CS*T_UG*

Zpravodaj Československého sdružení uživatelů TeXu Zpra sdružení uživatelů TeXu Zpravodaj Československého sdružení uži j Československého sdružení uživatelů TeXu Zpravodaj Českoslove elů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zp ého sdružení uživatelů TeXu Zpravodaj Československého sdružen vodaj Československého sdružení uživatelů TeXu Zpravodaj Česko živatelů TeXu Zpravodaj Československého sdružení uživatelů TE venského sdružení uživatelů TeXu Zpravodaj Československého sd Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Č ení uživatelů TeXu Zpravodaj Československého sdružení uživate skoslovenského sdružení uživatelů TeXu Zpravodaj Československ TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpra sdružení uživatelů TeXu Zpravodaj Československého sdružení uži j Československého sdružení uživatelů TeXu Zpravodaj Českoslove elů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zp ého sdružení uživatelů TeXu Zpravodaj Československého sdružen vodaj Československého sdružení uživatelů TeXu Zpravodaj Česko živatelů TeXu Zpravodaj Československého sdružení uživatelů TE venského sdružení uživatelů TeXu Zpravodaj Československého sd Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Č ení uživatelů TeXu Zpravodaj Československého sdružení uživate skoslovenského sdružení uživatelů TeXu Zpravodaj Československ TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpra sdružení uživatelů TeXu Zpravodaj Československého sdružení uži j Československého sdružení uživatelů TeXu Zpravodaj Českoslove elů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zp ého sdružení uživatelů TeXu Zpravodaj Československého sdružení vodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení u živatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Českoslo venského sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdruž ení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Če skoslovenského sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravoda

ZPRAVODAJ

ení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Če skoslovenského sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů

Československého sdružení uživatelů TeXu

elů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československ ého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpra vodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení u živatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Českoslo venského sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdruž ení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Če skoslovenského sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravoda j Československého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživat elů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpravodaj Československ ého sdružení uživatelů TeXu Zpravodaj Československého sdružení uživatelů TeXu Zpra

4
2009

OBSAH

Vykreslení středové části obrazce Shri Yantra pomocí METAPOSTu

Luís Nobre Gonçalves, Michal Mádr, Pavel Stříž

Abstrakt

Tento článek popisuje, jak lze pomocí programu METAPOST jednoduše nakreslit komplikovaný obrazec – středovou část kultovního indického symbolu Shri Yantra. Obrazec se skládá z devíti trojúhelníků rozmístěných uvnitř kruhu. Vztahy mezi trojúhelníky jsou popsány několika pravidly a vytvoření kresby respektující tato pravidla je netriviální. Článek popisuje úspěšný pokus, jak se k správnému nakreslení obrazce přiblížit.

Za pečlivé násobné pročtení rukopisu patří naše poděkování Vítu Zýkovi.

Klíčová slova: METAPOST, Shri Yantra, Yantra, Indie, indické symboly, optimalizace, metoda pokus-omyl, simulace, náboženské symboly, kultovní symboly.

Shri Yantra properties

Throughout history, the Shri Yantra has been drawn in many ways but its most used (plane) form is similar to the one shown in Figure 1 on page 213. These are the properties of the drawing:

- There are nine interlocking triangles inside a circle. Four triangles point upwards and five point downwards.
- The two largest triangles touch the outer circle on all three apexes.
- Except for the two largest triangles, all apexes touch the base of another triangle.
- There are 37 intersections[1] resulting from three lines meeting at a point.[2]
- There are 24 intersections resulting from two lines crossing.
- The figure respects symmetry by reflection on a vertical axis.

When we talk about Shri Yantra in this text, we mean a drawing having the above mentioned properties. In the rest of the article we will present a META-POST program attempting to draw a Shri Yantra. First, we will generally describe how we went about writing the program. Then, we will list the source code of the program. Finally, we will discuss how to attain a configuration for the program to achieve an acceptable result.

---

[1] Different requirement on the drawing exists, that specifies 33 such intersections.

