_E_0_1,'o-
','DisplayName',txt);

end
if j==4

LaCa_st_temp_max_E_1_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_1_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 1 - 1.2 T'];
plot(LaCa_st_temp_max_E_1_12,LaCa_max_E_1_12,'o-
','DisplayName',txt);

end
if j==5

LaCa_st_temp_max_E_0_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_0_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0 - 1.2 T'];
plot(LaCa_st_temp_max_E_0_12,LaCa_max_E_0_12,'o-
','DisplayName',txt);
end
if j==6

LaCa_st_temp_max_E_025_075=start_temp_max_E_LaCaSrMnO_
v;
LaCa_max_E_025_075=Max_E_LaCaSrMnO_v;
```

```
txt = ['Peak Curve, 0.25 - 0.75 T'];
plot(LaCa_st_temp_max_E_025_075,LaCa_max_E_025_075,'o-
','DisplayName',txt);
end
if j==7

LaCa_st_temp_max_E_07_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_07_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0.7 - 1.2 T'];
plot(LaCa_st_temp_max_E_07_12,LaCa_max_E_07_12,'o-
','DisplayName',txt);
end
end
```

Harvest_integration_ALL_Tcenter_vd_Tspan_all_fields

```
clear all;
close all;


global_field_1 = 0; %input in Tesla
global_field_2 = 1.2; %input in Tesla

% GD
%#####################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%SEARCH OF DESIRED FIELDS:
```

```
%starting filed
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<(n-dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));

    %startfield = 0.01;
    startfield =closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
```

```
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED


    starttemp_v(i)=starttemp(ii);
    Energy_v(i)=Energy(ii);
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));

%     txt = ['dT (Gd) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
legend('Location','NorthWest')
% txt = ['Peak Curve - Gd'];
% plot(dT,Max_E_v,'o-','DisplayName',txt);
%     xlabel('dT span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span - Peak
Curves');
%     hold on
 txt = ['Gd'];
 plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
 xlabel('dT span (k)');
ylabel('Tcenter (k)');
txt2 = ['Field Span: ',num2str(global_field_1),' -
',num2str(global_field_2),' (T)' ];
title({'Tcenter (T of peak) vs dT',txt2});
legend('Location','NorthEast')
hold on
```

```
% LaFeCoSi5
%#########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
```

```
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19 %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %800195.686000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
%       txt = ['dT (LaFeCoSi5) = ',num2str(dT(ii))];
```

```
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi5'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
txt = ['LaFeCoSi5'];
 plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
 legend('Location','NorthEast')
 hold on

% LaFeCoSi11
%#########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
```

```
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);
```

```
LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi11) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
     [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi11'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
txt = ['LaFeCoSi11'];
 plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
hold on

% LaFeCoSi15
%##################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
```

```
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).
```

```
    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %792880.989; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi15) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi15'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
txt = ['LaFeCoSi15'];
 plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
 legend('Location','NorthEast')
```

```
hold on

% LaCaSrMnO
%#########################################################
#########################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,2);
fieldInt=linspace(field(1,1),field(end,1),100);%100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector
```

```
dT=5:5:20;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %-649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %794132.790000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%  txt = ['dT (LaCaSrMnO) = ',num2str(dT(ii))];
%     plot(starttemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
```

```
%     hold on
%     xlabel('Tcold of dT span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span');
%    % plot('DisplayName',txt)
   [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_LaCaSrMnO_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaCaSrMnO'];
% plot(start_temp_max_E_LaCaSrMnO_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
txt = ['LaCaSrMnO'];
 plot(dT,start_temp_max_E_LaCaSrMnO_v,'o-
','DisplayName',txt);
 legend('Location','NorthEast')
```

Harvest_integration_ALL_fields_E_vs_dT

```
clear all;
close all;


% global_field_1 = 0.5; %input in Tesla
% global_field_2 = 1; %input in Tesla

% GD
%#####################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));

fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);
```

```
[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=5:5
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
field1_tesla=global_field_1;;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
```

