Teaching Basics of Inductive Power Components by Interactive Java Applets for FEM-Based On-Line Calculation of the Magnetic Flux Distribution

Uwe DROFENIK and Johann W. KOLAR

Power Electronic Systems Laboratory
Swiss Federal Institute of Technology Zurich
ETH-Zentrum / ETL, CH-8092 Zurich, Switzerland
Phone: +41-1-632-4267, Fax: +41-1-632-1212
E-mail: **drofenik@lem.ee.ethz.ch**, **kolar@lem.ee.ethz.ch**

Key words

Education, Education tools, Magnetic Devices, Passive Components, Software, Teaching

Abstract

Inductive components like inductors and transformers are basic elements in power electronic circuits. Therefore, understanding the electromagnetic behavior and the dimensioning of theses components is essential. However, after attending introductory courses in basic electrical engineering many students still lack a deeper understanding of the physics of inductors and transformers. This paper describes educational software (*iPES* - Java applets freely available on the Internet) which should help the students to get a better insight into the physics of passive power electronics components. *iPES* does show the distribution of the electromagnetic fields of inductors, transformers where the geometric properties of the magnetic cores and the permeability of the magnetic material could be changed interactively.

1. Introduction

1.1 Teaching Passive Components in Power Electronics

Generally, passive components in power electronics like inductors, transformers or capacitors are often taught in power electronics courses based on equivalent circuits and idealized mathematical descriptions. One important reason for this is that it is difficult to visualize the distribution, the time behavior and the interaction of electromagnetic fields, especially in the presence of media described by permeability or permittivity. Equivalent circuits of inductors and transformers could give the students a quick introduction but fail to show the physical background of non-ideal behavior like stray fields and capacitive coupling of windings which is important for determining the behavior of the components in a power electronic circuit.

In order to support the teacher by explaining passive components being essential for power electronic systems like inductors, transformers, capacitors and heat sinks based on field theory, the authors have developed the software module *iPES-Fields* as part of the Interactive Power Electronics Seminar [1], [2].

1.2 Interactive Power Electronics Seminar (*iPES*)

The Interactive Power Electronics Seminar (*iPES*) is software developed at the Swiss Federal Institute of Technology (ETH Zurich) for supporting the teaching of a basic course on power electronic circuits and systems. *iPES* consists of HTML text with embedded programs (so-called Java applets) for interactive animation and simulation. The user-interface is very easy to use and mainly self-explaining. The software requires just a standard web-browser, i.e. no installations are necessary. *iPES* is freely available at www.ipes.ethz.ch, can be accessed online via Internet or downloaded and installed on the local hard-disc.

iPES went online in July 2001, the free download is available since January 2002. At the time of this writing (May 2002) there have been about 2,000 downloads and more than 12,000 visitors (3,000 visitors during May) on the *iPES* website.

The seminar ("*iPES-Circuits*") consists of a basic course on power electronics which is currently available in English, German, Japanese, Korean and Chinese. A second module, called "*iPES+*" represents a collection of more complex modern power electronic systems still under research or already used in industry with emphasis on three-phase PWM converter systems and three-phase matrix converters.

A third module, called “*iPES-Fields*” is dedicated to teaching the basics and the dimensioning of passive components in power electronics with emphasis on the underlying physics and electromagnetic fields. *iPES-Fields* will be described in detail in this paper. There, the algorithms employed for calculating the distribution of the electromagnetic fields are treated and problems related with the programming of such computational-intensive algorithms in the Java programming language are discussed. Furthermore, it will be shown how the applets could be used in education.

2. Inductors

2.1 Motivation

In lectures dedicated to basics of electrical engineering inductors are often discussed only in a very general way. Typically, the geometry of the magnetic circuit and the number of windings are given and the student learns how to calculate parameters like the magnetic flux density and the inductance. Most practical problems are requiring a quite different approach, i.e. a desired inductance is given and the engineer has to find the optimum geometry of the magnetic circuit and/or number of turns with respect to volume, power losses or realization costs. Frequently, student do experience difficulties with

- understanding the interdependence between the magnetic circuit geometry and the number of turns under the consideration of the maximum admissible magnetic induction of the magnetic core material, the available area for the windings, and a maximum tolerable temperature of the windings
- understanding that an air gap is necessary for preventing magnetic saturation of the core, and/or for storing the magnetic energy of the inductor
- understanding that a defined inductance can be realized using a magnetic core with small cross section in combination with a high number of turns, or vice versa.

Such deficiencies have motivated the development of an interactive Java applet for showing the fundamental relationship and interdependence of the geometry of the magnetic circuit, i.e. the cross section of the magnetic core, the width of the air gap, and the windings area, and the number of turns, the cross section of the current carrying conductors, the magnetic flux density, and the temperature of the windings. Furthermore, the distribution of the magnetic flux of the inductor is calculated using a finite element method (FEM). Therefore, the student also could gain a very detailed insight into the physics of the inductor, e.g. by studying the applets it becomes immediately clear how the magnetic core of the inductor influences and guides the magnetic flux lines.

2.2 Applets Describing General Magnetic Fields and Magnetic Flux

The inductance is the magnetic flux divided by the current causing the magnetic flux density **B** which defines the magnetic flux. Therefore it is important to have a clear understanding of the relation between current, resulting magnetic flux density and magnetic flux as a basis for understand the physical background of inductance.

For making students familiar with the basics of magnetic fields, an applet visualizing the magnetic flux lines of conductors of infinite length orientated perpendicular to the drawing plane is provided (cf. **Fig.1**). The student can change the position of the conductors by dragging with the mouse and experience how different positions do influence the field distribution as visualized and described by the magnetic flux lines. The applet is self-explaining and the new field distributions are calculated very fast so that the student gets the impression of an immediate reaction of the software.

The applet depicted in **Fig. 1** shows the magnetic field distribution (magnetic flux lines) in free space. However, in technical systems materials do show magnetic properties and therefore do take influence on the magnetic field. For calculating the magnetic field in the presence of permeable material one has to refer to a finite element analysis. A detailed discussion of the finite element algorithm which is used in the applets (like the one shown in **Fig. 2**) and also of the implementation in the Java programming language is presented in **section 5**. In this section the discussion is restricted mainly to educational issues for the sake of clarity.

