Efficient Field Computation for Muon Magnets

Martin Berz and Kyoko Makino

Department of Physics and Astronomy
Michigan State University

Department of Physics and Astronomy
University of Illinois at Urbana-Champaign

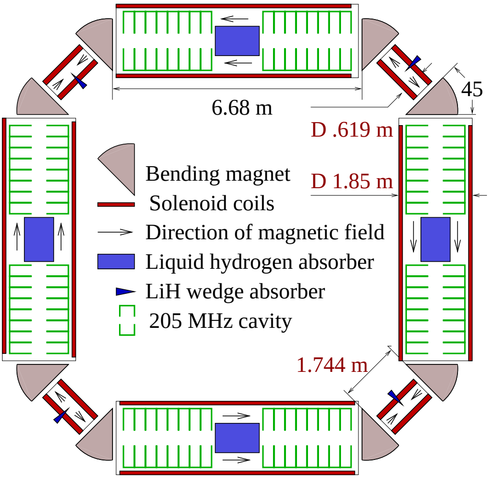
Problems for Muon Dynamics Simulations

- **Large phase space** has to be transported - requires careful treatment of nonlinear effects
- Elements have **large aperture** compared to their length, so have significant fringe fields
- **Fringe fields** for many of the elements are absolutely non-negligible

However, field computations are **difficult:**

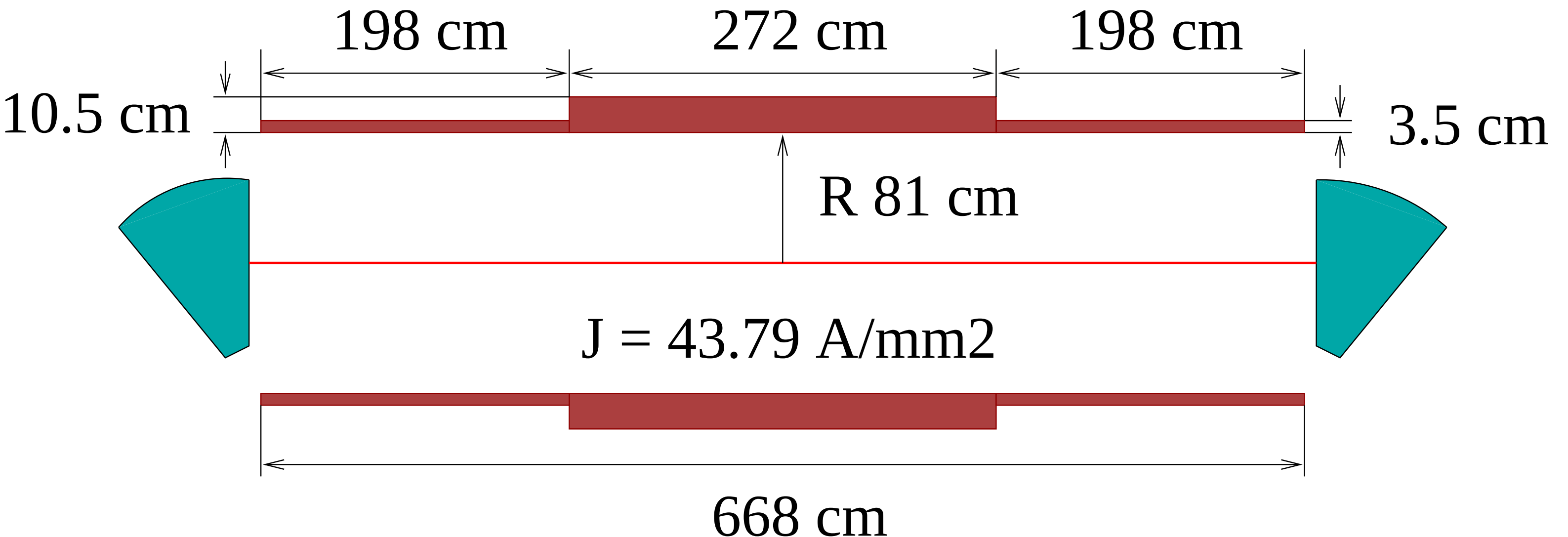
- Often field models are assumed **only on axis or in midplane** - how do we get the field in space?
- Fields are often represented as **large sums of field contributions** of current loops, current sheets, block solenoids etc
- Off-axis form of these field formulas involve **elliptic integrals** or other complicated expressions that have to be approximated somehow. On-axis or midplane fields much simpler.

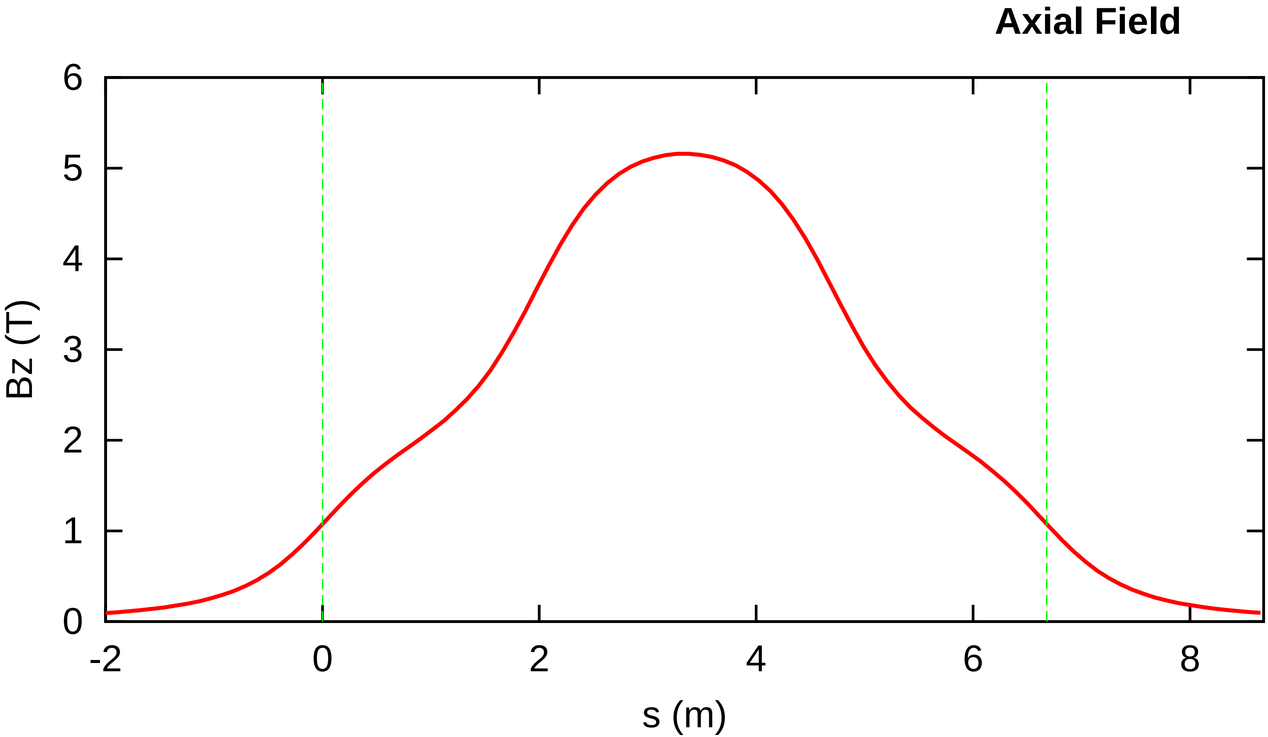
Thus, for particle tracking, **significant effort** necessary to just compute the field that a particle sees.