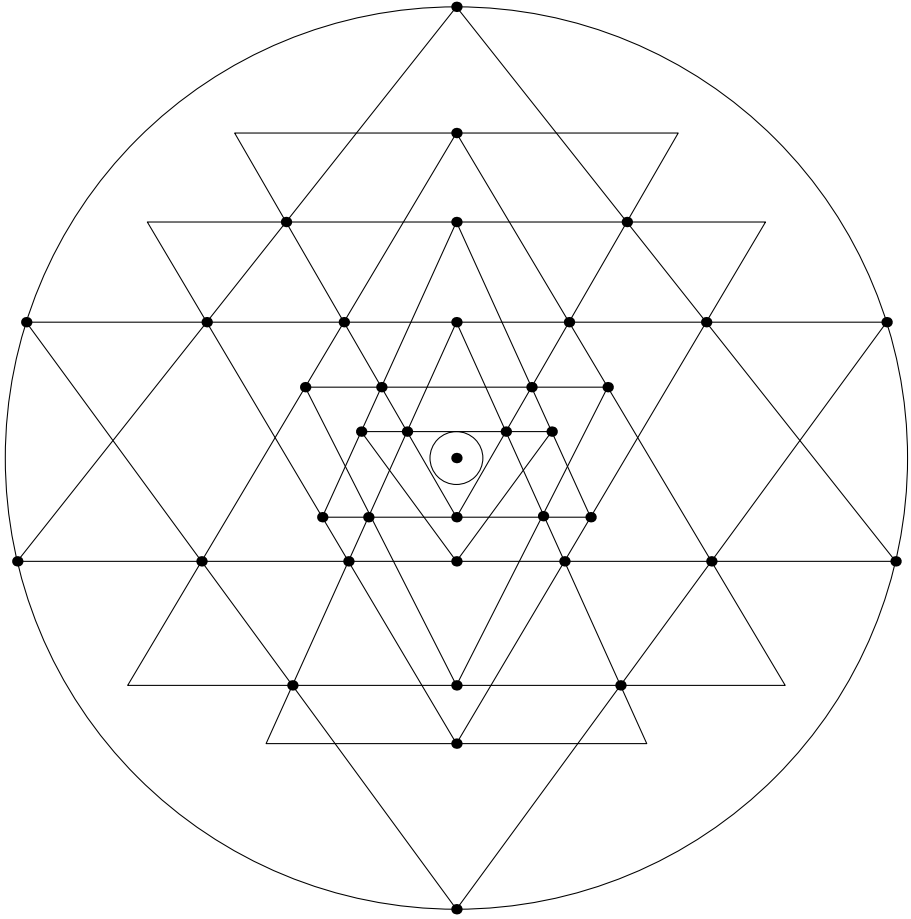[2] The six apexes of the two largest triangles touching the circle are considered to belong to these 37 points.

Figure 1: Shri Yantra with the 37 triple intersections marked with black dots. Parameter values (described later) are: A=0.63295, B=0.22896, C=0.13113, D=0.44115, E=0.68517.

How we went about the drawing

We wrote a program in METAPOST that tried to emulate the Shri Yantra drawing from Figure 2 on page 215.[3] The key thing that the program must do is to locate the forty points 1–40.[4] To achieve that, the locations of all the forty points are described in the program using typical METAPOST statements. Figure 2 guided us in formulating the statements. For example, from the figure we could see that the point $z10$ lies halfway between the points $z4$ and $z5$, thus we described the location of $z10$ in the program simply by the statement:

```
z10 = 0.5[z4,z5];
```

Also from Figure 2 we could see that the point $z13$ lies at the intersection of the lines $z11$--$z12$ and $z4$--$z6$, thus we described the location of $z13$ as:

```
z13 = whatever[z12,z11];
z13 = whatever[z4,z6];
```

Once three points of any of the triangles (e.g., $z14$, $z15$, and $z16$) had been located, we could finally draw it:

```
draw z14--z15--z16--cycle;
```

But how do we start? Well, there are at least two standard initial steps:
1. Choose a frame of reference.
2. Choose the parametrization.

Let's use a frame of reference with its origin at the center of the circle (of unitary ray), with one horizontal axis pointing to the right (x) and with one vertical axis point up (y). The parametrization is achieved by trial-and-error. Among all possibilities, one chooses arbitrarily the drawing sequence. The coordinates of placements which are not imposed, are parameters.

Of course, I needed to *choose* the location of several points to then be able to start describing other points as shown above. As it turned out, there were only five coordinates of points that needed to be chosen throughout the drawing. Positions of all the rest of the points followed from these five selections and from the properties of the drawing. We denoted the five selected values A–E and created from them five numeric parameters that needed to be passed to our program. Let's describe them:

---

[3]If you are wondering – yes, such drawings were available even before we wrote our program, see articles [1, 3].

