EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics

Large Hadron Collider Project

LHC Project Report 1005

INTEGRATION OF MATHEMATICA IN THE LARGE HADRON COLLIDER DATABASE

J. Beauquis, E. Wildner

Abstract

The CERN Large Hadron Collider (LHC) is the major project in particle physics in the world. The particle accelerator is a 27 km ring where many thousands of superconducting magnets keep protons on track. Results from complex measurements of, for example, the magnetic field and the geometry of the main bending and focusing magnets are stored in databases for analysis and quality control. The geometry of the 15 m long main bending magnet weighing almost 30 tons has to be controlled within tenths of mm. All measurements are stored in ORACLE data bases. They are organized in two types: raw and derived data. Raw data come from the measurement devices and derived data describe quality measures calculated from the raw measurements. For example, the transverse position of the beam tube center relative to the theoretical axis of the accelerator is measured along the magnet. This data is used to simulate improvements or to calculate quality criteria, used in the daily quality checks of all produced magnets. The positions of the corrector magnets housed inside the magnet assembly are measured in industry before the closing of the magnet cold mass; they have to be calculated from reference points on the outside of the cold mass one measured after delivery to CERN. The results from these calculations are re-injected into the data base for easy access. The calculations cannot be performed by the ORACLE query language. There comes the interest of Mathematica, which is easy to interface with the existing ORACLE and Java environment. Maintenance and improvements of calculations are comfortable due to Mathematica's explicit functional language.

CERN, Accelerator Technology Department, Geneva, Switzerland

Presented at IMS'06 (8th International Mathematica Symposium)
19-23 June 2006, Avignon, France

CERN
CH - 1211 Geneva 23
Switzerland

Geneva, 26 March 2007

International Mathematica Symposium 2006

Integration of Mathematica in the Large Hadron Collider Database

J. Beauquis
E. Wildner

The CERN Large Hadron Collider (LHC) is the major project in particle physics in the world. The particle accelerator is a 27 km ring where many thousands of superconducting magnets keep protons on track. Results from complex measurements of, for example, the magnetic field and the geometry of the main bending and focusing magnets are stored in databases for analysis and quality control. The geometry of the 15 m long main bending magnet weighing almost 30 tons has to be controlled within tenths of mm. All measurements are stored in ORACLE data bases. They are organized in two types: raw and derived data. Raw data come from the measurement devices and derived data describe quality measures calculated from the raw measurements. For example, the transverse position of the beam tube center relative to the theoretical axis of the accelerator is measured along the magnet. This data is used to simulate improvements or to calculate quality criteria, used in the daily quality checks of all produced magnets. The positions of the corrector magnets housed inside the magnet assembly are measured in industry before the closing of the magnet cold mass; they have to be calculated from reference points on the outside of the cold mass one measured after delivery to CERN. The results from these calculations are re-injected into the data base for easy access. The calculations cannot be performed by the ORACLE query language. There comes the interest of *Mathematica*, which is easy to interface with the existing ORACLE and Java environment. Maintenance and improvements of calculations are comfortable due to *Mathematica*'s explicit functional language.

■ Introduction

In the future Large Hadron Collider, currently under construction at CERN, Geneva, two counter rotating beams will provide proton-proton collisions at 7 TeV center of mass energy. One of the major efforts toward the accelerator completion is the manufacturing of the 1232 superconducting twin aperture magnets required to bend the particle beams along the 27 km long circular trajectory. The manufacturing procedure of each 15m long, 0.5m wide and 30 ton heavy magnet is strongly driven by the respect of the tight geometrical tolerances imposed by the accelerator beam performance. In order to ensure the maximum transverse clearance for the circulating particles with the minimum coil size, the central part of the magnet is bent in the horizontal plane of about 5mrad. In addition, to compensate for systematic field-harmonics in the dipole field, multipolar correctors are placed in one or both ends of each dipole. Thus, to satisfy the geometrical tolerances imposed on the dipole shape, the central part should be enclosed in a torus of 1.3 mm radius and the ends in two cylinders of 0.6 mm radius with respect to the geometric axis of the dipole. On the other hand all the correctors should stay in average within 0.3 mm from the geometric axis with a standard deviation of the error not in excess of 0.5 mm.

The dipole geometry must be preserved throughout the assembly (in industry and at CERN), transport and testing circumstances (cold test) till the positioning in the tunnel and the start of the operative stages.