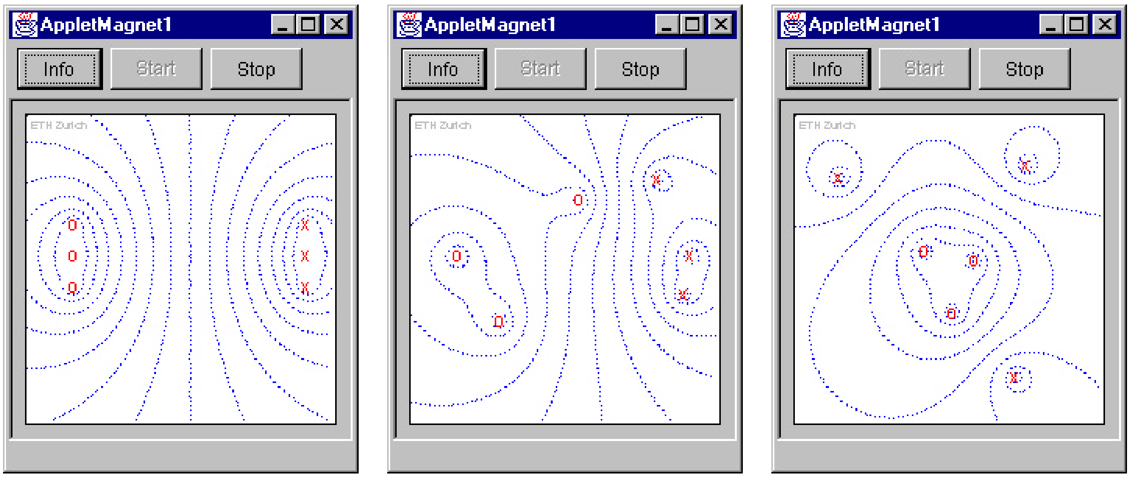


Fig. 1: Magnetic flux lines of impressed currents of opposite directions in conductors of infinite length oriented perpendicular to the drawing plane. The positions of the current carrying conductors can be changed with the mouse.

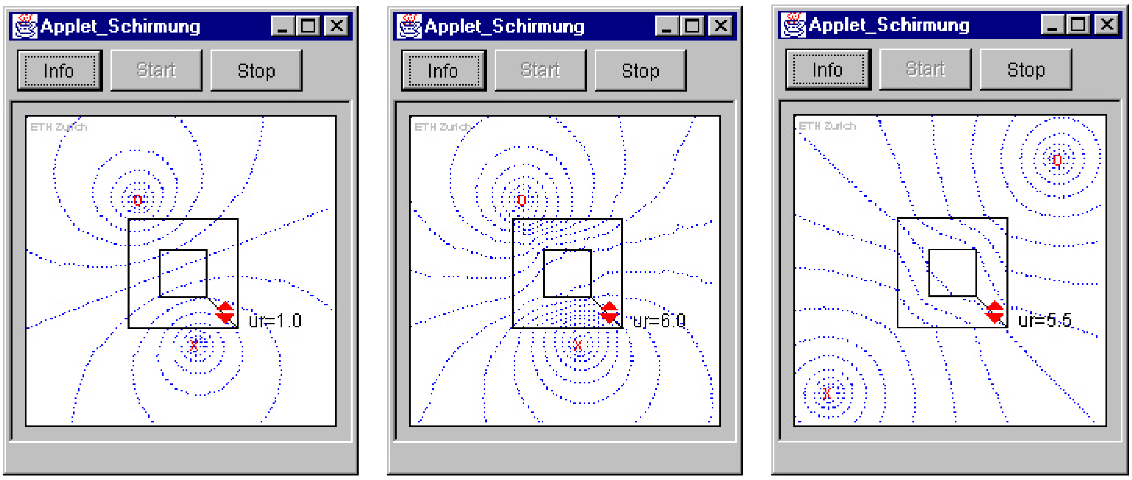


Fig. 2: By changing the permeability of the rectangular magnetic material one can experience how the magnetic flux lines are influenced by permeable material.

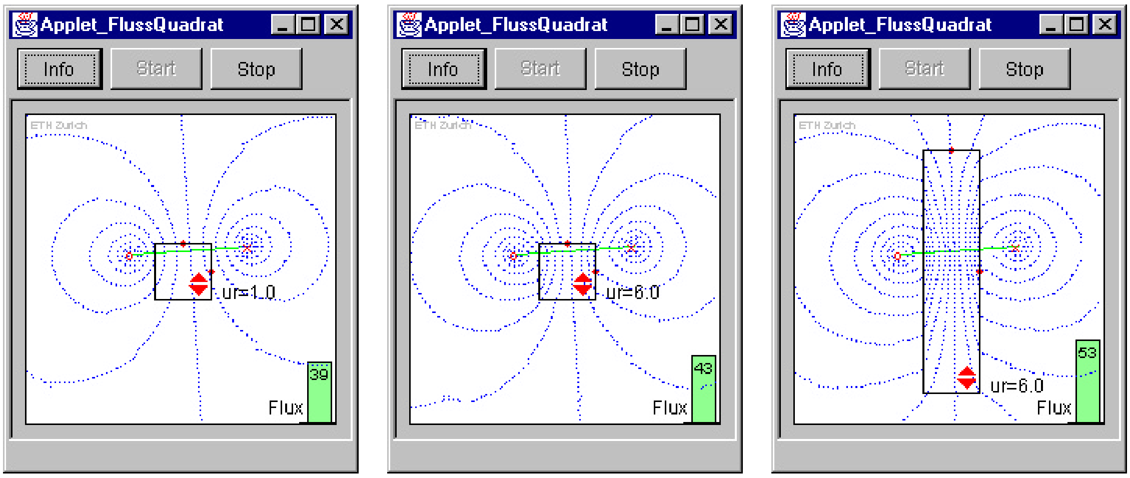


Fig. 3: By changing the size, shape and permeability of the rectangular shaped magnetic material one can see how the magnetic flux lines are influenced by material with magnetic properties. Furthermore, the influence of the magnetic material and of the geometry on the magnetic flux is shown.

The applet depicted in **Fig. 2** shows the magnetic field distribution (visualized by the magnetic flux lines) in the presence of permeable material. The current carrying conductors again can be dragged with the mouse. The permeability of the magnetic material can be changed by clicking one of the two triangular symbols shown in red. The relative permeability m_r can be increased starting from *1.0* (free space). In **Fig. 2** the permeability is set to $m_r=6.0$ and the influence of the material on the magnetic flux lines is obvious. The applet shown in **Fig. 2** was designed to show the effect of magnetic shielding.

Finally, **Fig. 3** shows one of the applets designed for explaining the meaning of magnetic flux. The plane lying in between the two conductors is represented by a green line, the magnitude of the magnetic flux through this plane is calculated and visualized by a bar diagram in the right-bottom corner of the applet.

With this, the student immediately learns that the relative position of the conductors (carrying constant current) does take a strong influence on the magnetic flux. The larger the distance the larger the magnetic flux. Furthermore, the student experiences that increasing the relative permeability m_r and/or the dimensions of the magnetic material do increases the magnetic flux.