```
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<(n-dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));

    %startfield = 0.01;
    startfield =closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);
    fieldIndEnd = fieldIndEnd-1; %This is to avoid
taking NAN numbers in trapz function later, which then
gives NAN in return

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
```

```
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy(ii) = (LowInt_3-HighInt_3)*mu;  % UNITS
NEED FIXING -- FIXED

    starttemp_v(i)=starttemp(ii);
    Energy_v(i)=Energy(ii);
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));

%     txt = ['dT (Gd) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
legend('Location','NorthWest')
 %txt = ['Peak Curve - Gd'];
% % plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
%     txt3 = ['Field Span: ',num2str(global_field_1),'
- ',num2str(global_field_2),' (T)' ];
%     title({'Energy Obtained vs delta T span
(k)',txt3});
%     hold on
% txt = ['Gd'];
% plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  xlabel('dT span (k)');
% ylabel('Tcenter (k)');
% txt2 = ['Field Span: ',num2str(global_field_1),' -
',num2str(global_field_2),' (T)' ];
% title({'Tcenter (T of peak) vs dT',txt2});
% legend('Location','NorthEast')
hold on
if j==1
    Gd_st_temp_max_E_0_05=start_temp_max_E_v;
    Gd_max_E_0_05=Max_E_v;
    legend('Location','NorthWest')
    txt = ['Peak Curve, 0 - 0.5 T (Gd)'];
    xlabel('Tcold dT span (K)');
```

```
    ylabel('E (J/m^3)');
    %title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 0.5 (T)','Gd'});
    plot(dT,Gd_max_E_0_05,'o-','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','Gd'});
    hold on
end
if j==2
    Gd_st_temp_max_E_05_1=start_temp_max_E_v;
Gd_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T (Gd)'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0.5 - 1 (T)'});
plot(dT,Gd_max_E_05_1,'o-','DisplayName',txt);

end
if j==3
    Gd_st_temp_max_E_0_1=start_temp_max_E_v;
Gd_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T (Gd)'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 1 (T)'});
plot(dT,Gd_max_E_0_1,'o-','DisplayName',txt);

end
if j==4
    Gd_st_temp_max_E_1_12=start_temp_max_E_v;
Gd_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T (Gd)'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 1 - 1.2 (T)'});
plot(dT,Gd_max_E_1_12,'o-','DisplayName',txt);

end
if j==5
    Gd_st_temp_max_E_0_12=start_temp_max_E_v;
Gd_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T (Gd)'];
% title({'Peak curves of Evergy vs dT
functions','Field Span: 0 - 0.5 (T)'});
plot(dT,Gd_max_E_0_12,'o-','DisplayName',txt);
end
if j==6
    Gd_st_temp_max_E_025_075=start_temp_max_E_v;
Gd_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T (Gd)'];
```

```
plot(dT,Gd_max_E_025_075,'o-','DisplayName',txt);
end
if j==7
    Gd_st_temp_max_E_07_12=start_temp_max_E_v;
Gd_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T (Gd)'];
plot(dT,Gd_max_E_07_12,'o-','DisplayName',txt);
end
end

% LaFeCoSi5
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i5_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=5:5
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
```

```
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
```

```
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19 %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %800195.686000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
%      txt = ['dT (LaFeCoSi5) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
end
 legend('Location','NorthWest')
```