We will focus on two aspects of the geometry: the calculation of the sagitta of the bent magnet and of the overall shape of the excursion of the magnet center along the magnet, to show the how we use a *Mathematica* package to update our data base with derived parameters not possible to calculate within the data base.

■ Description of the problem

□ LHC dipole geometry

The theoretical geometry of the twin aperture dipole is shown in figure. The (x,y) plane is the plane of the accelerator. The two theoretical beam-trajectories consist of an arc of a circle, extending over the magnetic field, with the angle Θ and the radius ρ and of straight lines between magnets. The 3 dimensional measurements of the centre of the two cold bore tubes, in which the beams are circulating, are fit to the two theoretical curves. The plane obtained in this way is called the mean plane and the axes the geometrical axes and this defines the coordinate system in whicg we express the results of the measurements.

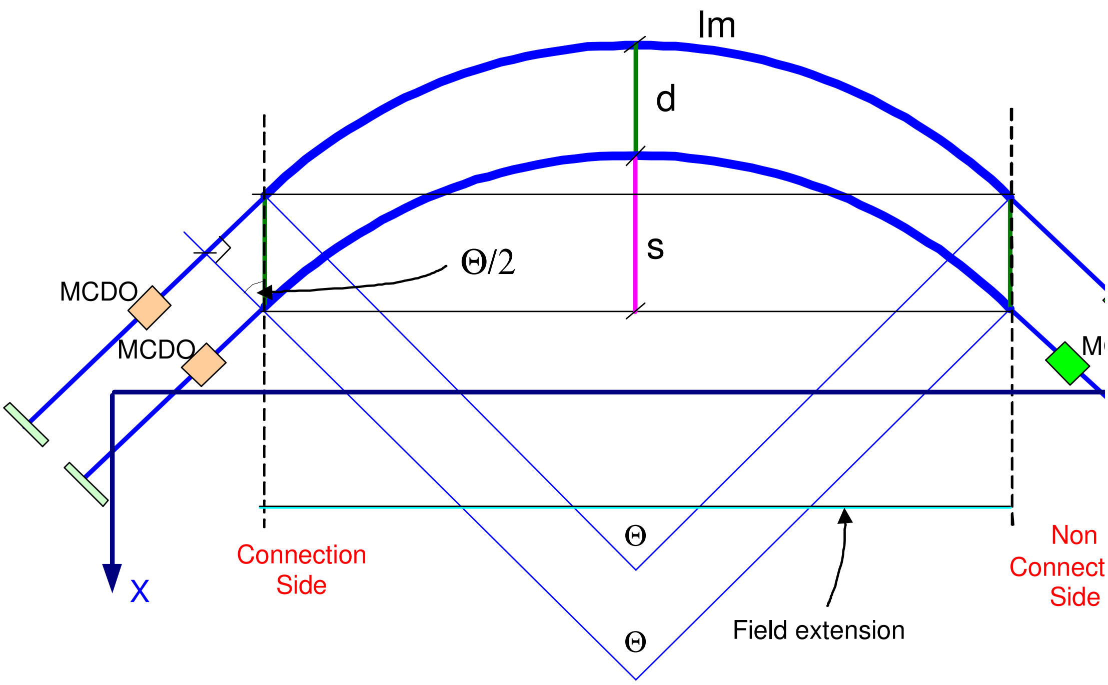

Figure 1: Geometry of LHC dipole

Some of the important parameters of the geometry are given in table 1.

Table 1 : Dipole geometry parameters at room and operational temperature

Parameter	Symbol	Value warm	Value cold	Unit
Bending angle per dipole	Θ	5.099998	5.099988	mrad
Magnetic length of each aperture	lm	14.34	14.30	m
Radius of curvature	r	2812.36	2803.93	m
Separation of tube centers	d	194.52	194.00	mm
Sagitta	s	9.14	9.12	mm

The sagitta is used to evaluate the quality of the geometry. The sagitta change with respect to the nominal sagitta is estimated by using the fact that the difference between two arcs (the nominal and the measured) can be expressed as a second order polynomial. In figure 2 the nominal sagitta is represented as the horizontal reference axis.