2.3 Applets Describing Inductance

The applet depicted in **Fig. 4** shows a coil (inductor) without a magnetic core. The shape of the coil can be changed by dragging one of the two the red dots on the circumference. The student could see that a long coil results in a magnetic field which is approximately constant within a large region inside the coil. If the coil is very short the magnetic field is not constant but more like a field resulting from two parallel conductors (as shown in **Fig. 1**).

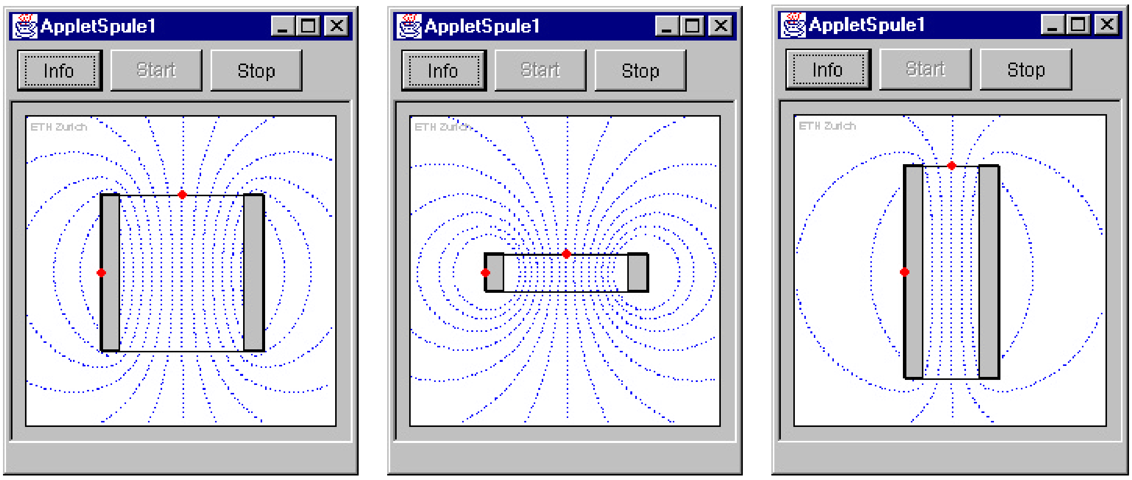


Fig. 4: The dimensions of the coil can be changed by dragging the red dots with the mouse. The applet shows the magnetic flux lines. While changing the dimensions of the inductor the value of the ampere-turns remains constant.

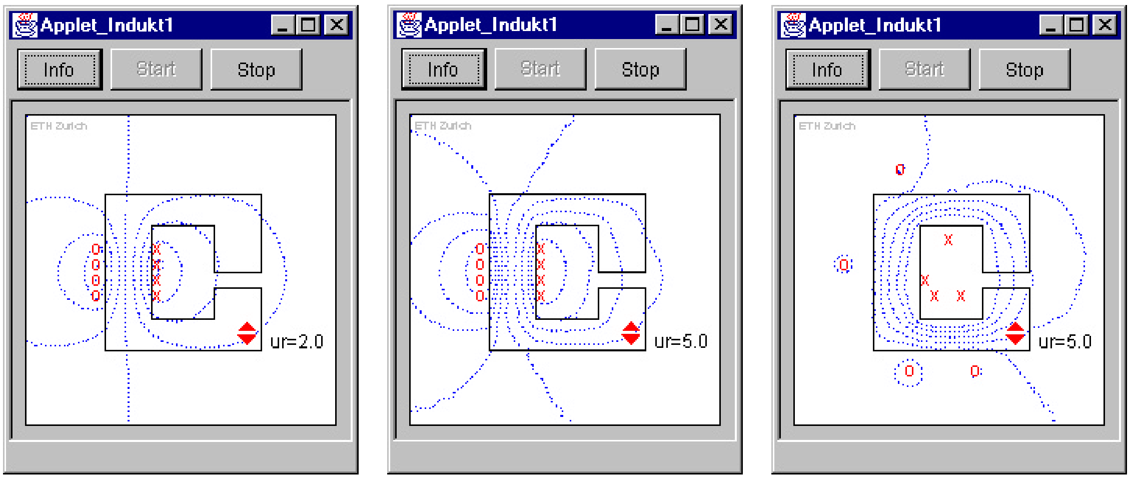
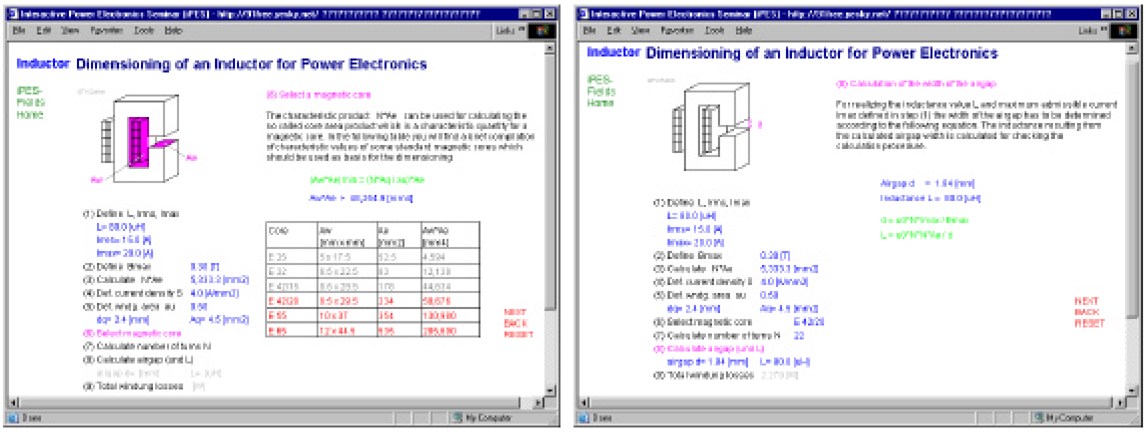


Fig. 5: By changing the permeability of the magnetic core one can see how the magnetic flux distribution is influenced by magnetic material. The shape of the magnetic material is typical for an inductor.

The applet shown in **Fig. 5** should help the student to understand the field distribution in case of the presence of a magnetic core with a high relative permeability m_r. The current carrying conductors can be dragged with the mouse in order to change their positions. The relative permeability m_r of the magnetic core can be increased starting from *1.0* (free space) to higher values by simply clicking on the triangular symbols with the mouse. The student learns that magnetic material does increase the magnetic field and, on the other hand, concentrates the magnetic flux into the core material. Since the current in the conductors is constant, the inductance of an inductor can be increased by increasing the relative permeability of the core material. This increase is limited by the air gap, as will be shown by the applet discussed in the following section.