[4]Some of these points are apexes of triangles, other points are just used by the program to locate other apexes.
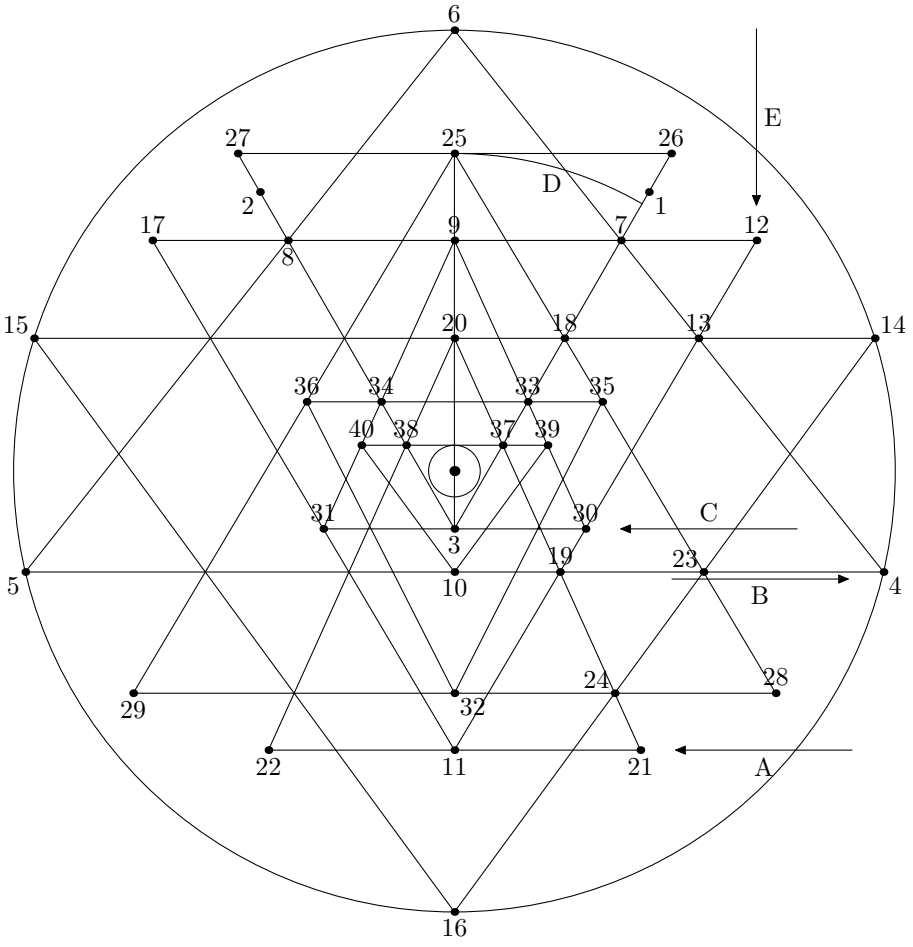
Figure 2: Shri Yantra from Figure 1 on page 213 with points constructed by our program labeled. Arrows highlight points located using the program parameters A–E.

- Parameter A designates the x-coordinate of point 1, parameter D designates the y-coordinate of point 1. Point 1 is a helping point. It is used only to locate other points. Parameter A also designates the (symmetric of the) y-coordinate of point 11, so parameter D is only a measure of the angle between `z3--z25` and `z3--z26`.
- Parameter B designates the y-coordinate of point 4.
- Parameter C designates the y-coordinate of point 3.
- Parameter E designates the x-coordinate of point 12.

Whether the drawing produced by the program will be successful depends on the values of these parameters. We will discuss them after we list the program.