```
% txt = ['Peak Curve - LaFeCoSi5'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
 xlabel('Tcold of dT span (K)');
 ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi5'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  legend('Location','NorthEast')
%  hold on

 if j==1
    L5_st_temp_max_E_0_05=start_temp_max_E_v;
    L5_max_E_0_05=Max_E_v;
    legend('Location','NorthWest')
    txt = ['Peak Curve, 0 - 0.5 T (LaFeCoSi5)'];
    xlabel('Tcold dT span (k)');
    ylabel('E (J/m^3)');
    plot(dT,L5_max_E_0_05,'o-','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi5'});
    hold on
end
if j==2
    L5_st_temp_max_E_05_1=start_temp_max_E_v;
L5_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T (LaFeCoSi5)'];
plot(dT,L5_max_E_05_1,'o-','DisplayName',txt);

end
if j==3
    L5_st_temp_max_E_0_1=start_temp_max_E_v;
L5_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T (LaFeCoSi5)'];
plot(dT,L5_max_E_0_1,'o-','DisplayName',txt);

end
if j==4
    L5_st_temp_max_E_1_12=start_temp_max_E_v;
L5_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T (LaFeCoSi5)'];
plot(dT,L5_max_E_1_12,'o-','DisplayName',txt);

end
if j==5
    L5_st_temp_max_E_0_12=start_temp_max_E_v;
L5_max_E_0_12=Max_E_v;
```

```
txt = ['Peak Curve, 0 - 1.2 T (LaFeCoSi5)'];
plot(dT,L5_max_E_0_12,'o-','DisplayName',txt);
end
if j==6
    L5_st_temp_max_E_025_075=start_temp_max_E_v;
L5_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T (LaFeCoSi5)'];
plot(dT,L5_max_E_025_075,'o-','DisplayName',txt);
end
if j==7
    L5_st_temp_max_E_07_12=start_temp_max_E_v;
L5_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T (LaFeCoSi5)'];
plot(dT,L5_max_E_07_12,'o-','DisplayName',txt);
end
end

% LaFeCoSi11
%##########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i11_magnetization_table.txt');
%data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=5:5
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
```

```
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector
```

```
dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %799633.017; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi11) = ',num2str(dT(ii))];
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
```

```
%     hold on
      [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi11'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (K)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi11'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
% hold on
if j==1
    L11_st_temp_max_E_0_05=start_temp_max_E_v;
    L11_max_E_0_05=Max_E_v;
    txt = ['Peak Curve, 0 - 0.5 T (LaFeCoSi11)'];
    plot(dT,L11_max_E_0_05,'o-','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi11'});
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2
    L11_st_temp_max_E_05_1=start_temp_max_E_v;
L11_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T (LaFeCoSi11)'];
plot(dT,L11_max_E_05_1,'o-','DisplayName',txt);

end
if j==3
    L11_st_temp_max_E_0_1=start_temp_max_E_v;
L11_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T (LaFeCoSi11)'];
plot(dT,L11_max_E_0_1,'o-','DisplayName',txt);

end
if j==4
    L11_st_temp_max_E_1_12=start_temp_max_E_v;
L11_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T (LaFeCoSi11)'];
plot(dT,L11_max_E_1_12,'o-','DisplayName',txt);
```

```
end
if j==5
    L11_st_temp_max_E_0_12=start_temp_max_E_v;
L11_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T (LaFeCoSi11)'];
plot(dT,L11_max_E_0_12,'o-','DisplayName',txt);
end
if j==6
    L11_st_temp_max_E_025_075=start_temp_max_E_v;
L11_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T (LaFeCoSi11)'];
plot(dT,L11_max_E_025_075,'o-','DisplayName',txt);
end
if j==7
    L11_st_temp_max_E_07_12=start_temp_max_E_v;
L11_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T (LaFeCoSi11)'];
plot(dT,L11_max_E_07_12,'o-','DisplayName',txt);
end
end
% LaFeCoSi15
%#########################################################
####################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LaFeCoS
i15_magnetization_table.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;

for j=5:5
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
```

```
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
```

```
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:60;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %7957.74700000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %792880.989; %we should
take 1T=795774.7155 but we are taking 799633.017 which
is the closest value on new table fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED

    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%     txt = ['dT (LaFeCoSi15) = ',num2str(dT(ii))];
```