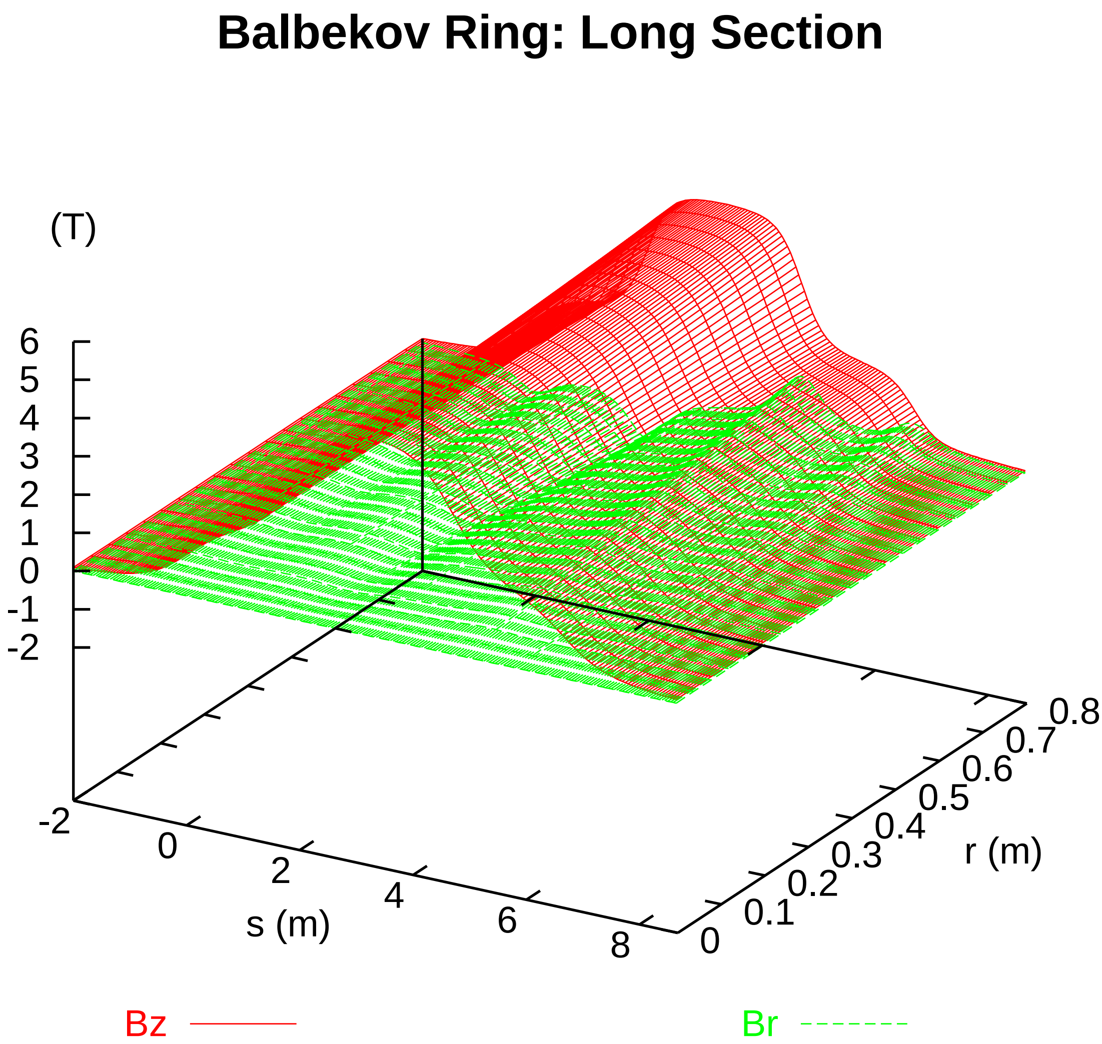
Circumference	36.963 m
Nominal energy at short SS and bends	250 MeV
Bending field	1.453 T
Norm. field gradient	0.5
Max. solenoid field	5.155 T
RF frequency	205.69 MHz
Accelerating gradient	15 MeV/m
LH_2 absorber length	128 cm
LiH wedge absorber	14 cm
Grad. of energy loss	0.75 MeV/cm

Figure 1: Layout and parameters of the solenoid based ring cooler.

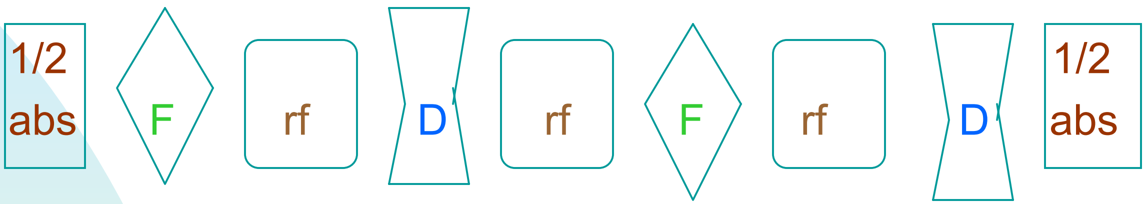

198 cm
272 cm
198 cm
10.5 cm
3.5 cm
R 81 cm
J = 43.79 A/mm2
668 cm



Axial Field
Bz (T)
0
1
2
3
4
5
6
-2
0
2
4
6
8
s (m)


Balbekov Ring: Long Section
(T)
6
5
4
3
2
1
0
-1
-2
-2
0
2
4
6
8
s (m)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
r (m)
Bz
Br

Construction of FODO Quad Cooling Cell



COOLING CELL PHYSICAL PARAMETERS:

Quad Length	0.6 m
Quad bore	0.6 m
Poletip Field	~1 T
Interquad space	0.4 - 0.5 m
Absorber length	0.35 m *
RF cavity length	0.4 - 0.7 m*
Total cooling cell length	4 m

*The absorber and the rf cavity can be made longer if allowed to extend into the ends of the magnets.

Or, more rf can be added by inserting another FODO cell between absorbers

In this design

For applications further upstream at larger emittances, this channel can support a 0.8 m bore, 0.8 m long quadrupole with no intervening drift without matching to the channel described here.

Local Multipole Expansion

Instead of computing fields for each particle, rather

- Compute **one** local field expansion for a particular value of arclength s
- Apply this field expansion to **all** particles (mere polynomial operation)
- Can also use this information for compuation of transfer map (requires multipole fields)

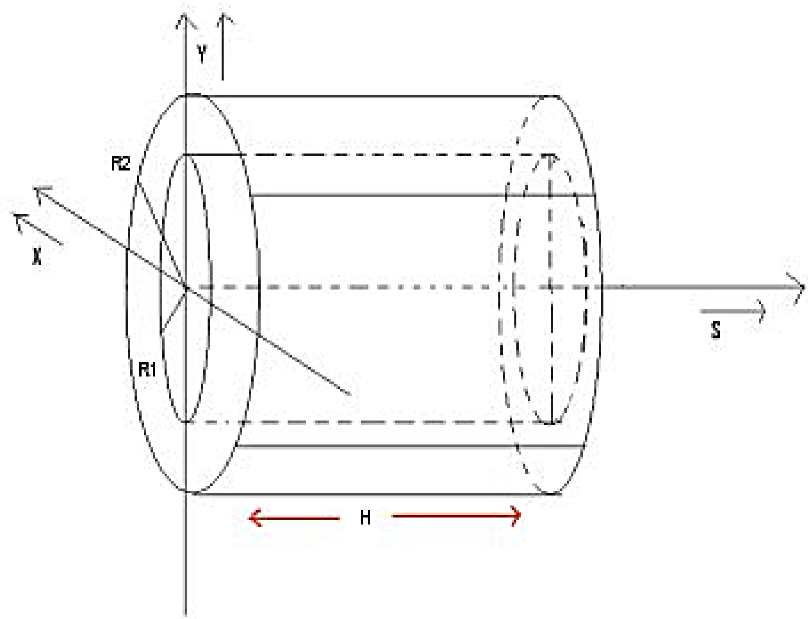
Advantages of this approach

- Significantly **faster**: field evaluation is only polynomial evaluation
- Particle pushing effort independent of complexity of the element
- Need field formulas **only in midplane or on axis** (they are much simpler there)
- Field is automatically order-by-order **Maxwellian**
- Motion is automatically order-by-order **Symplectic** (if integrated with symplectic integrator)

The formulae for the Magnetic fiel on the axis is

$$B\ (s) = 2\,i\,k\,\pi \left(S\,\mathrm{Log}\left[\frac{R_2 + \sqrt{S^2 + R_2^2}}{R_1 + \sqrt{S^2 + R_1^2}}\right] + (H - S)\ \mathrm{Log}\left[\frac{R_2 + \sqrt{(H - S)^2 + R_2^2}}{R_1 + \sqrt{(H - S)^2 + R_1^2}}\right] \right)$$

k is equal to $\frac{\mu_0}{4 * Pi}$, and 'i' is the current



Transfer Map Method and Differential Algebras

- The transfer map $\mathcal{M}$ is the flow of the system ODE.

$$\vec{z}_f = \mathcal{M}(\vec{z}_i, \vec{\delta}),$$

where $\vec{z}_i$ and $\vec{z}_f$ are the initial and the final condition, $\vec{\delta}$ is system parameters.

- For a repetitive system, only one cell transfer map has to be computed. Thus, it is much faster than ray tracing codes (i.e. tracing each individual particle through the system).
- The Differential Algebraic method allows a very efficient computation of high order Taylor transfer maps.
- The Normal Form method can be used for analysis of nonlinear behavior.

Differential Algebras (DA)

- it works to arbitrary order, and can keep system parameters in maps.
- very transparent algorithms; effort independent of computation order.

The code **COSY Infinity** has many tools and algorithms necessary.

Field Description in Differential Algebra

There are various DA algorithms to treat the fields of beam optics efficiently. For example, **DA PDE Solver**

- requires to supply only
  - the midplane field for a midplane symmetric element.
  - the on-axis potential for straight elements like solenoids, quadrupoles, and higher multipoles.
- treats arbitrary fields straightforwardly.
  - Magnet (or, Electrostatic) fringe fields:
    The Enge function fall-off model

    $$F(s) = \frac{1}{1 + \exp(a_1 + a_2 \cdot (s/D) + ... + a_6 \cdot (s/D)^5)}$$

    where D is the full aperture.
    Or, any arbitrary model including the measured data representation.
  - Solenoid fields including the fringe fields.
  - Measured fields: E.g. use Gaussian wavelet representation.
  - Etc. etc.

DA Fixed Point PDE Solvers

The **DA fixed point theorem** allows to **solve PDEs iteratively** in **finitely many steps** by rephrasing them in terms of a fixed point problem.

Consider the rather general PDE

$$a_1 \frac{\partial}{\partial x}\left(a_2 \frac{\partial}{\partial x} V\right) + b_1 \frac{\partial}{\partial y}\left(b_2 \frac{\partial}{\partial y} V\right) + c_1 \frac{\partial}{\partial z}\left(c_2 \frac{\partial}{\partial z} V\right) = 0,$$

where a_i, b_i, c_i are functions of x, y, z.