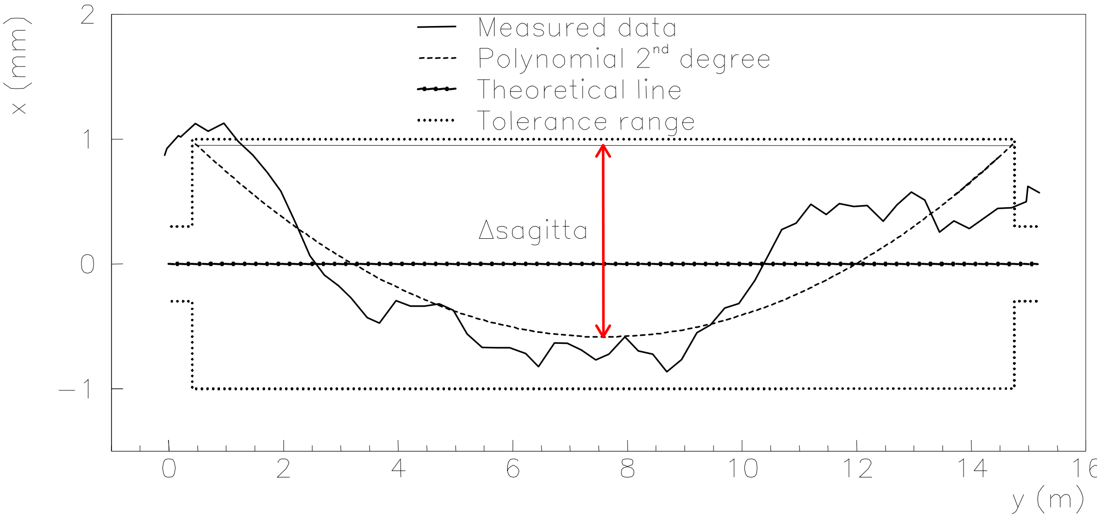

Figure 2: Sagitta calculation

The correctness of the shape can be controlled by acting on the sagitta between different measurements enabled by the degree of freedom of the central foot (figure 3). The central support is blocked to avoid lateral excursions but small corrections of the shape can be made by adjusting the lateral position of the magnet at the position of the central support.

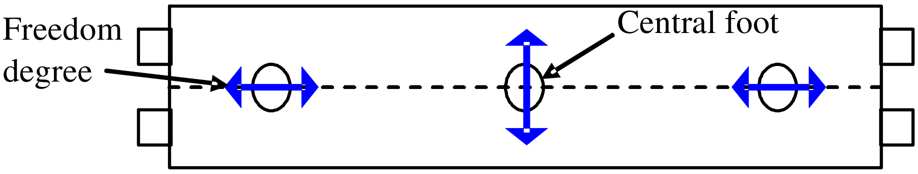

Figure 3: possible movements allowed by the cold mass supports in the cryostat

The change of sagitta has an impact on the position of the flanges (used for interconnection of the magnets) and on the position of the corrector (MCS & MCDO in figure 1). The corrector positions are not accessible after closing the cold mass so their position relative to the geometric mean plane is calculated using the fact that the corrector should move rigidly with the cold mass end cover since the corrector is welded to the cold mass end plate. Similarly the flange should move rigidly with the end cover. The standard deviation of the observed difference in movements of the flange and the end cover is around 0.1 mm.

Beam requirements

The aperture tolerance (figure 4) needed for the magnet is expressed in form of a race-track around theoretical position of the cold bore tube center (green in figure 4). The center of te cold bore tube should not exceed the race track shaped tolerances at ant position along the magnet. Some magnets with tighter tolerances are needed for some critical positions in the machine (rectangle). Some magnets may be placed in non critical positions in the accelerator positions and have relaxed tolerances (largest race-track, magenta). The sagitta control is important to be able to optimize the aperture for the beam. We use the central support to adjust the magnets. The change of position of the central support is directly related to the change in sagitta. By changing the sagitta we can optimize the position of the center of the cold bore tube to avoid too large excursions from the theoretical center.