```
%
plot(starttemp_v,Energy_v,'o','DisplayName',txt);%,'Co
lor',cllist(ii,:))
%     hold on
    [Max_E_v(ii),Idx_Max_E_v(ii)] = max(Energy_v);
    start_temp_max_E_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaFeCoSi15'];
% plot(start_temp_max_E_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaFeCoSi15'];
%  plot(dT,start_temp_max_E_v,'o-','DisplayName',txt);
%  legend('Location','NorthEast')
% hold on
if j==1
    L15_st_temp_max_E_0_05=start_temp_max_E_v;
    L15_max_E_0_05=Max_E_v;
    txt = ['Peak Curve, 0 - 0.5 T (LaFeCoSi15)'];
    plot(dT,L15_max_E_0_05,'o-','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','LaFeCoSi15'});
    xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2
    L15_st_temp_max_E_05_1=start_temp_max_E_v;
L15_max_E_05_1=Max_E_v;
txt = ['Peak Curve, 0.5 - 1 T (LaFeCoSi15)'];
plot(dT,L15_max_E_05_1,'o-','DisplayName',txt);

end
if j==3
    L15_st_temp_max_E_0_1=start_temp_max_E_v;
L15_max_E_0_1=Max_E_v;
txt = ['Peak Curve, 0 - 1 T (LaFeCoSi15)'];
plot(dT,L15_max_E_0_1,'o-','DisplayName',txt);

end
if j==4
    L15_st_temp_max_E_1_12=start_temp_max_E_v;
```

```
L15_max_E_1_12=Max_E_v;
txt = ['Peak Curve, 1 - 1.2 T (LaFeCoSi15)'];
plot(dT,L15_max_E_1_12,'o-','DisplayName',txt);

end
if j==5
    L15_st_temp_max_E_0_12=start_temp_max_E_v;
L15_max_E_0_12=Max_E_v;
txt = ['Peak Curve, 0 - 1.2 T (LaFeCoSi15)'];
plot(dT,L15_max_E_0_12,'o-','DisplayName',txt);
end
if j==6
    L15_st_temp_max_E_025_075=start_temp_max_E_v;
L15_max_E_025_075=Max_E_v;
txt = ['Peak Curve, 0.25 - 0.75 T (LaFeCoSi15)'];
plot(dT,L15_max_E_025_075,'o-','DisplayName',txt);
end
if j==7
    L15_st_temp_max_E_07_12=start_temp_max_E_v;
L15_max_E_07_12=Max_E_v;
txt = ['Peak Curve, 0.7 - 1.2 T (LaFeCoSi15)'];
plot(dT,L15_max_E_07_12,'o-','DisplayName',txt);
end
end
% LaCaSrMnO
%#####################################################
#########################
data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\LCSM2ne
w.txt');
data(:,1)=data(:,1)*795774.7155;
temp=data(1,2:end);
field=data(2:end,1);
mu=4*pi*10^-7;

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
tempInt=round(tempInt,2);
fieldInt=linspace(field(1,1),field(end,1),100);%100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
dataInt(1,:)=zeros;
```

```
for j=5:5
%SEARCH OF DESIRED FIELDS:
%starting filed
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%SEARCH OF DESIRED FIELDS:
%starting filed
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1) %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
```

```
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)) %looks for the closest value of field in
A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2) %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:20;
cllist=lines(length(dT));
n=length(tempInt);
for ii=1:length(dT)
tempIndStart=0;
tempIndEnd=tempIndStart+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-21,1);
Energy_v=zeros(n-21,1);
while i<n-(dT(ii)-1)%19=(dT-1) %for i=n-18 or i=n-
any_number_<_19 the tempIndStart is so that when
summing dT to get tempIndEnd, tempIndEnd will be >
209, causing eeror as tempInt has only 209 values (it
would look for the 210 value, that doesn't exist).