The PDE is re-written in **fixed point form** as

$$\begin{aligned} V = V|_{y=0} &+ \int_0^y \frac{1}{b_2}\left(b_2 \frac{\partial V}{\partial y}\right)\bigg|_{y=0} \\ &- \int_0^y \frac{1}{b_2} \int_0^y \left(\frac{a_1}{b_1}\frac{\partial}{\partial x}\left(a_2 \frac{\partial V}{\partial x}\right) + \frac{c_1}{b_1}\frac{\partial}{\partial z}\left(c_2 \frac{\partial V}{\partial z}\right)\right) dy dy. \end{aligned}$$

Assume the derivatives of V and $\partial V/\partial y$ with respect to x and z are **known in the plane** $y = 0$. Then the right hand side is **contracting** with respect to y (which is necessary for the DA fixed point theorem), and the various orders in y can be **iteratively** calculated by mere iteration.

3D Midplane Laplace Solver

Laplace equation in curvilinear coordinates:

$$\triangle V = \frac{1}{1+hx}\frac{\partial}{\partial x}\left\{(1+hx)\frac{\partial V}{\partial x}\right\} + \frac{\partial^2 V}{\partial y^2} + \frac{1}{1+hx}\frac{\partial}{\partial s}\left(\frac{1}{1+hx}\frac{\partial V}{\partial s}\right) = 0.$$

Fixed point form:

$$V = V\big|_{y=0} + \int_0^y \left(\frac{\partial V}{\partial y}\right)\bigg|_{y=0} dy$$
$$-\int_0^y \int_0^y \left[\frac{1}{1+hx}\frac{\partial}{\partial x}\left\{(1+hx)\frac{\partial V}{\partial x}\right\} + \frac{1}{1+hx}\frac{\partial}{\partial s}\left(\frac{1}{1+hx}\frac{\partial V}{\partial s}\right)\right] dydy.$$

For whatever V, all parts not depending on y are reproduced exactly (midplane info is preserved)

The advantages of the method are:

- One needs code for the field only for the midplane.
- The resulting field will always satisfy Maxwell's equations.
- It works to any order.

3D Axis Laplace Solver

Laplace Equation in cylindrical coordinates:

$$\triangle V = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial V}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 V}{\partial \phi^2} + \frac{\partial^2 V}{\partial s^2} = 0.$$

If V does not depend on ϕ, namely V is rotationally symmetric, as in solenoid magnets, the fixed point form of the Laplace equation is simplified to

$$V = V|_{r=0} - \int_0^r \frac{1}{r}\int_0^r r\,\frac{\partial^2 V}{\partial s^2} dr dr,$$

For whatever V, all parts not depending on r are reproduced exactly (on=axis info is preserved)

The advantages of the method are:

- One needs code for the field only on the axis
- The resulting field will always satisfy Maxwell's equations.
- It works to any order.

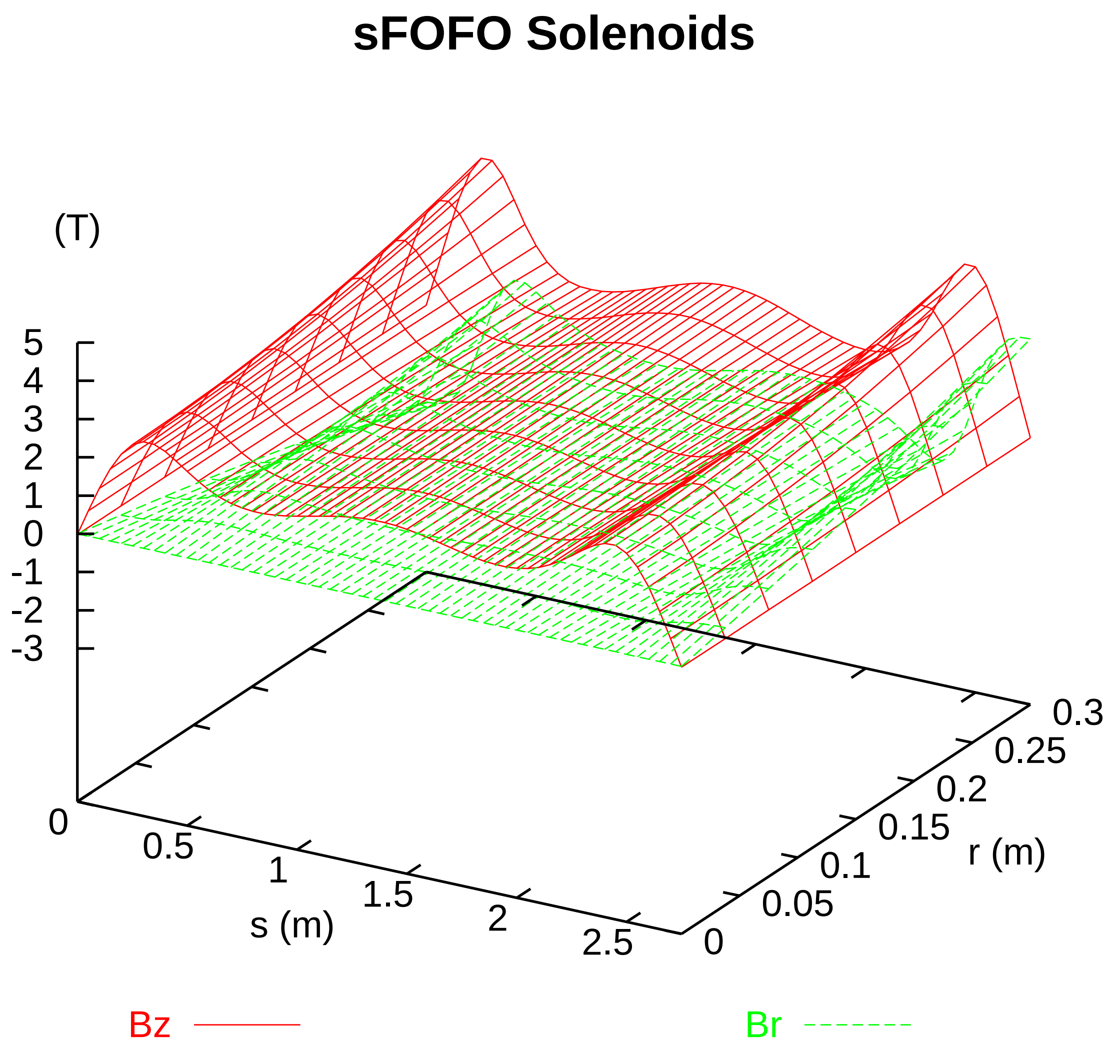
COSY Code for 3D Laplace Solver

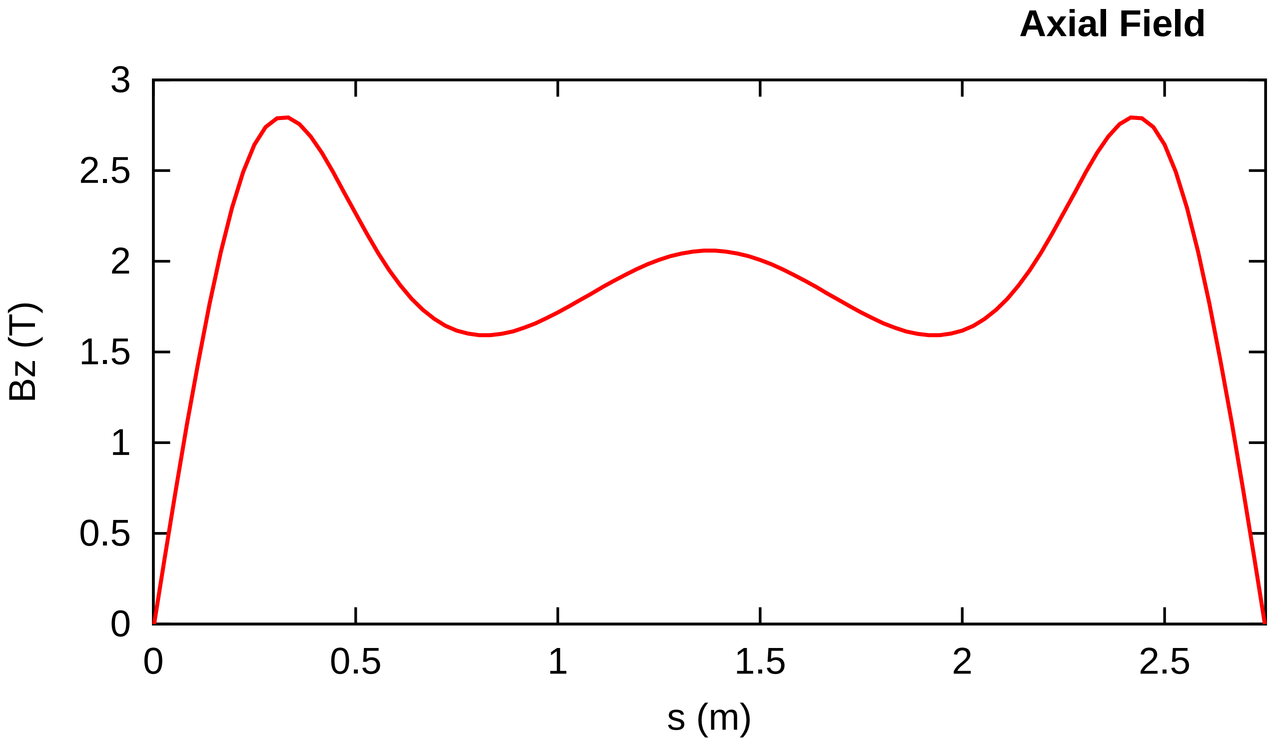
(Midplane symmetry case; rotationally symmetric case similar)

