Solution to the “Hundred-dollar, Hundred-digit Challenge”

Michel Kern*

May 17, 2002

Abstract

We present solutions to the 10 problems in the “Hundred-dollar, Hundred-digit Challenge”, along with a few words on the solution procedures.

1 Summary of the results

Problem	Solution
1	0.3233674317
2	0.9952629196
3	$0.1274224153\ 10^1$
4	-$0.3306868648\ 10^1$
5	0.2143352346
6	$0.6191395447\ 10^{-1}$
7	0.7250783463
8	0.4240113870
9	0.7859336744
10	$0.3837587979\ 10^{-6}$

Table 1: Summary of the 10 magic numbers

2 General remarks

As the hint advises: these problems are hard! In a few cases, how to compute *some* approximation to the answer is clear, in most of the others, it is not even clear how to get any answer, let alone an accurate one. Also, for someone accustomed to discretizing continuous problems, a source of psychological difficulty is the mere fact of trying to compute solutions to 10 digits accuracy!

For all problems, the main decision was find the right tool for the job. Most of the time, the task was to compute an integral, or a series, or solve an equation to 10 digits accuracy. Programming languages, like Fortran or C, are limited to 15 digits accuracy. On the other hand, modern computer algebra systems, like Maple or Mathematica, provide a fairly complete numerical library, including all the tasks listed above, in arbitrary precision arithmetic. In most cases, this was the deciding factor. Of course in a few cases, Maple was not the right tool for the jobs: two of the problems reduced to linear algebra problems, and Matlab is much better suited to “pure” linear

*INRIA, Rocquencourt, BP 105, Le Chesnay Cedex, France. E-mail: `Michel.Kern@inria.fr`

algebraic problems. In one other case, I found a Matlab toolbox, and in the last one a special purpose Fortran code (though Maple finished the job).

Of course Maple is not the final answer to all these problems. In each case, one has to find some specific way of assessing what accuracy to expect. I try to address this point below.

3 The problems

3.1 A singular integral

This problem becomes much simpler when introducing the Lambert W function, defined as the inverse function to $x \to x \exp x$. This function has been extensively studied in [4], and is known to Maple.

The change of variable $x = \exp(-W(z))$ (so that $z = -\dfrac{\ln x}{x}$ and $\dfrac{dx}{x} = W'(z)dz$) transforms the integral to

$$I_\epsilon = \int_0^{1/\epsilon \ln(1/\epsilon)} \cos z \, W'(z) \, dz. \tag{3.1.1}$$

As $1/\epsilon \ln(1/\epsilon)$ goes to ∞ as ϵ goes to zero, the limit is the same as the limit for $A \to \infty$ of

$$I_A = \int_0^A \cos z \, W'(z) \, dz. \tag{3.1.2}$$

It is still not obvious that the limit is finite. To see this, we recall that W is asymptotic to $\ln z$ for large z, and so $W^{(k)}$ will be asymptotic to z^{-k}, for $k > 0$. By integrating by parts and differentiating W (although my first thought was to *integrate* W), we can make the integral absolutely convergent. In order to obtain an easily computable integral, we integrate by parts several times: doing it three times seems like an acceptable compromise, giving z^{-4} decay. We obtain :

$$\lim_{\epsilon \to 0} I_\epsilon = -W''(0) + \int_0^\infty \sin z \, W^{(4)}(z) \, dz, \tag{3.1.3}$$

with $W''(0) = -2$.

It is “clear” (though I can’t prove it) that $W^{(4)}$ is negative and increasing on $\mathbf{R}^+$. By integrating between the zeros of the sine function, the integral can be seen as an alternating series, so that the remainder is bounded by the first neglected term. To keep matters simple, I argue as follows: I need an integer k such that $\int_{2k\pi}^{(2k+1)\pi} \sin z \, |W^{(4)}(z)| \, dz < 10^{-10}$. By the mean value theorem, there is a $\xi_k \in [2k\pi, (2k+1)\pi]$ such that

$$\int_{2k\pi}^{(2k+1)\pi} \sin z \, |W^{(4)}(z)| \, dz = \pi |W^{(4)}(\xi_k)| \leq \pi |W^{(4)}(2k\pi)|. \tag{3.1.4}$$

A little experimentation shows that taking $k = 300$ should give 11 digits accuracy. Indeed, using Maple with 15 digits accuracy, I computed the integral for several values of k, and the difference between the values obtained for $k = 240$ and $k = 300$ is in the 11th digit. Eventually

$$\int_0^{300\pi} \sin z \, W^{(4)}(z) \, dz \approx -1.6766325683 \tag{3.1.5}$$

so the requested value is .3233674317, with all digits shown believed to be correct.