2.4 Design of an Inductor

After learning about magnetic flux density **B**, magnetic flux lines, stray fields, magnetic material and/or permeability the student has to design an inductor. As typically given in power electronics practice the rms value and peak value of the current, and the desired value of the inductance are defined and the conductor diameter (cross section), geometry and permeability of the core material, the number of turns and the width of the air gap have to be selected. These parameters have to be selected during the design process considering target functions like minimum size, minimum volume or weight or minimum costs.


Interactive Power Electronics Seminar (iPES) - http://ifiiter.peaky.net/
File Edit View Favorites Tools Help
Links
Inductor
IPES-Fields Home
Air
Air
NEXT
BACK
RESET
Done
My Computer

Dimensioning of an Inductor for Power Electronics

(6) Select a magnetic core

The characteristic product N*Ae can be used for calculating the so called core area product which is a characteristic quantity for a magnetic core. In the following table you will find a short compilation of characteristic values of some standard magnetic cores which should be used as basis for the dimensioning.

(Aw*Ae) min = (N*Ae) Imax/(S*au)

Aw*Ae > [illegible] [mm4]

(1) Define L, Irms, Imax
L= 80.0 [uH]
Irms= 15.0 [A]
Imax= 28.0 [A]
(2) Define Bmax 0.30 [T]
(3) Calculate N*Ae [illegible] [mm2]
(4) Def. current density S 4.0 [A/mm2]
(5) Def. wdg. area au 0.58
dcu= 2.4 [mm] Acu= 4.5 [mm2]
(6) Select magnetic core
(7) Calculate number of turns N
(8) Calculate airgap (and L)
airgap d= [mm] L= [uH]
(9) Total winding losses [W]

Core	Aw [mm x mm]	Ae [mm2]	Aw*Ae [mm4]
E 25	5 x 17.5	52.5	4,594
E 32	6.5 x 22.5	83	12,138
E 42/15	8.5 x 29.5	178	44,624
E 42/20	8.5 x 29.5	234	58,676
E 55	10 x 37	354	130,980
E 65	12 x 44.5	535	285,690

Dimensioning of an Inductor for Power Electronics

(8) Calculation of the width of the airgap

For realizing the inductance value L and maximum admissible current Imax defined in step (1) the width of the airgap has to be determined according to the following equation. The inductance resulting from the calculated airgap width is calculated for checking the calculation procedure.

Airgap d = 1.84 [mm]
Inductance L = 80.0 [uH]

d = u0*N*Imax / Bmax
L = u0*N*N*Ae / d

(1) Define L, Irms, Imax
L= 80.0 [uH]
Irms= 15.0 [A]
Imax= 28.0 [A]
(2) Define Bmax 0.30 [T]
(3) Calculate N*Ae 5,333.3 [mm2]
(4) Def. current density S 4.0 [A/mm2]
(5) Def. wdg. area au 0.58
dcu= 2.4 [mm] Acu= 4.5 [mm2]
(6) Select magnetic core E 42/20
(7) Calculate number of turns N 22
(8) Calculate airgap (and L)
airgap d= 1.84 [mm] L= 80.0 [uH]
(9) Total winding losses [illegible]

Fig. 6: The applet shows an intuitive and user-friendly step-by-step approach for dimensioning an inductor for a power electronics converter. The calculations are based on simplified analytical equations as given in the applet. Stray fields are neglected.

Since there are a lot of parameters to be considered it is often confusing for the student how to start. The applet depicted in **Fig. 6** guides the student through the design process, gives hints what to do and why, shows the relevant mathematical relations and does perform all calculations in the background. There, the equations used are based on approximations like neglecting stray fields. Furthermore, the available core geometries are limited to a few standard types.

3. Magnetic Fields in Rotating Machinery

Besides inductors and transformers, electric machines are a main topic for applied electromagnetic field theory. For electric machines besides changes of the fields over time also changes of the geometry could occur. A basic property of electric machines is a rotating magnetic field. Such rotating field can be formed by feeding conductors lying on a circle and displaced by $2\pi/3$ with a symmetric three-phase current system, as formed by phase currents of equal amplitudes and a phase displacement of $2\pi/3$. The resulting rotating magnetic field is visualized by the applet.

If in the applet depicted in **Fig. 7** the “RUN”-button is clicked the vertical time-marker starts to move and the magnetic field starts to rotate, accordingly. The student could change other parameters with the mouse and could e.g. experience that changing the positions of the conductors does result in a distortion of the constant magnitude and/or constant angular speed of the rotating field.

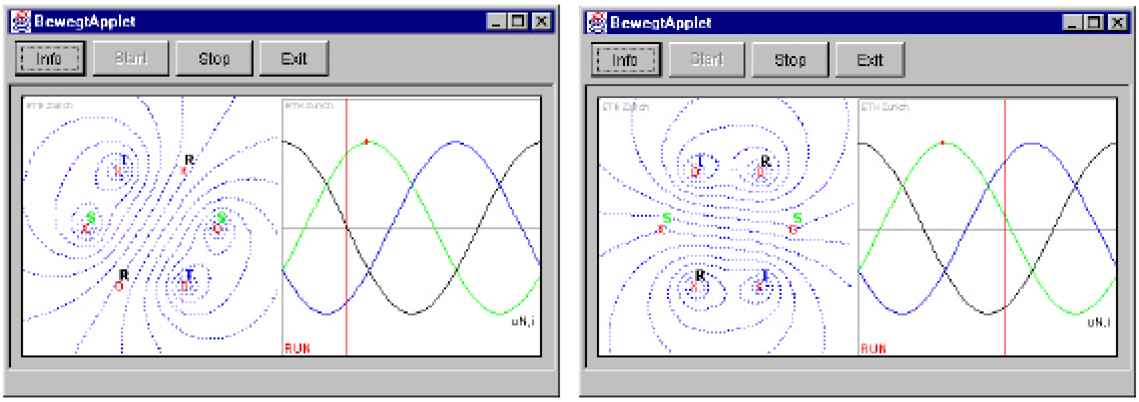


Fig. 7: The symbols "x" and "o" shown in red do denote impressed currents of opposite directions in conductors of infinite length oriented perpendicular to the drawing plane. Always two conductors of equal denomination can be considered to form a phase winding.