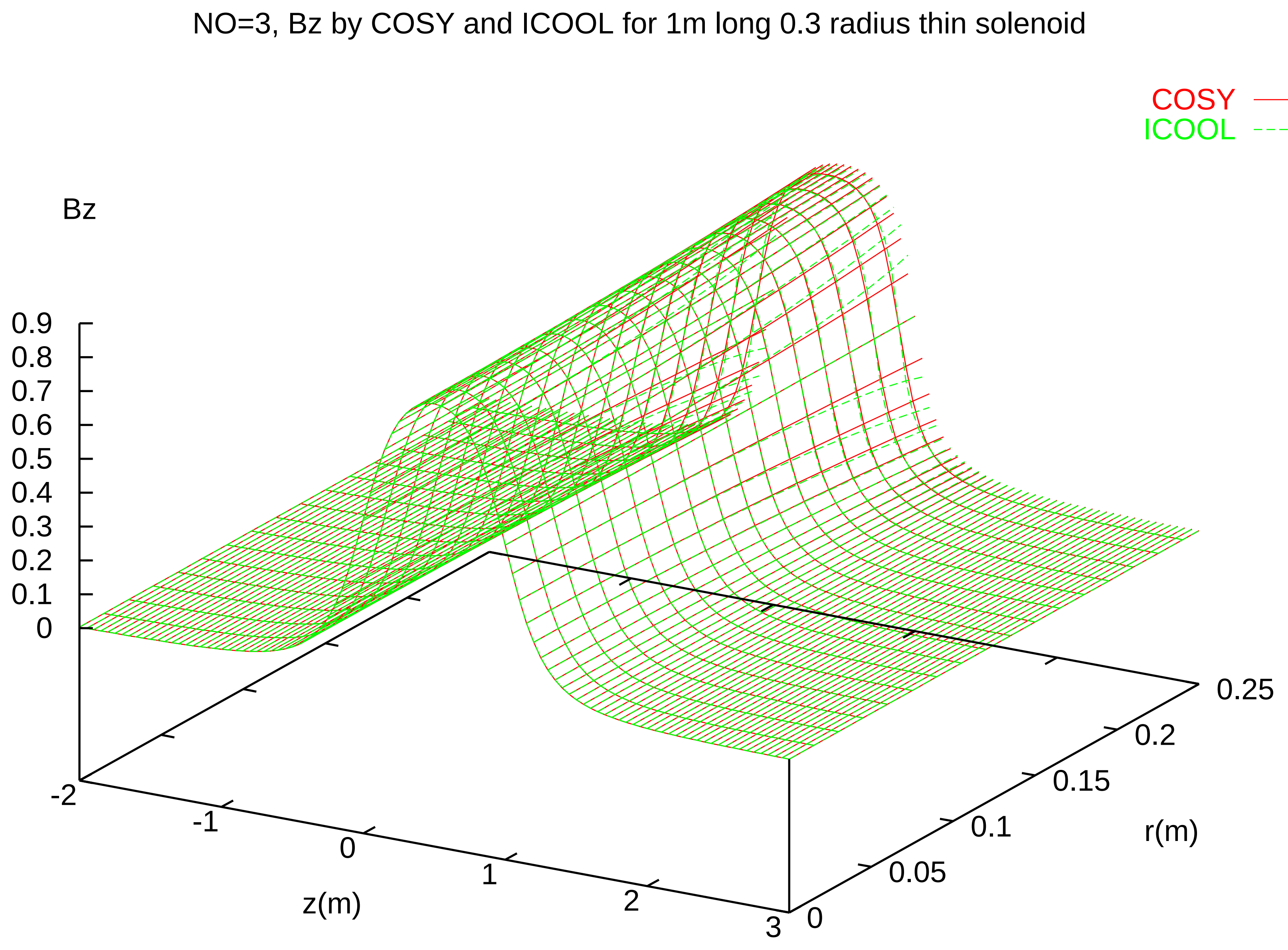
```
HF := 1+H*DA(IX) ; HI := 1/HF ; POLD := P ;
LOOP I 2 NOC+2 2 ;
P := POLD - INTEG(IY,INTEG(IY,
HI*( DER(IX,HF*DER(IX,P)) + DER(IS,HI*DER(IS,P)) ) )) ;
ENDLOOP ;\bigskip
```

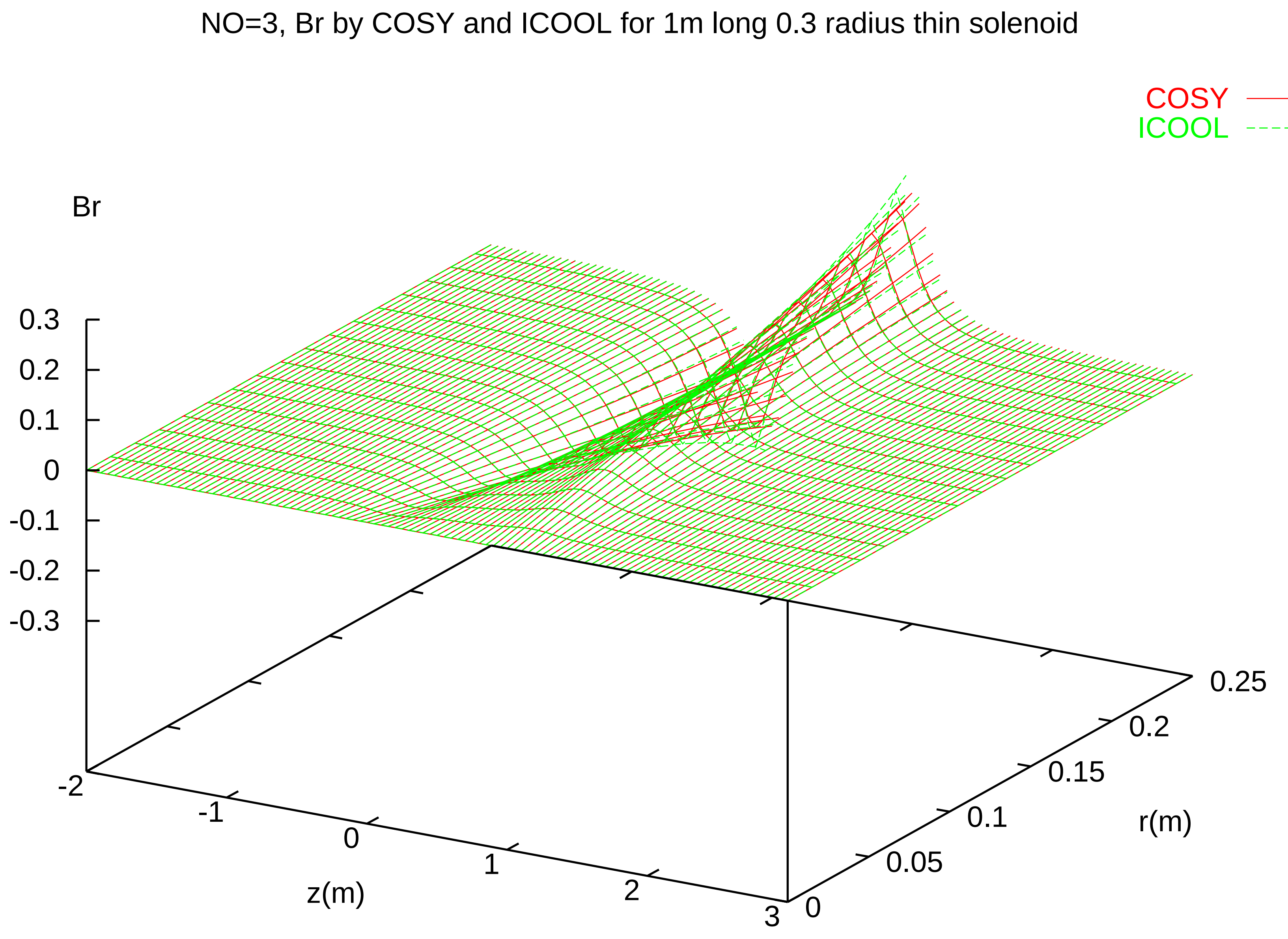
- Variable P is pre-loaded with expansion in midplane only
- Loop over variable I goes to order of interest
- Function INTEG performs integration w.r.t. y
- 3D field is derived from the solution potential `P`, using again the DA technique as

```
BX := DER(IX,P) ;
BY := DER(IY,P) ;
BZ := DER(IS,P) ;
```


sFOFO Solenoids
(T)
5
4
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
s (m)
0
0.05
0.1
0.15
0.2
0.25
0.3
r (m)
Bz
Br