    tempIndStart=tempIndStart+1;
    tempIndEnd=tempIndEnd+1;
    starttemp = tempInt(tempIndStart);
    endtemp =   tempInt(tempIndEnd);

    startfield = closest_field1; %-649.880000000000;
    fieldIndStart = find(fieldInt == startfield);
    endfield = closest_field2; %794132.790000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
    fieldIndEnd = find(fieldInt == endfield);

    H_spacing=fieldInt(2)-fieldInt(1);


LowInt_3=H_spacing*trapz(dataInt(fieldIndStart:fieldIn
dEnd,tempIndStart));
    HighInt_3
=H_spacing*trapz(dataInt(fieldIndStart:fieldIndEnd,tem
pIndEnd));

    Energy = (LowInt_3-HighInt_3)*mu;  % UNITS NEED
FIXING -- FIXED
```

```
    starttemp_v(i)=starttemp;
    Energy_v(i)=Energy;
i=i+1;
end
%  txt = ['dT (LaCaSrMnO) = ',num2str(dT(ii))];
%     plot(starttemp_v,Energy_v,'o-
','DisplayName',txt);%,'Color',cllist(ii,:))
%     hold on
%     xlabel('Tcold of dT span (k)');
%     ylabel('E (J/m^3)');
%     title('Energy Obtained vs delta T Span');
%    % plot('DisplayName',txt)
   [Max_E_LaCaSrMnO_v(ii),Idx_Max_E_v(ii)] =
max(Energy_v);
    start_temp_max_E_LaCaSrMnO_v(ii) =
starttemp_v(Idx_Max_E_v(ii));
end
% legend('Location','NorthWest')
% txt = ['Peak Curve - LaCaSrMnO'];
% plot(start_temp_max_E_LaCaSrMnO_v,Max_E_v,'o-
','DisplayName',txt);
% xlabel('Tcold of dT span (k)');
% ylabel('E (J/m^3)');
% title('Energy Obtained vs dT Span');
% txt = ['LaCaSrMnO'];
%  plot(dT,start_temp_max_E_LaCaSrMnO_v,'o-
','DisplayName',txt);
%  legend('Location','NorthEast')
 if j==1

LaCa_st_temp_max_E_0_05=start_temp_max_E_LaCaSrMnO_v;
    LaCa_max_E_0_05=Max_E_LaCaSrMnO_v;
    txt = ['Peak Curve, 0 - 0.5 T (LaCaSrMnO)'];
    plot(dT,LaCa_max_E_0_05,'o-','DisplayName',txt);
    title({'Peak curves of Evergy vs dT functions for
several field spans','All materials'});
     xlabel('Tcold dT span (K)');
    ylabel('E (J/m^3)');
    legend('Location','NorthWest')
    hold on
end
if j==2

LaCa_st_temp_max_E_05_1=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_05_1=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0.5 - 1 T (LaCaSrMnO)'];
```

```
plot(dT,LaCa_max_E_05_1,'o-','DisplayName',txt);

end
if j==3

LaCa_st_temp_max_E_0_1=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_0_1=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0 - 1 T (LaCaSrMnO)'];
plot(dT,LaCa_max_E_0_1,'o-','DisplayName',txt);

end
if j==4

LaCa_st_temp_max_E_1_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_1_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 1 - 1.2 T (LaCaSrMnO)'];
plot(dT,LaCa_max_E_1_12,'o-','DisplayName',txt);

end
if j==5

LaCa_st_temp_max_E_0_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_0_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0 - 1.2 T (LaCaSrMnO)'];
plot(dT,LaCa_max_E_0_12,'o-','DisplayName',txt);
end
if j==6

LaCa_st_temp_max_E_025_075=start_temp_max_E_LaCaSrMnO_
v;
LaCa_max_E_025_075=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0.25 - 0.75 T (LaCaSrMnO)'];
plot(dT,LaCa_max_E_025_075,'o-','DisplayName',txt);
end
if j==7