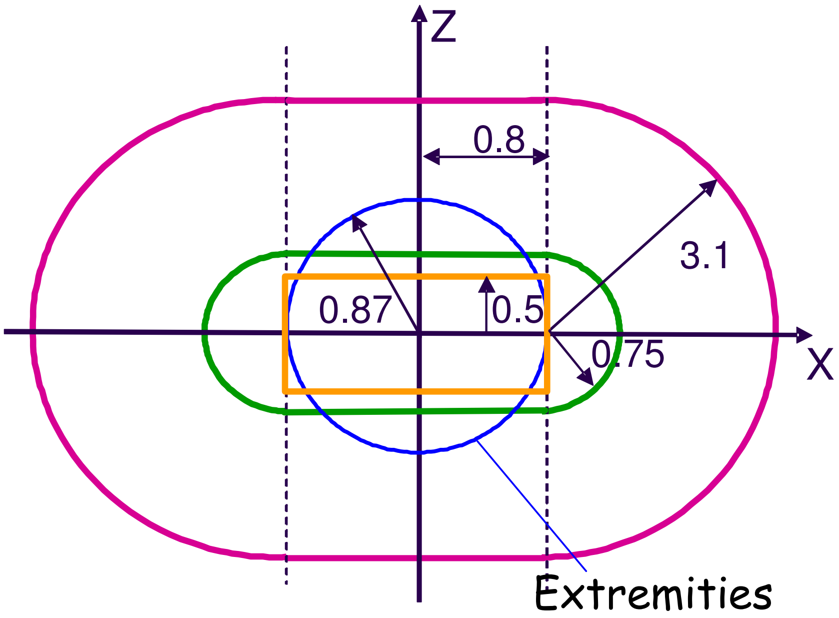

Figure 4: Aperture tolerance

■ Context

Measurements at different steps of dipoles show an intrinsic instability showing up as a tendency to increase their curvature (sagitta increasing) with time. The spread of the positions of the ends also increases which implies a change in the position of the correctors magnets. With the required tolerances these magnets need care to be accommodated in the LHC machine. There is also evidence that some magnets show unexpected shape changes.

Using the data collected at different stages of assembly of the magnet did not afford to predict the behavior of the magnet and no apparent reason in the assembly procedure could explain this behavior.

The magnet supports are designed to allow the cold mass to slide freely along the y-axis in the horizontal plane at the ends and horizontally in the middle (figure 3). To avoid unwanted changes of the magnet shape, it was decided to adjust and block the central foot as to give the “industry shape” to the magnet before installation in the tunnel.

Due to time constraints we had to set up in a short time derived data, like sagitta and aperture qulification, which are results of computations from measurements stored in an Oracle database.

Mathematica allows powerful computations and could be used from Java with JLink, and Java connect Oracle database (via JDBC) in a very easy way. That’s why we decided to use this efficient mix of technologies for our needs.

N.B: Java was used for database manipulation because we were working with *Mathematica* 5.0. Under later versions of *Mathematica* we could connect to Oracle directly from *Mathematica* with the DatabaseLink` package.

■ Implementation

The java program act as "master": open (and close) database connection and link to *Mathematica* via JLink, deal with data exchange to the database and to *Mathematica*.