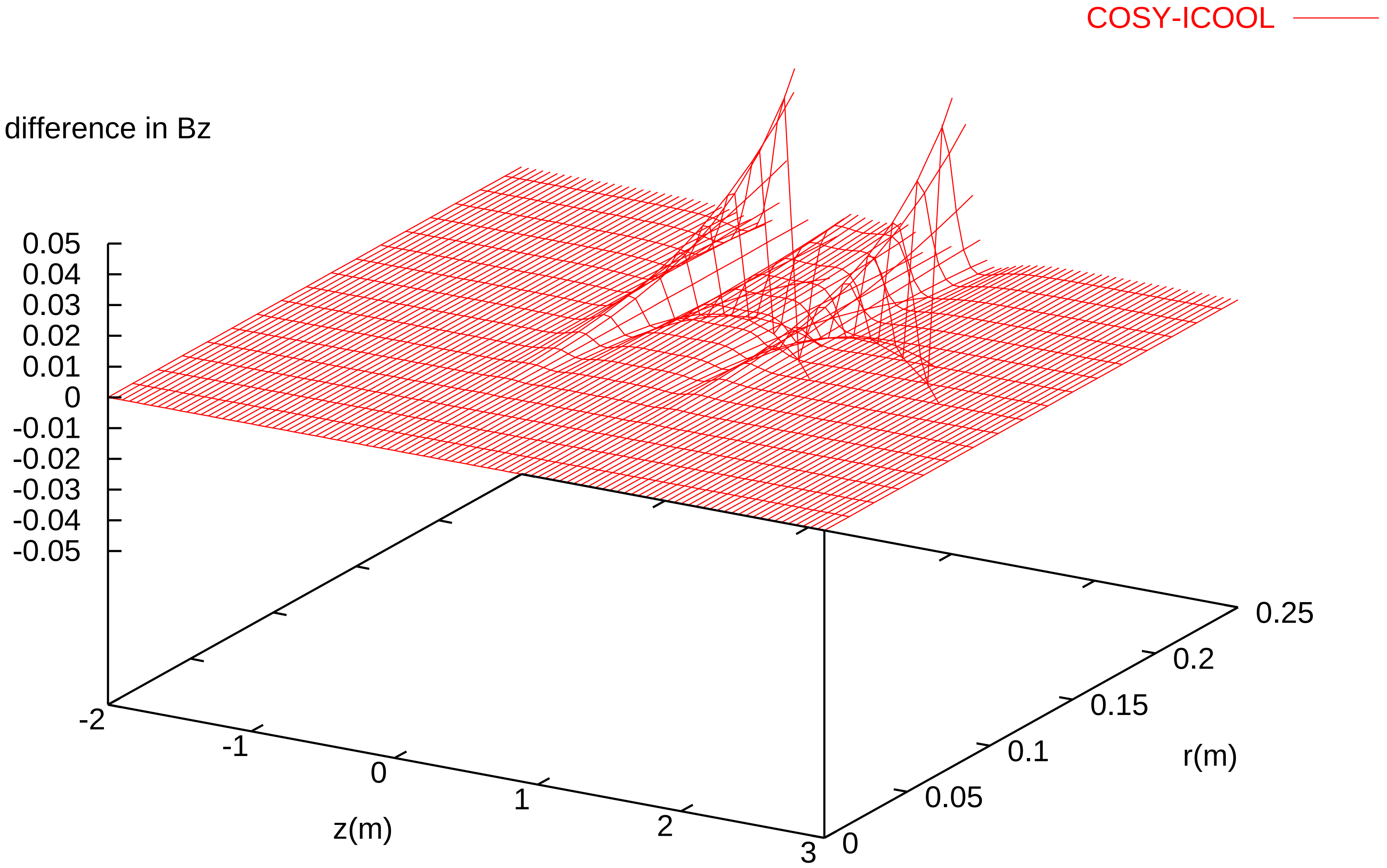

Axial Field
Bz (T)
s (m)
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2
2.5

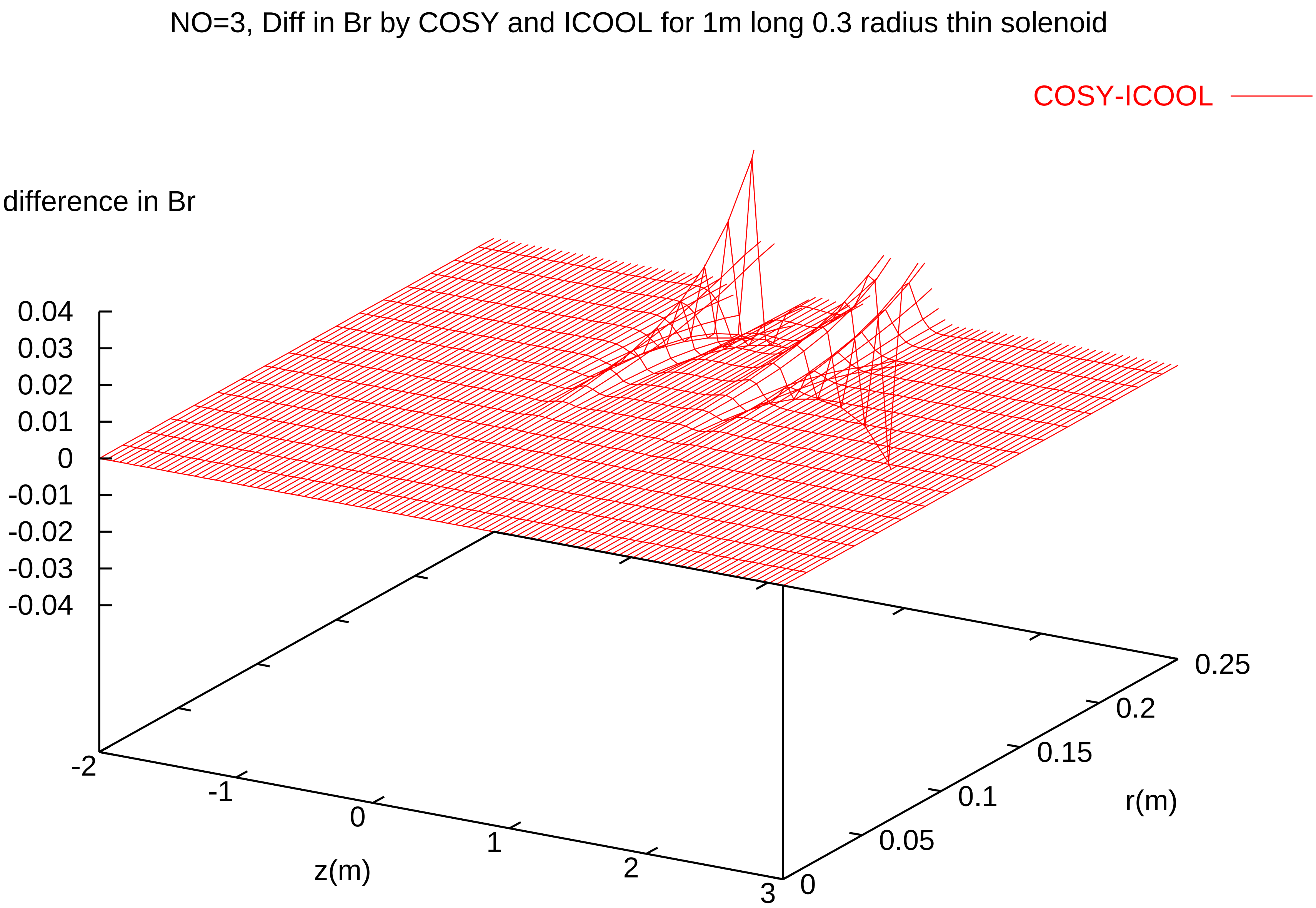

NO=3, Bz by COSY and ICOOL for 1m long 0.3 radius thin solenoid
COSY
ICOOL
Bz
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)