4. Current Research to be Released in the Future

4.1 Transformer

In introductory lectures on electrical engineering, which are the basis of more specialized courses like power electronics, transformers are often explained in a strict mathematical manner in addition to equivalent circuits. Therefore, students do have difficulties with getting an understanding of the fundamental physical principles of a transformer, e.g., frequently the fundamental relationship between the magnetic flux and the applied voltage and the physical reason for the balance between the primary and secondary ampere turns is not fully understood. Furthermore, the magnetic stray field and/or stray inductance of a transformer often remains mysterious to the students.

The authors are currently developing applets (for future release) which can be used to teach and demonstrate the following relationships:

- the magnetic flux in the transformer magnetic core is in a first approximation independent from the load connected to the secondary as primary current changes such that the both current flows do cause a magnetic flux as defined by the impressed primary input voltage
- the magnetic flux is defined by the applied voltage and therefore does not change if an air gap is inserted into the magnetic circuit and/or the width of the air gap only does take influence on the magnetizing current in a first approximation
- a short circuit of the secondary (representing an extreme load condition, like $R_{LOAD}=0$) prevents changes of the magnetic flux linked with the secondary winding, therefore, the transformer magnetic field can only exist as stray field in between primary and secondary. This clearly shows that in this case the transformer input current is defined by the stray inductance what can be proven by changing the physical location of the primary winding in relation to the secondary winding.

4.2 Capacitor

A collection of Java applets for studying the basic behavior and design of capacitors from the viewpoint of power electronics is currently under development. The applets will explain the electric field intensity **E** and the electric scalar potential of a geometry carrying electric charges. The physical meaning of electrical field intensity and its relation to the electric scalar potential will be shown, furthermore the meaning of voltage will be shown by the difference of the electrical scalar potentials in between two locations.

The electric field intensity **E** of geometries (like parallel plates of a capacitor) will be calculated in a way comparable to the applets shown in **section 2**, i.e. the finite element method will be employed in the applets in order to show the influence of dielectric material on the electric field intensity. One applet will show how the capacitance of a capacitor changes if material with a relative permittivity $e_r > 1$ is inserted in between the two plates. The material can be dragged to different positions with the mouse, and the relative permittivity e_r can be increased by clicking an according symbol with the mouse.

While the applets described above should help the student to understand the physics of a capacitor, there furthermore will be an applet to do a practical design (helping to define selection criterions) of a capacitor for power electronics applications.

4.3 Heat Sink

Switching and conduction losses of power semiconductors do increase the junction temperature. In order to limit the junction temperature which directly does influence the useful life a low thermal resistance in between junction and ambient has to be provided what could be achieved by employing a heat sink.

For teaching the student the basics of temperature increase due to heat conduction, the concept of thermal resistance and the corresponding electric equivalent circuit, as well as the transient thermal behavior (i.e. thermal capacitance) the authors are currently developing Java applets for visualizing the basics of thermodynamics related to power electronics. Some of the applets will employ finite element methods while others will be based on analytical calculations. These applets to be released in near future will constitute another *iPES* module (*"iPES-Thermal"*) and will have interfaces comparable to the applets described in this paper.

5. Finite Element Method Implemented by a Java Applet

5.1 The Finite Element Method

The electromagnetic field is described by the four Maxwell equations:

$$\nabla \times \boldsymbol{H} = \boldsymbol{S} + \partial \boldsymbol{D} / \partial t \qquad \nabla \times \boldsymbol{E} = - \partial \boldsymbol{B} / \partial t \qquad \nabla \boldsymbol{B} = 0 \qquad \nabla \boldsymbol{D} = \rho \qquad (1)$$

with the three-dimensional time-dependent vector fields

$$\boldsymbol{D} = e\, \boldsymbol{E} \qquad \text{and} \qquad \boldsymbol{B} = m\, \boldsymbol{H} \qquad (2)$$

D … electric flux density ***E*** … electric field intensity ε … permittivity
B … magnetic flux density ***H*** … electric field intensity μ … permeability

The current density **S** is also a three-dimensional time-dependent vector field, while the electric charge density ρ is a scalar function dependent on position *(x,y,z)* and time *(t)*. If one knows the distribution of the electromagnetic field, i.e. ***E****(x,y,z,t),* ***D****(x,y,z,t),* ***H****(x,y,z,t),* and ***B****(x,y,z,t)* it is straight forward to calculate the sources ***S****(x,y,z,t)* and *r (x,y,z,t)* of this electromagnetic field distribution using the equations given above. Unfortunately, in practice the sources like the location and current of a winding are well known, but the resulting electromagnetic fields have to be calculated. Generally, this is not easy to perform.

The finite element method (FEM) is a mathematical method, which could be used for solving such problems in approximation. The idea is to divide the area of the unknown field into geometrically small elements, e.g. triangles or rectangles. The equations describing the field are linearized within each geometric element and the linearized equations are coupled to each other resulting in a large set of equations. The number of elements directly correlates with the number of equations to be solved. The FEM has proven to be very effective for calculating electromagnetic field distributions. Since a large number of elements is needed for achieving a given accuracy of the solution, the set of equations which is represented by a matrix becomes very large what results in a long computation time.

5.2 Finite Element Method in Magnetostatics Employed in *iPES-Fields*

In order to calculate electromagnetic field distribution in an interactive Java applet the points as listed in the following have to be considered.

- The algorithm should show a very low computation time.

iPES-applets performing numerical *circuit simulation* are based on the following idea: Whenever the user changes a parameter with the mouse, the new parameter value is used by the simulation algorithm for calculate the new time behavior shown in an according applet. The impression of the user should be that the new time behavior is not calculated in the background but present immediately. This should also work on a slow computer (*230 MHz* CPU) because students studying at home are a large share of the user group. Therefore, simple Euler- and Trapezoidal-algorithms are employed in most of the circuit simulations. To the experience of the authors as long as the calculation in the background takes less time than about *200ms* the user regards the applet to be immediate responsive. If the calculations running after a parameter has been changed do take longer than about *1.0 second* the responsiveness of the applet is not acceptable for many users. Accordingly, making the algorithms fast enough (calculation time not longer than *2* or *3 seconds* on a *230 MHz* PC) is of major importance for *FEM-applets*.

- The compiled code for the algorithm should be of small size (small number of bytes)

Whenever opening the web page containing an applet, the whole applet-code is downloaded. While on a fast Internet-connection as installed at many universities and companies downloading even a few Megabyte is completed within seconds, it is essential to keep the size of the applet below *100kB* in order to make it acceptable for people connecting to the Internet with modems. Therefore, *iPES* applet size is always below *70kB* what typically is comparable to one or two pictures in JPG-format. Putting a FEM-algorithm into a Java applet can be done under the restrictions of size as long as the menu-options for the user are limited and the applet is dedicated to a specific topic like the distribution of the magnetic field of an inductor.