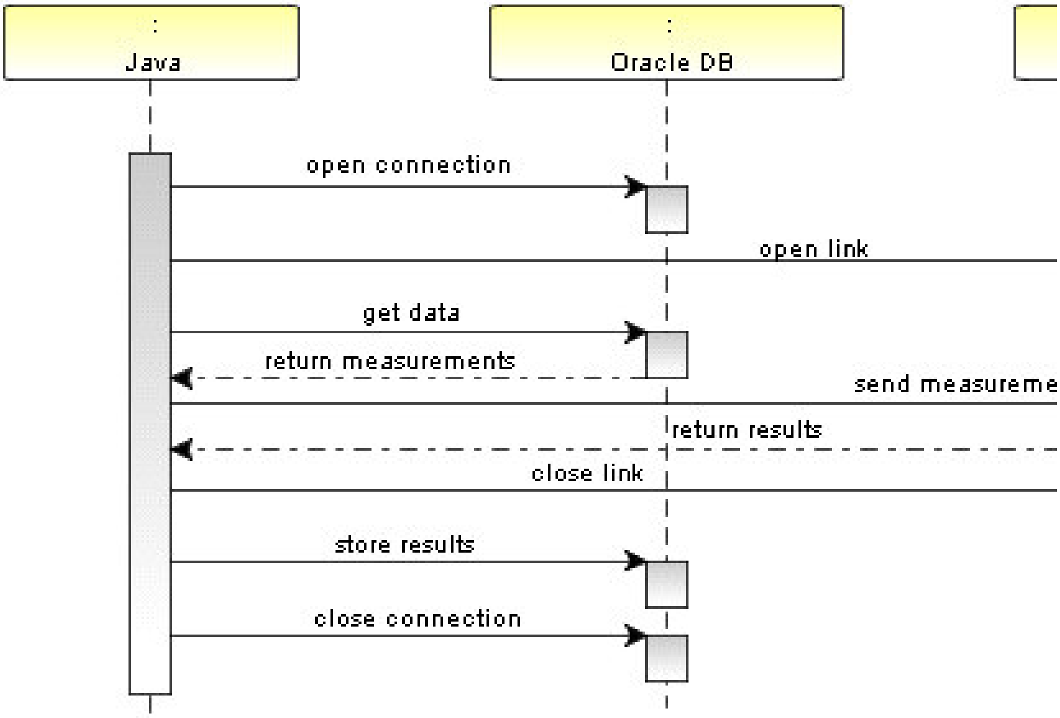

Figure 5: sequence diagram

We first connect to the database :

```
DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
Connection   conn =
DriverManager.getConnection("jdbc:oracle:thin:@(DESCRIPTION=(FAIL
OVER=on)(LOAD_BALANCE=off)(ADDRESS_LIST=(ADDRESS=(PROTOC
OL=TCP)(HOST=orapdb2.cern.ch)(PORT=1521))(ADDRESS=(PROTOCO
L=TCP)(HOST=orapdb1.cern.ch)(PORT=1521)))(CONNECT_DATA=(SER
⋮
VICE_NAME=PDB.WORLD)(FAILOVER_MODE=(TYPE=SELECT)(METHOD
=BASIC))))","user","pwd");
conn.setAutoCommit(false);
```

Then, the link to mathematica is created and the package loaded :

```
mLink = MathLinkFactory.createKernelLink("-linkmode launch -lin⋮
kname 'c:\\program files\\wolfram
research\\mathematica\\5.0\\mathkernel'");
mLink.discardAnswer();
mLink.evaluate("<<OwnPackages`DerivedData`");
mLink.discardAnswer();
```

Measurement are retrieved from the DB and put in the appropriate objects to be passed to Mathematica:

```
Statement query = connect.createStatement();
ResultSet rsetShape;
rsetShape = query.executeQuery ("select Y,DX,DZ,Aperture from
CRYOGEO_AXIS  WHERE MAGNET_NU=" + MagnetNumber + " AND
step='" + Step1 + "' and final_meas='T' ORDER BY APERTURE,Y");

i=0;
j=0;
while (rsetShape.next())
{
 if (rsetShape.getInt("APERTURE")==1) {
   ShapeAp1[0][i]=(Double)Yap1.elementAt(i);
   ShapeAp1[1][i]=(Double)DXap1.elementAt(i);
   ShapeAp1[2][i]=(Double)DZap1.elementAt(i);
   i++;
 }
 else {
   ShapeAp2[0][j]=(Double)Yap2.elementAt(i);
   ShapeAp2[1][j]=(Double)DXap2.elementAt(i);
   ShapeAp2[2][j]=(Double)DZap2.elementAt(i);
   j++;
 }
}
```

Then the Mathematica package is called and results :

```
mLink.activate();

mLink.putFunction("EvaluatePacket",1);
mLink.putFunction("RaceTrackAp",1);
mLink.put(ShapeAp1);
mLink.endPacket();
mLink.waitForAnswer();
RaceTrack[0]=mLink.getDoubleArray1();

mLink.putFunction("EvaluatePacket",1);
mLink.putFunction("RaceTrackAp",1);
mLink.put(ShapeAp2);
mLink.endPacket();
mLink.waitForAnswer();
RaceTrack[1]=mLink.getDoubleArray1();
```