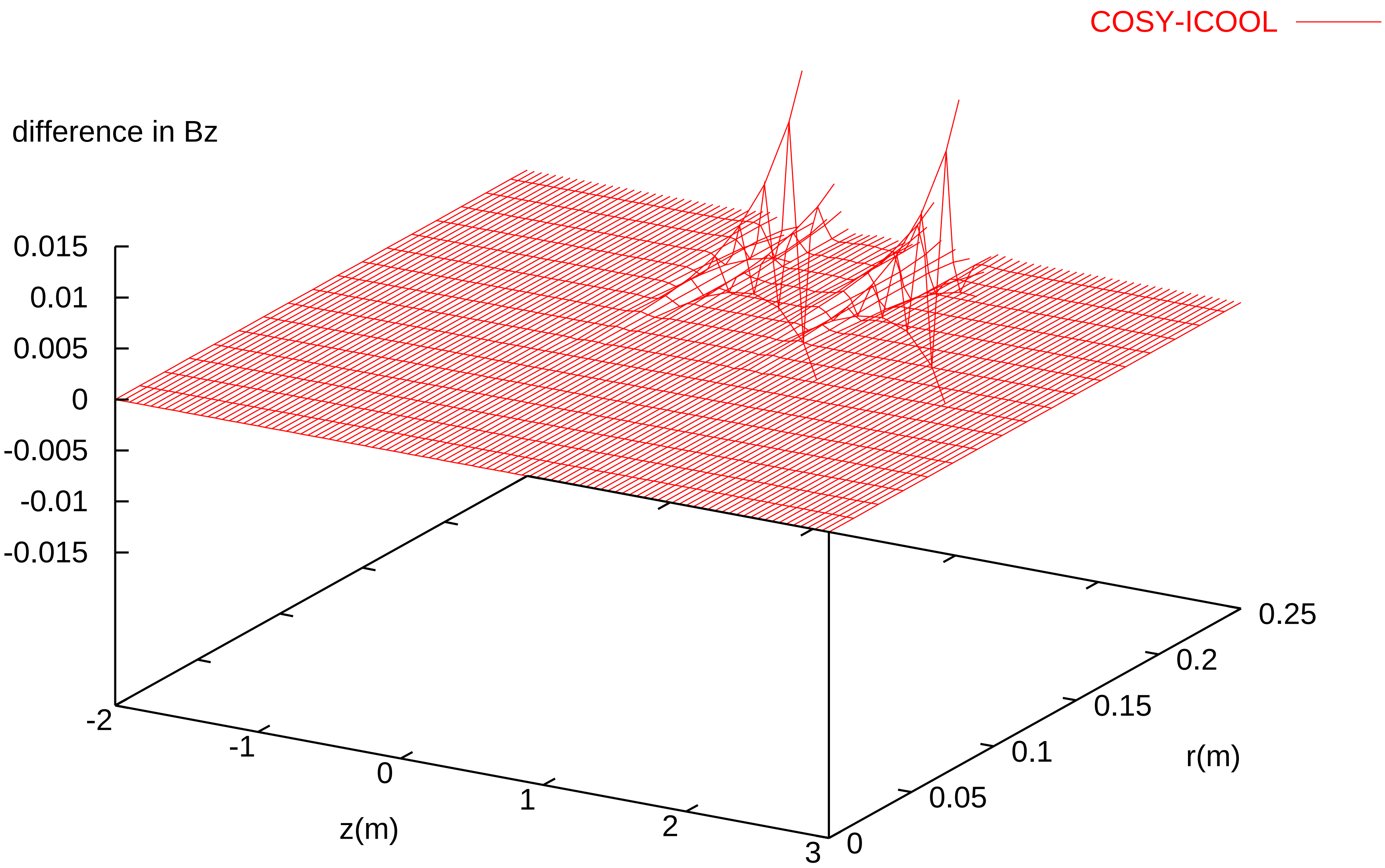
NO=3, Br by COSY and ICOOL for 1m long 0.3 radius thin solenoid
COSY
ICOOL
Br
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)

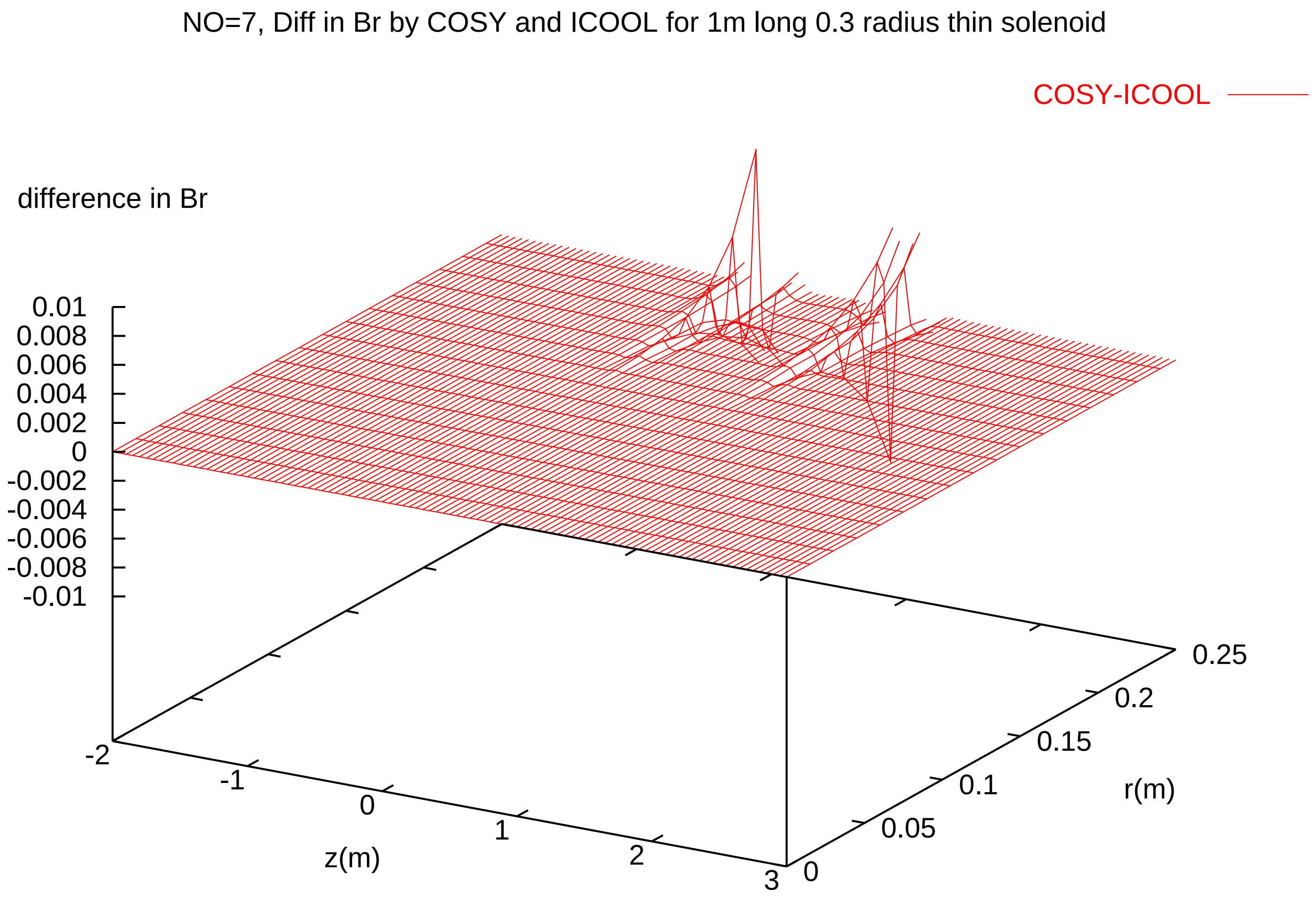
NO=3, Diff in Bz by COSY and ICOOL for 1m long 0.3 radius thin solenoid

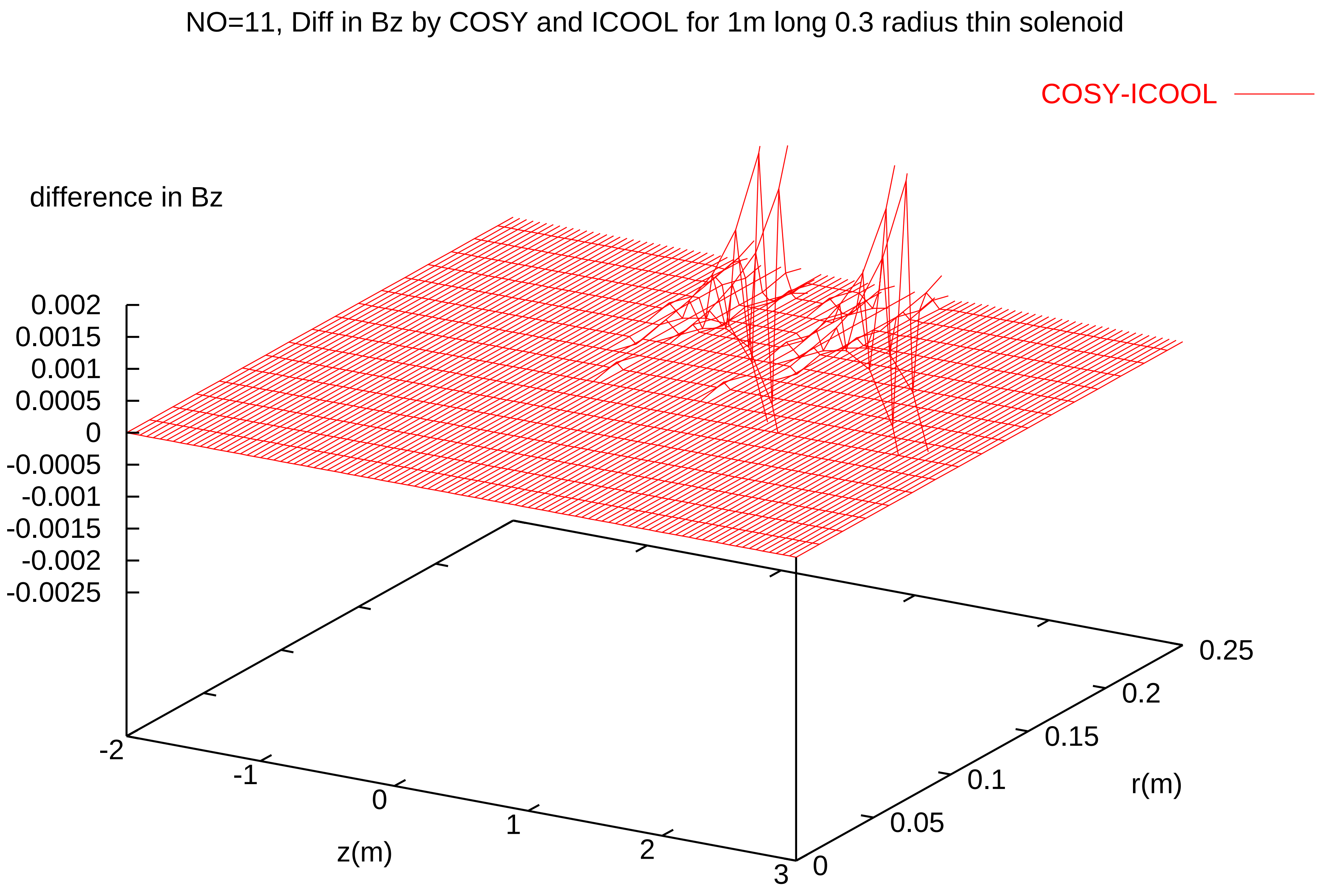



NO=3, Diff in Br by COSY and ICOOL for 1m long 0.3 radius thin solenoid
COSY-ICOOL
difference in Br
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)