The program in METAPOST

The drawing sequence in METAPOST commands are as follows. Parameter D, as indicated in Figure 2 on page 215, is equivalent to $\arctan(D/(A+C))$ as in this drawing sequence. The full METAPOST program used to search for the correct Shri Yantra configurations may be found at the author's site [2].

```
pA := A*u;
pB := B*u;
pC := -C*u;
pD := D*u;
pE := E*u;
path outercircle;
outercircle = fullcircle scaled (2*u);
draw outercircle;
z1 = (pD,pA);
z2 = (-pD,pA);
z3 = (0,pC);
z4 = ((-u,-pB)--(u,-pB)) intersectionpoint reverse outercircle;
z5 = (-x4,-pB);
z6 = (0,u);
draw z4--z5--z6--cycle;
z7 = (z1--z3) intersectionpoint (z4--z6);
z8 = (-x7,y7);
z9 = 0.5[z7,z8];
z10 = 0.5[z4,z5];
z11 = (0,-pA);
y12 = y7;
x12 = pE;
z13 = whatever[z12,z11];
```

```
z13 = whatever[z4,z6];
z14 = ((-u,y13)--(u,y13)) intersectionpoint outercircle;
z15 = (-x14,y13);
z16 = (0,-u);
draw z14--z15--z16--cycle;
z17 = (-x12,y12);
draw z12--z11--z17--cycle;
z18 = whatever[z1,z3];
z18 = whatever[z14,z15];
z19 = whatever[z11,z12];
z19 = whatever[z4,z5];
z20 = 0.5[z14,z15];
y21 = y11;
z21 = whatever[z20,z19];
z22 = (-x21,y21);
draw z21--z22--z20--cycle;
z23 = whatever[z14,z16];
z23 = whatever[z4,z5];
z24 = whatever[z21,z20];
z24 = whatever[z14,z16];
x25 = 0;
z25 = whatever[z23,z18];
y26 = y25;
z26 = whatever[z3,z7];
z27 = (-x26,y26);
draw z27--z26--z3--cycle;
y28 = y24;
z28 = whatever[z23,z25];
z29 = (-x28,y28);
draw z25--z28--z29--cycle;
z30 = (z3--(u,pC)) intersectionpoint (z13--z11);
z31 = (-x30,y30);
draw z30--z31--z9--cycle;
z32 = 0.5[z28,z29];
z33 = (z3--z1) intersectionpoint (z30--z9);
z34 = (-x33,y33);
z35 = whatever[z25,z28];
z35 = whatever[z33,z34];
z36 = (-x35,y35);
draw z32--z35--z36--cycle;
z37 = (z21--z20) intersectionpoint (z1--z3);
z38 = (-x37,y37);
```

```
z39 = whatever[z37,z38];
z39 = whatever[z30,z9];
z40 = (-x39,y39);
draw z10--z39--z40--cycle;
```

Finding suitable parameter values

As already mentioned, we will not be able to draw Shri Yantra using the listed program above unless we pass suitable values for the five parameters A–E. Figure 3 on page 219 shows what may happen if we pass unsuitable parameter values.

Let's mention how we can obtain suitable values for the parameters. We have seen a drawing using one suitable set of parameter values in Figure 1 on page 213, another one can be seen in Figure 4 on page 220.

One approach could be to measure the *amount of disrespect* in the triple intersections and then minimize it. Knowing that a disrespected triple intersection forms a triangle[5], the *amount of disrespect* may be the length of its sides or its area. We could use any minimization method.

However, another approach worked for us. In fact, it is the least sophisticated method: we randomly generated, e.g., five thousand sets of parameter values, tried them all and registered the set that produced the smallest *amount of disrespect*. With this approach, we needed to take two more things into consideration, though.

First, for some sets of parameter values it may not be possible to reach the end of the drawing sequence, therefore, one must implement defenses against program crashes. The METAPOST interpreter, for instance, crashed when calculating the `intersectionpoint` of two non-intersecting `path`s[6].

Second, using the criterion of the *amount of disrespect* mentioned above, many sets of parameter values may be equally suitable. In order to choose the best one from among them we may also consider some additional criteria:

Amount of nonalignment – All 27 triangle edges could have their global center of mass coincident with the center of the circle.

Amount of inner disharmony – The smallest central triangular area[7] could have all its edges tangent to a concentric circle.

Amount of heterogeneity – the range of distances between consecutive intersections should not be very wide, meaning that consecutive intersections should be very near.

Amount of outness – Triangles should not go out of the outer circle.