- The post-processing has to be done in a very short time.

Post-processing means that field distribution is visualized employing field lines, equipotential lines, vector fields and the like. For the visualization of the magnetic field distribution one has to calculate the magnetic flux lines. If the FEM algorithm directly calculates the magnetic vector potential $\boldsymbol{A_P}\,(x,y,z)$, the equipotential lines of the magnitude of $\boldsymbol{A_P}\,(x,y,z)$ are the magnetic flux lines to be shown [4]. There, equal magnitude increments between neighboring flux lines have to be considered. If the FEM algorithm gives field quantities like $\boldsymbol{B}(x,y,z)$ or $\boldsymbol{H}(x,y,z)$, the vector potential $\boldsymbol{A_P}$ has to be calculated from those field quantities resulting in significantly increased computation effort and reduced accuracy. Since the whole algorithm has to be as fast as possible, a FEM algorithm calculating directly $\boldsymbol{A_P}\,(x,y,z)$ has been selected in order to reduce the post-processing efforts.

- Minor errors in the resulting field distributions are acceptable because the applet is for educational purposes and its aim is to show basics of the electromagnetic field, and not the detailed calculation of a technical problem.

This means that the number of elements for describing the problem can be relatively low. This reduces the accuracy of the results but also does reduce the calculation time significantly. For making the applets interactive (in the style of the circuit simulation applets of the *iPES* project, *iPES-Circuits*) reduced calculation time is essential as long as the quality of the results is sufficient to teach the theory.

Considering the points listed above the FEM algorithm described in the following was selected and is employed in the Java applets. The currents flow always is in conductors of infinite length oriented perpendicular to the drawing plane. All calculated fields are two-dimensional.

In case of isotropic magnetic material the electromagnetic field is described according to

$$\boldsymbol{B} = m_0\,(\boldsymbol{H} + \boldsymbol{M}) = m_0\, m_r\, \boldsymbol{H} \qquad (3)$$

$$\boldsymbol{H} = 1/(m_0\, m_r)\, \boldsymbol{B} \qquad (4)$$

$$\boldsymbol{M} = 1/m_0\,(1 - 1/m_r)\, \boldsymbol{B} \qquad (5)$$

For a two-dimensional magnetic flux density $\boldsymbol{B}(x,y)$ in the (xy)-plane and one or more current-carrying conductors orientated in z-direction we can write

$$\boldsymbol{B}(P) = (m_0\, I_1)/(2\pi\, r_{P,I1})\, \boldsymbol{e_B} \qquad r_{P,I1} \ldots \text{ distance between point } P \text{ and conductor (current } I_1\text{)} \qquad (6)$$

$$\boldsymbol{B}(P) = \Sigma_{(i)}\, (m_0\, I_i)/(2\pi\, r_{P,Ii})\ \boldsymbol{e_{B,i}} \qquad r_{P,Ii} \ldots \text{ distance between point } P \text{ and conductors (current } I_i\text{)} \qquad (7)$$

Regarding to the elementary current theory of magnetism [5] the magnetic flux density of material with permeability different from air is caused by so-called elementary currents within the material. If we divide the magnetic material into many small two-dimensional geometric elements, then each element has such elementary currents flowing on the element's border in z-direction. Under this assumption the magnetic flux density in each point can be calculated from currents impressed in current conductors and elementary currents on the border of the elements.

$$\boldsymbol{B(P)} = \Sigma_{(k)} (m_0\, I_k)/(2\pi\, r_{P,Ik})\, \boldsymbol{e_{B,k}} + \Sigma_{(i)} (m_0\, I_{Ei})/(2\pi\, r_{P,IEi})\, \boldsymbol{e_{B,i}} \qquad (8)$$

I_k .... impressed current of conductor k

I_{Ei} .... elementary current of geometric element i

The magnetic flux density ***B*** can be calculated directly from all currents (impressed conductor currents and elementary currents) because of the 3rd Maxwell equation ***B*** is a pure curl-field. This is not true for the magnetic field intensity ***H***. In the following, we will calculate ***B***, therefore.

Generally, for two-dimensional field-components we can write for the vector field of the magnetization vector ***M(x,y,z)*** at the boundary between two materials 1 and 2 with vector ***n*** (normalized, pointing from material 1 into material 2) defining the shape of the border [5]

$$\nabla \times \boldsymbol{M} = \boldsymbol{n} \times (\boldsymbol{M}_2 - \boldsymbol{M}_1) = \mathrm{a}_{\mathbf{E}} = \alpha_{\mathrm{E}}\, \boldsymbol{e}_{\mathbf{Z}} \qquad (9)$$

For each geometric element 1 and the neighboring element 2 the density of the elementary current α_E (flowing in z-direction at the border between the two geometric elements) can be calculated as

$$\alpha_{\mathrm{E}}\, \mathbf{e_Z} = I_E/\Delta d\, \mathbf{e_Z} = \mathbf{n} \times (\boldsymbol{M_2}\text{-}\boldsymbol{M_1}) = \boldsymbol{n} \times 1/m_0\, (1 - 1/m_r)\, (\boldsymbol{B_2}\text{-}\boldsymbol{B_1}) \qquad (10)$$

where Δd is the length of the boundary between the two geometric elements, $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ are the magnetic flux densities (to be calculated) within the two elements, and m_r is the magnetic permeability of the two elements. If magnetic saturation has to be considered the permeability m_r of each element has to be defined individually for each element.

Solving Eq.(10) we can express the elementary current I_E flowing in z-direction at the border of the two neighboring elements as

$$\mathrm{I_E} = \Delta d\, /m_0\, (1 - 1/m_r)\, (n_X\, DB_Y - n_Y\, DB_X) \qquad (11)$$

with

$$D\boldsymbol{B} = \mathbf{B_1} - \mathbf{B_2} = (DB_X,\, DB_Y,\, 0) = (B_{1X}\text{-}B_{2X},\, B_{1Y}\text{-}B_{2Y},\, 0) \qquad (12)$$

as the difference field and $\boldsymbol{n} = (\mathrm{n_X}, \mathrm{n_Y}, 0)$.

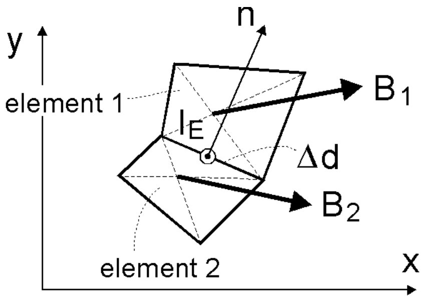


Fig. 8: Magnetic material modeled with two elements. The elementary current I_E between the two elements 1 and 2 is calculated according to equation (11).