Finally results are loaded into Oracle :

```
PreparedStatement InsertPstmt = conn.prepareStatement("Insert into
CRYOGEO_RACETRACK(Magnet_Nu,Step,Reference_Meas,Magnet_Nam
e,Stage,Aperture,Y,RTrack) values(?,?,?,?,?,?,?,?)");

InsertPstmt.setInt(1,Integer.parseInt(AdminData[0]));
InsertPstmt.setString(2,AdminData[1]);
InsertPstmt.setInt(3,Integer.parseInt(AdminData[2]));
InsertPstmt.setString(4,AdminData[3]);
InsertPstmt.setString(5,AdminData[4]);

for(i=0;i<ShapeAp1[0].length;i++){
  InsertPstmt.setInt(6,1);
```

```
InsertPstmt.setDouble(7,Double.parseDouble(ShapeAp1[0][i].toString(
)));
  InsertPstmt.setDouble(8,RaceTrack[0][i+1]);
  InsertPstmt.executeUpdate();
}//for

for(i=0;i<ShapeAp2[0].length;i++){
  InsertPstmt.setInt(6,2);

InsertPstmt.setDouble(7,Double.parseDouble(ShapeAp2[0][i].toString(
)));
  InsertPstmt.setDouble(8,RaceTrack[1][i+1]);
  InsertPstmt.executeUpdate();
}//for
InsertPstmt.close();
conn.commit();
conn.close();
```

A subset of the Mathematica package :

```
BeginPackage["OwnPackages`DerivedData`"]

DSagitta::usage =
 "returns the delta sagitta computed on the bent part of
   the dipole (14.343m), the sagitta residue which is the
   difference between maximum DX computed on the fit of
   order 2  and half of delta sagitta. It also returns and
   the fitting error with an order 2 polynomial fitting."

Begin["`Private`"]

DSagitta[ShapeAp_List] :=
 Module[{DeltaSagitta, SagittaResidue, FitErrorO2,
   iFirst, iLast, ShapeToFit, Fito2, MaxDX},
  iFirst = Max[Catch[Do[If[ShapeAp[[1, i]] > 0.4075,
       Throw[i]], {i, Length[ShapeAp[[1]]]}]], 1];
  iLast = If[ShapeAp[[1, Length[ShapeAp[[1]]]]] < 14.7505,
    Length[ShapeAp[[1]]],
    Catch[Do[If[ShapeAp[[1, i]] > 14.7505, Throw[i]],
       {i, Length[ShapeAp[[1]]]}]] - 1];
  ShapeToFit = {Take[ShapeAp[[1]], {iFirst, iLast}],
    Take[ShapeAp[[2]], {iFirst, iLast}]};
  Fito2 = Fit[Transpose[ShapeToFit],
    Table[x^i, {i, 0, 2}], x];
  MaxDX = Mean[{Fito2 /. x → 0.4075, Fito2 /. x → 14.7505}];
  DeltaSagitta = MaxDX - (Fito2 /. x → 7.579);
  SagittaResidue = MaxDX - 0.5 * DeltaSagitta;
  DataFito2 = Fito2 /. x → ShapeToFit[[1]];
  FitErrorO2 =
   Sqrt[(Plus @@ ((ShapeToFit[[2]] - DataFito2) ^2)) /
     Length[ShapeToFit[[1]]]];
  {DeltaSagitta, SagittaResidue, FitErrorO2}
 ]

End[]

EndPackage[]
```

The java program is run several times every day as a scheduled task.

■ Conclusion

For non trivial calculations of derived entities in the data base, we needed an easy way to interface with a high level language. *Mathematica* was well adapted. The *Mathematica* routines may easily be modified by the users without changing the access software. This gives high flexibility. The rapid implementation of the system was essential for the evaluation and the success of the adjustment procedures of the LHC dipole.

About the Authors

J. Beauquis
CERN AT-MAS
1211 Geneva 23
Switzerland
jerome.beauquis@cern.ch

J. Beauquis has a fellowship at CERN, Geneva, and is responsible for the databases storing warm magnetic field and geometry measurements performed on superconducting magnets for the Large Hadron Collider. He holds a Masters degree in computer science and applied mathematics from the university of Grenoble.

Elena Wildner
CERN
1211 Geneva 23
Elena.Wildner@cern.ch

E.Wildner is a staff senior scientist at CERN, Geneva, with experience in control system design, accelerator operation, accelerator physics, databases and magnet design and analysis. She holds a Masters degree in physics and technology of Chalmers University of Technology in Gothenburg and a Swedish "Technology Licientiate" (physics) of the Swedish Royal Institue of Technology in Stockholm. Between 1973 and 1981 (when she joined CERN), she worked for ASEA, Sweden, (now ASEA/Brown Boveri) with refrigeration systems and control systems.