---

[5]In fact, a disrespected triple intersection forms three double intersections.
[6]The solution is to check that `intersectiontimes` does not return $(-1,-1)$.
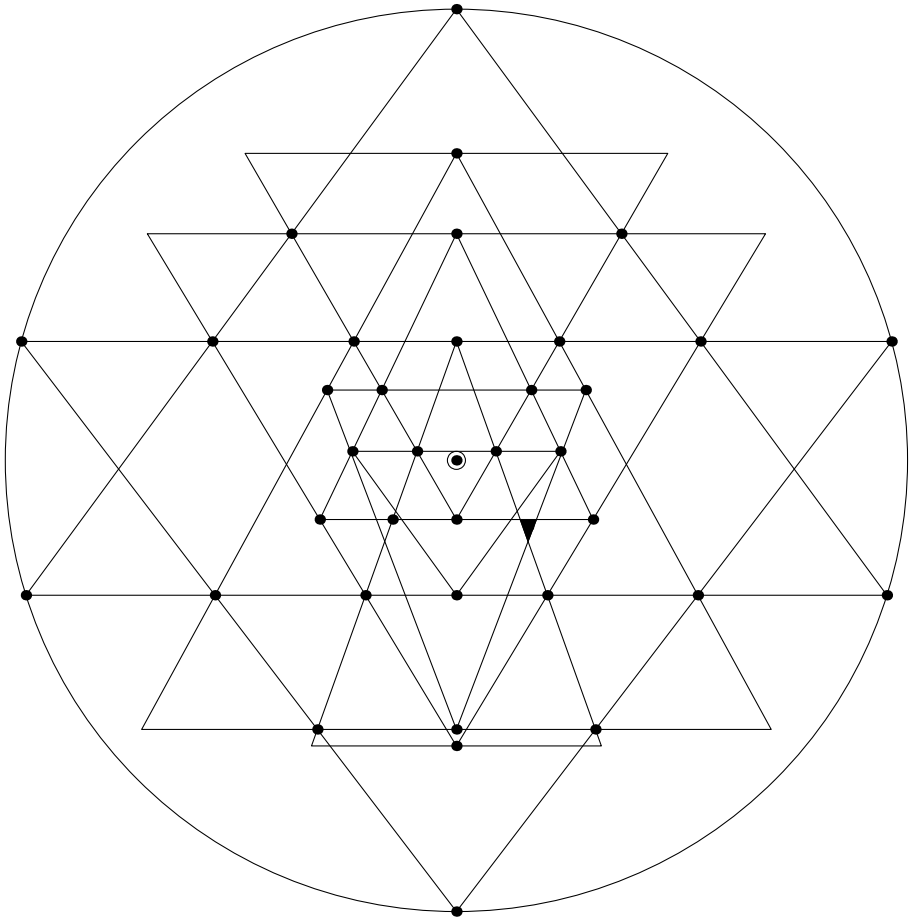[7]`z3--z37--z38--cycle`

Figure 3: Shri Yantra drawn using unsuitable parameters. One triple intersection is not respected due to a change in parameter B (see the black triangle). A=0.63295, B=0.29896, C=0.13113, D=0.44115, E=0.68517.
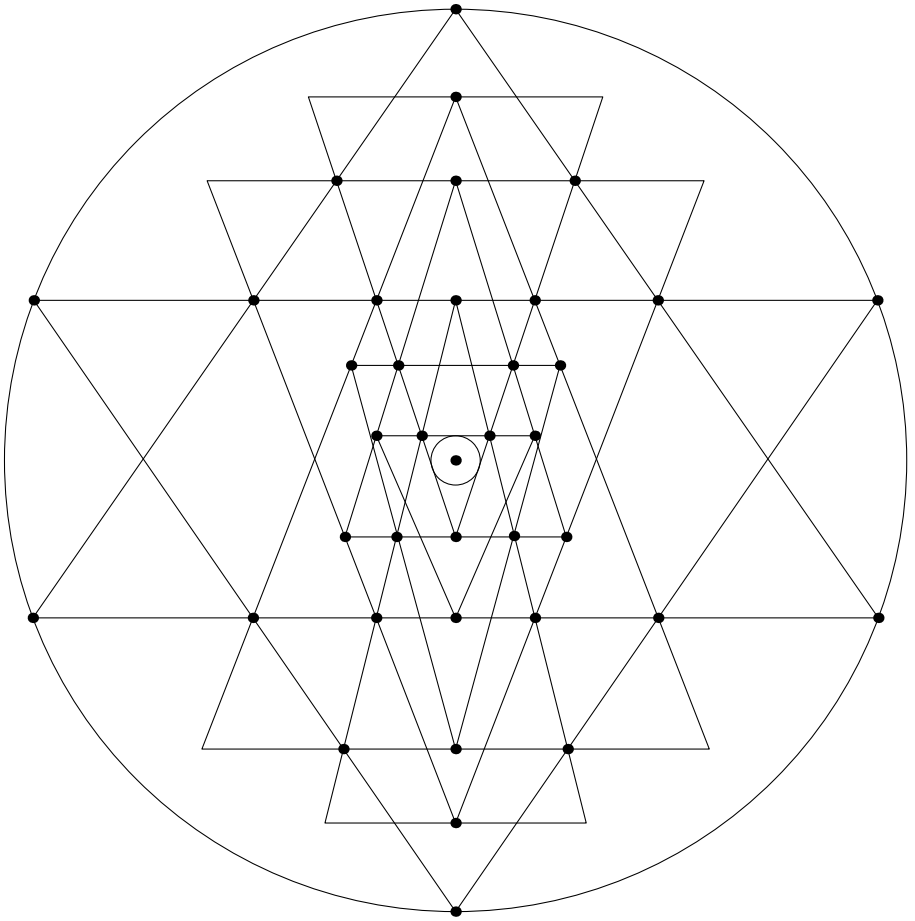
Figure 4: Shri Yantra with a high *amount of heterogenity*.
A=0.80368, B=0.34901, C=0.16962, D=0.32564, E=0.55078.