LaCa_st_temp_max_E_07_12=start_temp_max_E_LaCaSrMnO_v;
LaCa_max_E_07_12=Max_E_LaCaSrMnO_v;
txt = ['Peak Curve, 0.7 - 1.2 T (LaCaSrMnO)'];
plot(dT,LaCa_max_E_07_12,'o-','DisplayName',txt);
end
end
 xlabel('dT span (K)');
 ylabel('Emax (J/m^3)');
 title({'Maximum evergy vs dT, Field span: 0 - 1.2
T','All materials'});
```

Harvest_Cp_mean_method_ALL

```
%PERFROMANCE = 4.2*10^5 / (405*10^5) =1.037%  NO,
TRAPZ() HAD INVERTED
%INDEXES
%NEW PERFORMANCE = 4.2*10^5 / (4.7196*10^7) = 0.89%

clear all;
close all;


global_field_1=0;
global_field_2=1;


data=load('D:\luis-
\Documents\Documents\DINAMARCA\DTU\TFG\scripts\Gd_comm
_cp_table.txt');
data(:,1)=data(:,1)*795774.7155; %covert field column
from tesla to A/m
temp=data(1,2:end); %first row are temperatures
field=data(2:end,1); %first column are fields
mu=4*pi*10^-7;
rho=7900; %Gd density, kg/m^3

tempInt=linspace(temp(1),temp(end),(temp(end)-
temp(1)+1));
fieldInt=linspace(field(1,1),field(end,1),100);
fieldInt=round(fieldInt,3);

[meshtemp,meshfield]=meshgrid(tempInt,fieldInt);

dataInt =
interp2(temp,field,data(2:end,2:end),meshtemp,meshfiel
d);
%dataInt(1,:)=zeros;
tempInt=tempInt+273; %Celcius to Kelvin

for j=1:7
%SEARCH OF DESIRED FIELDS:
%starting field
if j==1
    global_field_1=0;
    global_field_2=0.5;
end
if j==2
    global_field_1=0.5;
    global_field_2=1;
```

```
end
if j==3
    global_field_1=0;
    global_field_2=1;
end
if j==4
    global_field_1=1;
    global_field_2=1.2;
end
if j==5
    global_field_1=0;
    global_field_2=1.2;
end
%to see 0.5T behaviour:
if j==6
    global_field_1=0.25;
    global_field_2=0.75;
end
if j==7
    global_field_1=0.7;
    global_field_2=1.2;
end
%setting desired fields:
field1_tesla=global_field_1;  %Desired field input in
Tesla
field1_find=field1_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest1,index_closest_field1] = min(abs(fieldInt-
field1_find)); %looks for the closest value of field
in A/m in the coverted data fieldInt.
closest_field1=fieldInt(index_closest_field1); %Output
we are looking for in A/m belonging to filedInt vector
%end field
field2_tesla=global_field_2;  %Desired field input in
Tesla
field2_find=field2_tesla*795774.7155;  %conversion
forn Teslas to A/m
[rest2,index_closest_field2] = min(abs(fieldInt-
field2_find)); %looks for the closest value of field
in A/m in the coverted data fieldInt.
closest_field2=fieldInt(index_closest_field2); %Output
we are looking for in A/m belonging to filedInt vector

dT=5:5:5;
cllist=lines(length(dT));
for ii=1:length(dT)
starttemp(ii)=tempInt(1);
```

```
endtemp(ii)=starttemp(ii)+dT(ii);
i=1;
n=length(tempInt);
starttemp_v=zeros(n-dT(ii)-1,1); %zeros(n-21,1);
Energy_v=zeros(n-dT(ii)-1,1); %zeros(n-21,1);
endtemp_v=zeros(n-dT(ii)-1,1);
while i<(n-dT(ii))

    starttemp(ii) = starttemp(ii)+1;
    tempIndStart = find(tempInt == starttemp(ii));
    endtemp(ii) =   endtemp(ii)+1;
    tempIndEnd = find(tempInt == endtemp(ii));

startfield = closest_field1; %7957.74700000000;
fieldIndStart = find(fieldInt == startfield);
endfield = closest_field2; %799633.017000000; %we
should take 1T=795774.7155 but we are taking
799633.017 which is the closest value on new table
fieldInt.
fieldIndEnd = find(fieldInt == endfield);
%fieldIndEnd = fieldIndEnd-1; %This is to avoid taking
NAN numbers in trapz function later, which then gives
NAN in return