3.2 Photon scattering

This problem is simple in principle, but was much harder to solve than I expected. The basic idea is straightforward : after each collision, find which mirror will be hit next, compute the intersection point, find the next direction for the photon.

Each of the three steps is itself simple:

find next mirror: the only way I could think of was a systematic search. The search can be restricted to one quarter of the plane by considering the half-line on which the photon moves, and can be speeded-up by searching along diagonals, or anti-diagonals (depending on the quarter-plane).

compute intersection: This is completely trivial in principle, but most likely quite tricky numerically. One has to solve a second degree equation, and some of them seem to be quite ill-conditioned. I have resorted to using high precision computations in Maple. Details are given below.

Find next direction: Again, this is quite simple: compute the symmetric of the incoming line with respect to the diameter through the intersection point.

The last point is not reached. One has to backtrack along the last segment, by an amount proportional to the amount by which the length exceeds 10.

It is quite clear that a geometric engine will be a very handy tool, if not an essential one, so one does not have to take care of the low-level tasks such as : computing the intersection of a line with a circle, taking the symmetric of a line with respect to another line,... My first attempt was to (mis)use the 2D mesh generator `emc2` [10] by Frderic Hecht (available at `http://www-rocq.inria.fr/gamma/cdrom/www/emc2/fra.htm`. It includes a geometric engine, able to compute the intersection of a line with a circle, and to draw the symmetric of a line with respect to another line. This is all that is needed for this problem. The main drawback is that this software is written partly in single precision, so I could certainly not get 10 digits accuracy for this problem (this should not be taken as a criticism of the software, after it a mesh generator, where 10 digits accuracy is usually not needed). As I found out later, things are much worse: single precision will give a completely wrong result... For later comparison, I show the results obtained with `emc2` on figure 1.

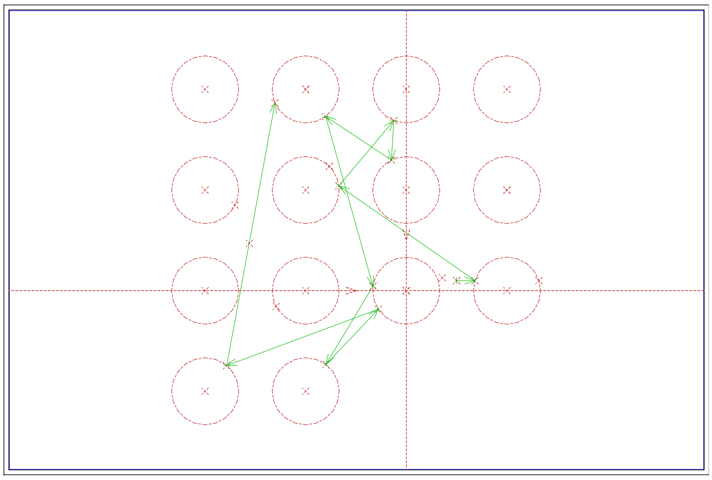

Figure 1: Photon scattering by circular mirrors, results with `emc2`

I then switched to Maple. Maple has a nice `geometry` package, that lets you manipulate directly geometric entities such as points, lines, segments, circles. A line can be defined by two points or by its Cartesian equation, and a circle can be defined by its center and radius, or by its Cartesian equation. Basic geometric operations such as reflection and intersection between objects are supported. And of course, the package benefits from the usual Maple features. In that case the important points will be the ability to use arbitrarily high precision arithmetic.

I wrote a short Maple program solving the problem, and ran it first in 10 digits accuracy (the default). To my surprise, the result was quite different from the one I had gotten with `emc2` (and not just a refinement). I then went to 20 digits, and obtained yet another result. Pictures of the trajectories obtained with Maple are shown in figure 2. They are qualitatively different (look at the last three reflections), and also different from the one obtained with `emc2` (compare figure 1 above). One sees that in order to obtain *qualitatively* correct results, at least 20 digits are needed.