Conclusion

In this article we have presented a program written in METAPOST that tries to draw the picture of Shri Yantra. The "quality" of the drawing depends on five parameters that we pass to the program. We have described the approach that we used to obtain suitable values for these parameters. We have illustrated two different sets of suitable parameter values.

References

[1] Huet, Gérard. Śrī Yantra Geometry. Elsevier Science B.V.: *Theoretical Computer Science*, vol. 281(**1**−**2**): 609–628, 2002. ISSN 0304-3975. Also available on-line from http://www.sciencedirect.com/ or may be downloaded from the author's website: http://yquem.inria.fr/~huet/PUBLIC/Nivat.ps

[2] Gonçalves, Luís Nobre. The sriyantraquest.mp program. [cit. August 1, 2009] The full source-code is available at: http://matagalatlante.org/nobre/MetaPost/PlainMetaPostProgs/sriyantraquest.mp. The program listing is also available at: http://matagalatlante.org/nobre/MetaPost/PlainMetaPostProgs/sriyantra.mp

[3] Rao, C. S. Śrīyantra – a study of spherical and plane forms. [Shri Yantra – studie kulovitých a rovinných forem.] *Indian Journal of History of Science*, vol. 33(**3**): 203–227, 1998. ISSN 0019-5235.
URL: http://www.sriyantraresearch.com/References/Rao.pdf

Summary:
Drawing the Shree Yantra Core with METAPOST

The Shree Yantra Core is a figure composed of nine interlocking isosceles triangles, all inside a circle. This figure is found in some of the oldest hindu temples and contains, in itself, a series of interesting mathematical problems. This article describes an attempt to solve the problem of drawing the Shree Yantra Core with METAPOST.

Key words: METAPOST, Shree Yantra Core, India, Optimization, Trial-and-Error, Simulation, Religion Symbols, Eastern Mysticism.

Luís Nobre Gonçalves, nobre@cii.fc.ul.pt
CFMC/UL, Complexo Interdisciplinar da Universidade de Lisboa
Avenida Professor Gama Pinto 2, P-1649-003 Lisboa, Portugal

Michal Mádr, m.madr@seznam.cz, *Pavel Stříž,* striz@fame.utb.cz
ÚSKM FaME UTB ve Zlíně, Mostní 5139
Zlín, CZ-760 01, Czech Republic

Zřízené poštovní aliasy sdružení CSTUG:

bulletin@cstug.cz, zpravodaj@cstug.cz
  korespondence ohledně Zpravodaje sdružení

board@cstug.cz
  korespondence členům výboru

cstug@cstug.cz, president@cstug.cz
  korespondence předsedovi sdružení

gacstug@cstug.cz
  grantová agentura CSTUGu

secretary@cstug.cz, orders@cstug.cz
  korespondence administrativní síle sdružení, objednávky CD a DVD

cstug-members@cstug.cz
  korespondence členům sdružení

cstug-faq@cstug.cz
  řešené otázky s odpověďmi navrhované k zařazení do dokumentu CSFAQ

bookorders@cstug.cz
  objednávky tištěné TeXové literatury na dobírku

ftp server sdružení:
  ftp://ftp.cstug.cz

www server sdružení:
  http://www.cstug.cz