T_spacing=tempInt(2)-tempInt(1);
LowInt_3=T_spacing*trapz(dataInt(fieldIndStart,tempInd
Start:tempIndEnd));
HighInt_3
=T_spacing*trapz(dataInt(fieldIndEnd,tempIndStart:temp
IndEnd));

C_mean_low_field=LowInt_3/dT(ii);
C_mean_high_field=HighInt_3/dT(ii);
Cmean(i)=(C_mean_low_field+C_mean_high_field)/2; %we
assume linear distribution
Energy = rho*Cmean(i)*dT(ii);

Energy_v(i) = Energy;
starttemp_v(i)=starttemp(ii);
endtemp_v(i)=endtemp(ii);


i=i+1;
end
figure(ii)
txt1 = ['Field Span: ',num2str(field1_tesla),' -
',num2str(field2_tesla),' (T)' ];
```

```
txt2 = ['dT = ',num2str(dT(ii)),' (K)' ];
txt5 = ['Field span: ',num2str(field1_tesla),' -
',num2str(field2_tesla),' (T)' ];
plot(endtemp_v,Energy_v,'o-','DisplayName',txt5)
xlabel('Twater (K)');
ylabel('Q (J/m^3)');
title({'Heat input vs Thot=Twater (Gd)',txt1});
hold on
legend('Location','NorthWest')
end
% hold off
% figure(j)
% txt3 = ['Field: ',num2str(field1_tesla),' T'];
% txt4 = ['Field: ',num2str(field2_tesla),' T'];
% plot(tempInt,dataInt(fieldIndStart,:),'-
b','DisplayName',txt3)
% hold on
% plot(tempInt,dataInt(fieldIndEnd,:),'-
r','DisplayName',txt4)
% xlabel('T (K)');
% ylabel('Cp (J/kgK)');
% title('Specific Heat Capacity (Gd)');
% legend('Location','NorthWest')
end
```

REFERENCES

[1] R. A. Kishore and S. Priya, "A Review on Low-Grade Thermal Energy Harvesting: Materials, Methods and Devices," *Materials,* pp. 22-27, 2018.

[2] N. Tesla, "LAB-TESLA," [Online]. Available: https://www.lab-tesla.org/396121.html.

[3] R. A. Kishore and S. Priya, "A review on design and performance of thermomagnetic devices," *Renewable and Sustainable Energy Reviews,* 2018.

[4] A. Smith, C. Bahl, R. Bjørk, K. Engelbrecht, K. Nielsen and N. Pryds, "Materials Challenges for High Performance Magnetocaloric Refrigeration Devices," *Advanced Energy Materials,* 2012.

[5] C. R. H. Bahl and K. K. Nielsen, "The effect of demagnetization on the magnetocaloric properties of gadolinium," *JOURNAL OF APPLIED PHYSICS,* 2009.

[6] A. R. Dinesen, "Magnetocaloric and magnetoresistiveproperties of La0.67Ca0.33-xSrxMnO3," Risø National Laboratory, 2004.

[7] R. Bjørk and C. R. Bahl, "Magnetocaloric properties of LaFe13-x-yCoxSiy and commercial grade Gd," *Journal of Magnetism and Magnetic Materials,* 2010.

[8] T. Edison, "Thomas Edison Muckers: Your Blog for Everything Edison, Everyday," [Online]. Available: http://www.edisonmuckers.org/thomas-edison-patents/edison-patents-image-gallery/.

[9] "https://slideplayer.com/slide/4889271/," [Online].