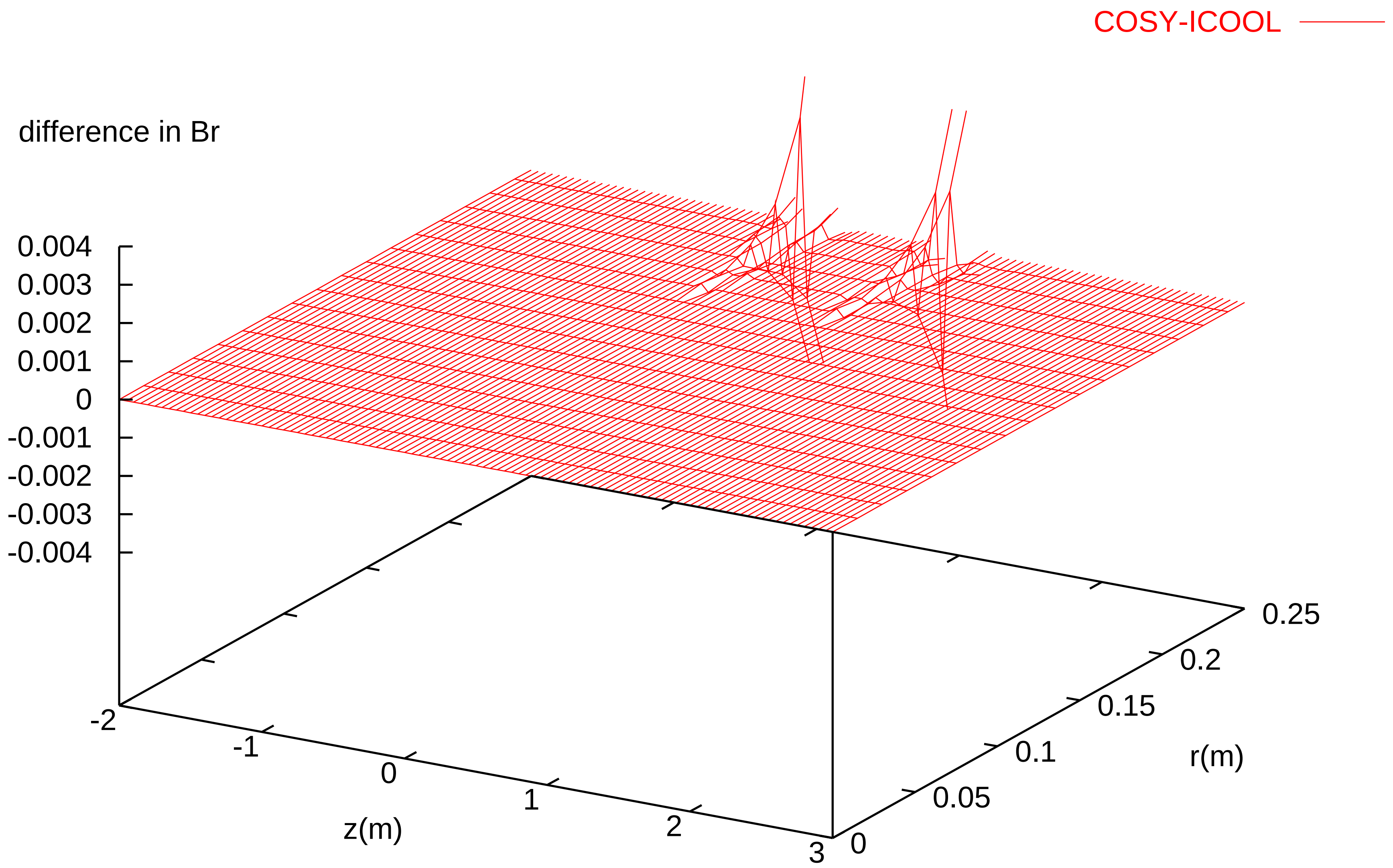
NO=7, Diff in Bz by COSY and ICOOL for 1m long 0.3 radius thin solenoid