Because the elementary current I_E can be calculated from the difference of the magnetic flux densities $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ of two neighboring elements 1 and 2 (equation 11), and both magnetic flux densities $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ can be calculated from all currents (equation 8), we can eliminate the magnetic flux densities $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ and express the elementary current I_E at a certain point as sum of quantities dependent on the other elementary currents between all other geometric elements, the impressed currents in the conductors, the geometric distances and the permeability.

Equation (11) shows the dependency of the unknown elementary current I_{E1} between two geometric elements. If we divide the magnetic material into many small geometric elements, between always two such neighboring elements an unknown elementary current has to be considered which can be express by an equation in dependency of all the other elementary currents and the impressed currents. If there are *n* unknown elementary currents, then there are *n* such equations forming a system of *n* linear equations which can be written and solved in matrix form

$$\underline{A}\;\underline{x} = \underline{b} \tag{14}$$

with $\underline{x}$ as the vector holding all unknown elementary currents, $\underline{A}$ as the (n x n)-matrix defined by the geometry of the problem (and with the permeability describing the magnetic material), and $\underline{b}$ as the vector considering the impressed currents in the conductors.

The matrix $\underline{A}$ is dense and not symmetric. This means that the solution of Eq. (14) can be very problematic because rounding-errors do influence and distort the correct solution and could cause instability of the solution for certain parameters. On the other hand the approach shows two main advantages. First, only the magnetic material has to be divided into geometric elements, while all fields outside magnetic materials can be calculated easily using Eq. (8). In the magnetic field problem considered large areas with non-magnetic material (free space) are surrounding magnetic material. This means that a few elements might be sufficient to describe the problem resulting in a small matrix $\underline{A}$. In the Java applets the number of the unknown elementary currents typically is between *20* and *200*. The calculation time on an *800 MHz* PC remains under about *0.5 seconds* in case of a *(200 x 200)*-matrix $\underline{A}$. The second advantage comes from a significant reduction of the post-processing effort. The FEM-algorithm results in the matrix equation (cf. Eq. (14)) with the elementary currents as solution. From these elementary currents the magnetic flux density ***B*** can be calculated directly (cf. Eq. (8)), but also the magnetic vector potential $\boldsymbol{A_P}$, required for defining the magnetic flux lines, can be calculated directly according to

$$\boldsymbol{B}(P) = \nabla \times \boldsymbol{A_P} = (\partial/\partial x, \partial/\partial y, \partial/\partial z) \times (A_{P,X}, A_{P,Y}, A_{P,Z}) \tag{15}$$

$$\boldsymbol{B}(P) = ((\mu_0\, I)/(2\pi\, y), 0, 0) \tag{16}$$

y …. distance between point P and conductor (current *I*)

$$\partial A_{P,Z} / \partial y = (\mu_0\, I)/(2\pi\, y) \tag{17}$$

$$\boldsymbol{A_{P,Z}} = (0, 0, -(\mu_0\, \mathrm{I})/(2\pi \ln(y))) \tag{18}$$

Knowing $\boldsymbol{A_P}$ the magnetic flux lines (which are typically used for visualizing the solution of the field problem) can be computed directly as equipotential lines of the scalar magnitude of $\boldsymbol{A_P}\,(x,y)$.

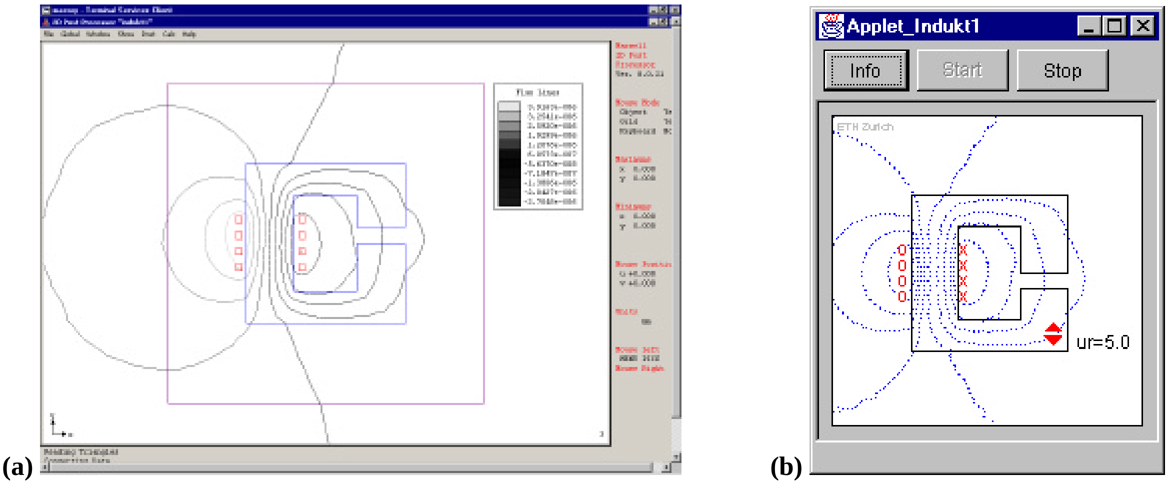


Fig. 9: (a) Computation of the flux lines for $\mu_r = 5$ using Maxwell 2D; **(b)** computation of the flux lines ($\mu_r = 5$) by the Java applet.

The FEM algorithm described above can be realized easily by a Java applet, the size of the applet is far below the limit of *70kB*, and the calculation time is sufficiently low, i.e. about *0.5 seconds* on a *800 MHz* PC for each calculation for a magnetic material as shown in **Fig. 9(b)** divided into *178 geometric elements*. For evaluating the accuracy of the FEM algorithm the flux lines of a given geometry were calculated with the Java applet and compared to the result of a Maxwell 2D [10] simulation (cf. **Fig. 9(a)**). The very close accordance of both calculations can be verified in **Fig. 9**.

6. Conclusions

In this paper educational software (a collection of Java applets) has been introduced for visualizing magnetic fields of inductors and magnetic components in an interactive style. The calculation of the magnetic flux lines is

based on a FEM (finite element method) algorithm based on elementary currents which gives results in good accordance with calculations performed by the field solver Maxwell 2D. By choosing a FEM algorithm as described in the paper and optimizing the program code the calculation times are kept very short.
The paper shows that it is possible to do FEM analysis inside a Java applet on a standard *800 MHz* PC within *one second* as long as the number of unknown elements is smaller than about *300*. Such a short calculation time is necessary to make the applets interactive in the style of the *iPES*-project. For the future (2003) it is planned to release educational Java applets to show stray fields in transformers, to show thermal effects relevant for power electronic devices and systems, and to visualize internal fields of power semiconductors (like diodes).

Appendix

A. The Java Programming Language

All applets as described in this paper are written in Java. Java is a very popular object-orientated computer programming language with syntax like C/C++. Java is easier to learn than C or C++ (e.g., in Java there are no pointers), and is designed and optimized keeping safety, platform independence and networking applications in mind. A Java program runs inside a so-called Virtual Machine (VM). This is, roughly speaking, a kind of software emulating a microprocessor, where the Java program is interpreted by this VM. VMs can be considered as software layer above the operating system and are available for various operating systems (Windows, Unix, Linux) via downloading and installing the freely available Java Runtime Environment (JRE) [8]. Furthermore, the JRE (with the VM) is part of and/or available as plug-in for modern web-browsers (like Netscape, Internet Explorer, Opera). If the Java program is embedded in a web page (HTML), the web-browser's VM automatically starts executing the Java program (which is called "applet" in this case) when the web page is opened. Then it is possible to run also complex programs inside the web-browser. Since the Applet only needs a Java-enabled web-browser it is very easy to distribute via Internet and can be immediately used without inconvenient installations.

Originally Java was not designed to do numerical calculations with good efficiency. Therefore, in the applications described in this paper, where FEM analysis is employed, the programming of the numerical algorithms has to be done very carefully with the performance in mind [7].

B. Basic Strategies of Tuning the Code of the FEM Java Applets

Basically, the sequence of computations inside a FEM applet can be characterized as

- setting up the matrix $\underline{A}$
- solving the matrix equation $\underline{A}\ \underline{x} = \underline{b}$
- calculating the flux lines for the whole drawing area
- drawing the flux lines and other graphics

The execution speed of the code will differ on various computers, not only because of different CPUs but also according to different VMs. The Java language defines a *method* (this is the name of a *function* in Java) for measuring the execution time within a code block

```
long t1 = System.currentTimeMillis();
/* code block of which the execution time has to be measured */
long t2 = System.currentTimeMillis();
long delta_t = t2-t1;
```

Using the above-given code scheme it is a simple task to consequently measure the execution times of the code blocks inside the applet and to find out where most time is spent. Experiments showed the following for the applet depicted in **Fig. 9(b)**:

Setting up the matrix and solving the matrix equation is not very time-consuming because the size of the matrix is *(178 x 178)* and/or relatively small.

A considerable amount of time is spent with calculating the flux lines because there are *(200 x 200)* pixel-points and the value of the magnetic vector potential $\boldsymbol{A_P}$ has to be calculated for each point, which means calculating equation Eq.(18) *40,000 times*. Therefore, it does make sense to concentrate performance tuning of the code on this loop. Because the problem is two-dimensional the vector potential has to be calculated based on a natural logarithm. As the calculation of the natural logarithm implemented in the mathematical standard library of Java is extremely slow, an implementation of the logarithm in form of a Taylor-series has been performed, where considering only a limited number of elements of the series does result in some distortions of the flux lines shown. This deficiency will be improved by a better approximation function in a later version of *iPES-Fields*.

Another significant improvement of the calculation speed is made possible by changing the data format of the floating-point numbers inside this *40,000-times* loop from `double` (taking up 8 bytes as specified by the IEEE 754 standard) to `float` (4 bytes, also specified in IEEE 754). This reduction in precision does not take effect on the resulting flux line distribution but does considerably increase the speed of the loop-execution.

Another large improvement could be achieved by using simple predefined drawing-methods of the standard Java object `Graphics` instead of setting each pixel-point directly with `MemoryImageSource` which is often used in animation or game-programming to realize sophisticated graphics.

Such changes of the code have improved the calculation speed by a total factor of about *10*. The parts of the code most promising for performance tuning were identified systematically with the scheme described above. In the meantime more code segments to be optimized concerning calculation speed have been identified and will be improved for a future version of *iPES-Fields*. It is important to mention that employing so-called "*native code*", i.e. executable code, for example resulting from C-compilation, inside the Java code for speeding up time-critical calculations, can not be used inside unsigned Java applets for security reasons (*native code* is not restricted by the security system of the Java Runtime Environment which makes execution very fast but has the potential to hide dangerous computer viruses).

The following execution times have been measured on a notebook-computer with a *Pentium III CPU / 650 MHz and 128 MB RAM*. The times given are average values

Setting up the matrix $\underline{A}$ *(178 x 178)*	10 ms
Solving the matrix equation $\underline{A}\ \underline{x} = \underline{b}$	65 ms
Calculating the flux lines for the whole drawing area	300 ms
Drawing the flux lines and other graphics	2 ms

The following table will give an impression of the average calculation time for different sizes of the matrix $\underline{A}$. The applets described in this paper work with matrix-sizes in between *(20 x 20)* and *(400 x 400)*. The applet shown in **Fig. 9(b)** employs magnetic material modeled by *178* elementary currents.

Number of elementary currents	50	100	150	200	300	400
Average execution time [ms]	110	190	290	470	980	1600

References

[1] **Drofenik U., and Kolar, J. W.:** www.ipes.ethz.ch

[2] **Drofenik U., and Kolar, J. W.:** *Interactive Power Electronics Seminar (iPES) – A Web-Based Introductory Power Electronics Course Employing Java-Applets.* Proceedings of the 33rd IEEE Power Electronics Specialists Conference, Cairns, Australia, June 23-27 (2002).

[3] **Silvester P. P., and Ferrari R. L.:** *Finite elements for electrical engineers.* Third edition. Cambridge University Press, 1996. ISBN 0 521 44953 7

[4] **Sykulski J. K.:** *Computational Magnetics.* First Edition. Chapman & Hall, 1995. ISBN 0 412 58570 7

[5] **Hofmann H.:** *Das elektromagnetische Feld: Theorie und grundlegende Anwendungen.* 3.Auflage. Wien; New York: Springer, 1986. ISBN 3-211-81918-5

[6] **Dirschmid H. J.:** *Mathematische Grundlagen der Elektrotechnik.* Vierte Auflage. Braunschweig; Wiesbaden: Vieweg, 1988. ISBN 3,528; 33034;1

[7] **Shirazi J.:** *Java Performance Tuning.* O'Reilly & Associates, 2000. ISBN 0-596-00014-4

[8] www.java.sun.com

[9] **Newman, M.J., Trowbridge, C.W. and Turner, L.R.:** *GFUN: an interactive program as an aid to magnet design.* Proceedings of the 4th International Conference on Magnet Technology, Brookhaven, N.Y, USA, pp. 617-626 (1972).

[10] www.ansoft.com/products.cfm