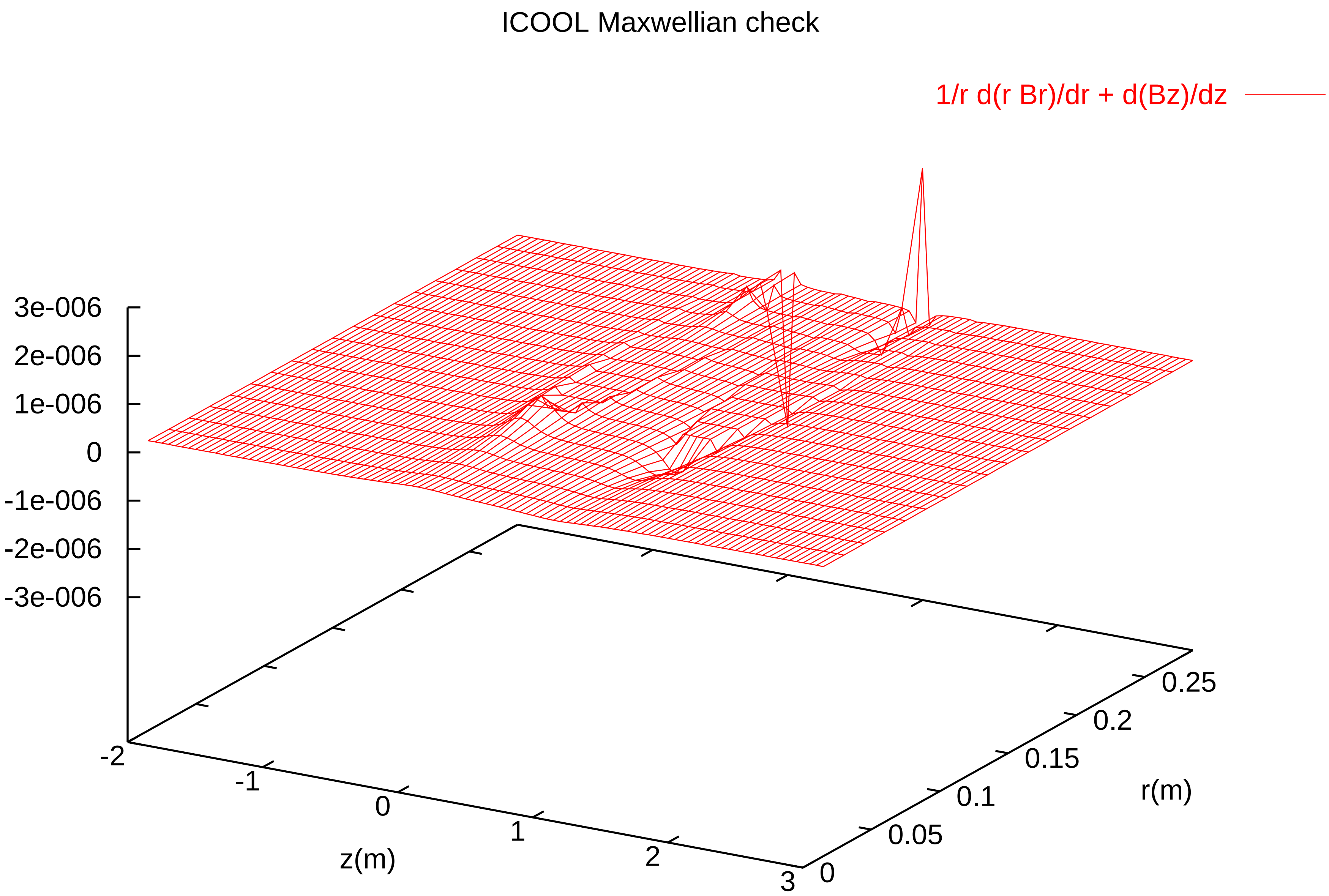


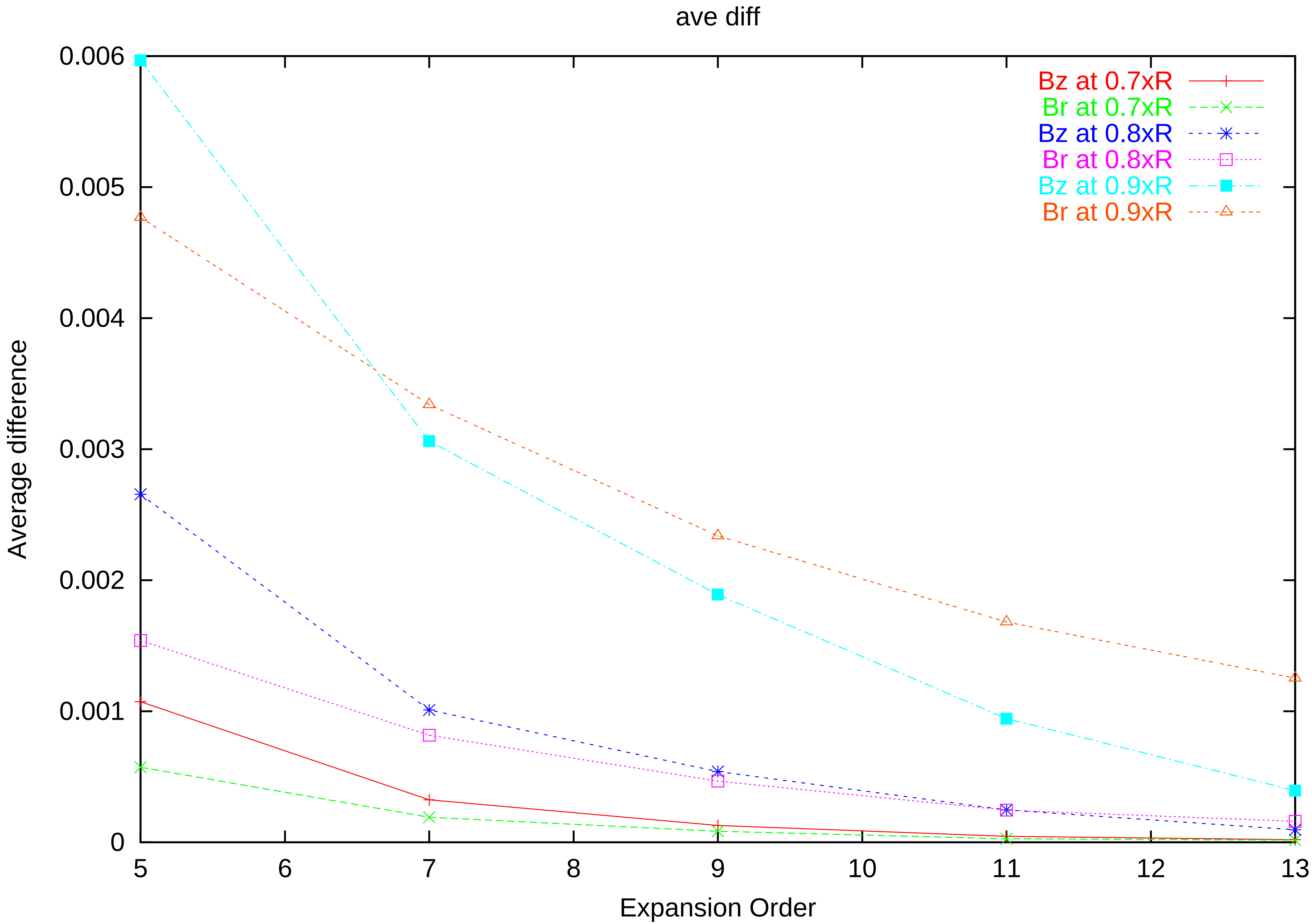

NO=7, Diff in Br by COSY and ICOOL for 1m long 0.3 radius thin solenoid
COSY-ICOOL
difference in Br
0.01
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
-0.01
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)

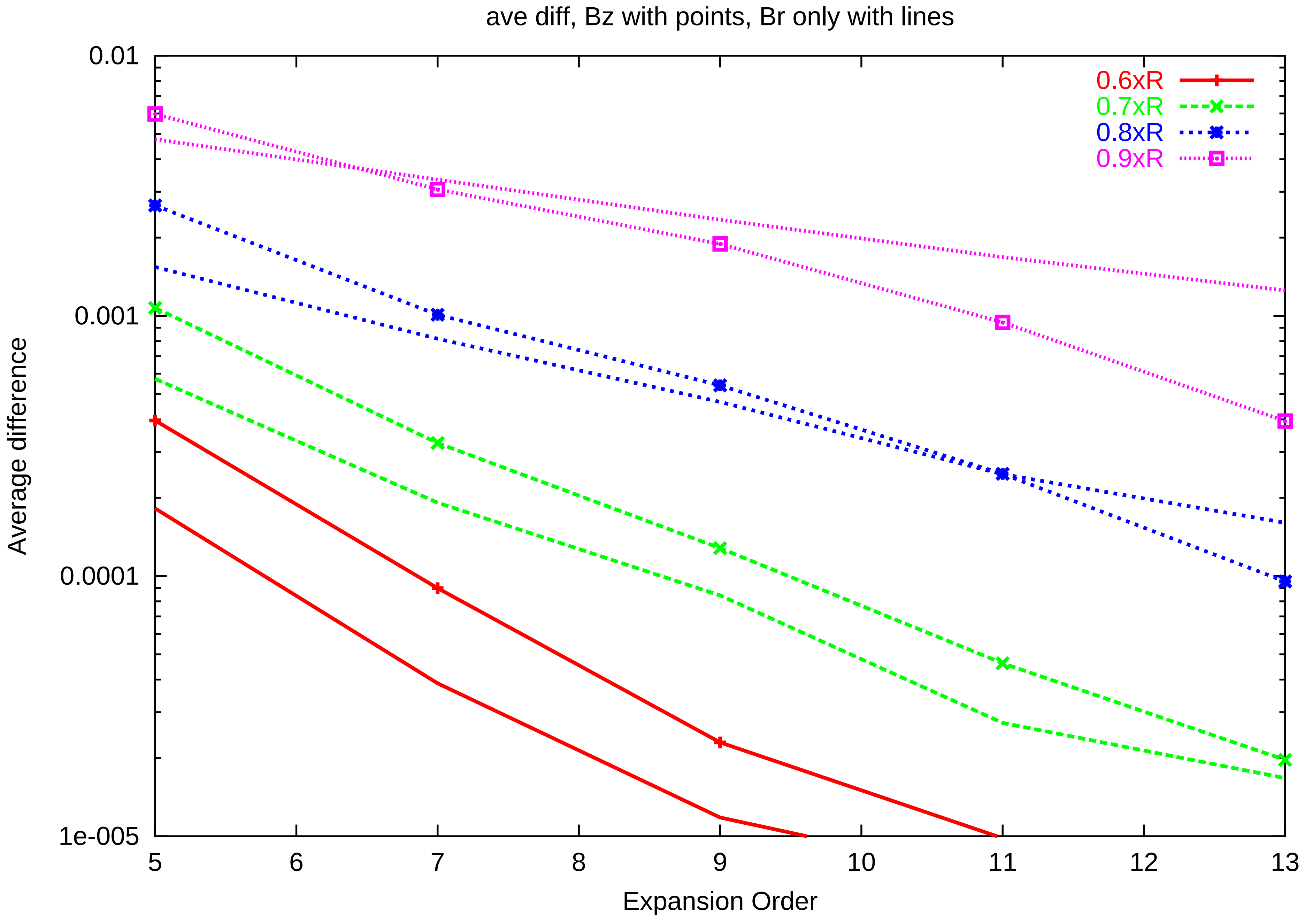

NO=11, Diff in Bz by COSY and ICOOL for 1m long 0.3 radius thin solenoid
COSY-ICOOL
difference in Bz
0.002
0.0015
0.001
0.0005
0
-0.0005
-0.001
-0.0015
-0.002
-0.0025
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)

NO=11, Diff in Br by COSY and ICOOL for 1m long 0.3 radius thin solenoid




ICOOL Maxwellian check
1/r d(r Br)/dr + d(Bz)/dz
3e-006
2e-006
1e-006
0
-1e-006
-2e-006
-3e-006
-2
-1
0
1
2
3
z(m)
0
0.05
0.1
0.15
0.2
0.25
r(m)


ave diff
Average difference
Expansion Order
Bz at 0.7xR
Br at 0.7xR
Bz at 0.8xR
Br at 0.8xR
Bz at 0.9xR
Br at 0.9xR
0
0.001
0.002
0.003
0.004
0.005
0.006
5
6
7
8
9
10
11
12
13



ave diff, Bz with points, Br only with lines
Average difference
0.01
0.001
0.0001
1e-005
5
6
7
8
9
10
11
12
13
Expansion Order
0.6xR
0.7xR
0.8xR
0.9xR

Conclusions

- Allows treatment of model fields based on **simple assumptions** about on-axis or in-plane information only
- Automatically leads to order-by-order **Maxwellian** fields
- Can provide certain **smoothing** if fields are based on discretized pieces
- Resulting multipole expansion provides **insight** for correction and optimization (what orders matter? what elements can be used to correct?)
- If many particles are transported and fields are complicated, computationally significantly **more efficient**.
- **Simpler Midplane or on-axis fields** of current rings, sheets, blocks etc etc than general formulas
- Allows directly the computation of **transfer maps**