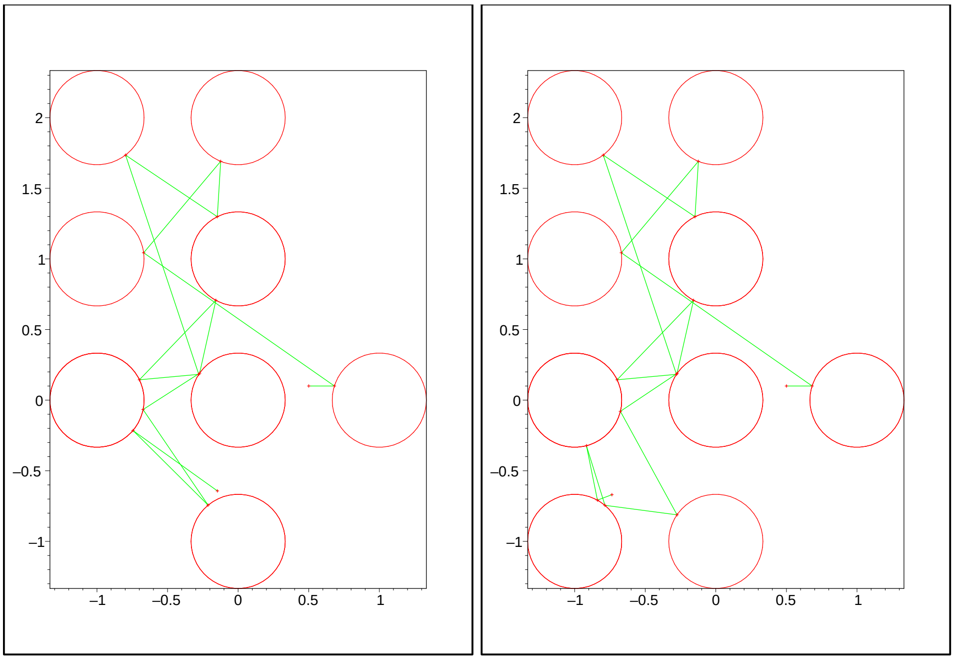

Figure 2: Trajectories obtained using Maple. Left: 10 digits accuracy right: 20 digits accuracy.

Now, I need to obtain (and justify) 10 digits accuracy. To do this, the easiest way seems to simply increase the number of digits used by Maple (though it is known that this is not foolproof). Using 40, then 80 digits (computer time is cheap), confirmed the result with 20 digits. The computation seems to be *very* sensitive to the precision used. I lose between one and two digit at each new collision. This is most likely due to “sensitive dependence on the initial conditions”, and is related to the chaotic nature of this “billiard”.

Figure 2 shows some of the results obtained with Maple (compare figure 1 above), while table 2 shows the difference between 20 and 40 digits (digits that differ are italicized). The results

	First point		Last point
x	20 digits	-.66949971878499157*767*	.69338200*642544047977*
	40 digits	-.66949971878499157*783*	.69338200*475953252931*
y	20 digits	1.0433667525635879516	-.13075365*053191627015*
	40 digits	1.0433667525635879516	-.13075364*662535335423*

Table 2: Results with varying number of digits. First line: first collision point, second line: last collision point

for 80 digits agree with those for 40 digits, to 20 digits. The results given below are taken from a 40 digits computation, truncated to 10 digits.

The time just before the last (extraneous) collision is 11.52786491, and the time along the last segment is 1.637361901. Eventually, at $t = 10$, the photon is at a distance .9952629196 from the origin (the coordinates of the corresponding point are $(-.73629269843, -.6696426968)$).

3.3 Norm of infinite matrix

The matrix is bounded on l^2 because it is actually a Hilbert-Schmidt operator, because obviously:

$$\sum_{i,j} a_{ij}^2 = \sum_{n=1}^{\infty} \frac{1}{n^2} < \infty.$$

Thus $\|A\| \leq \pi/\sqrt{6} \approx 1.2825$.

I cannot see a way of simplifying this problem, so I used brute force. That is, I replaced the infinite matrix by a finite section, as large as I could handle, and let Scilab compute the norm Scilab is a free, interactive, scientific software package for numerical computations, with a syntax close to that of Matlab. It is available at `http://www-rocq.inria.fr/scilab/`, and is described in the book [3].

The first task is to find a convenient way to generate the matrix entries. A little experimentation (helped by Sloane's "Online Encyclopedia" [5], at `http://www.research.att.com/~njas/sequences/`) reveals that:

$$a_{ij} = \frac{1}{(i+j-1)(i+j-2)/2+j}. \tag{3.3.1}$$

Next I computed the norm of this matrix for several sizes. Results are show in table 3:

n	norm(A)
500	1.27422411595291
1000	1.27422414812948
1500	1.27422415142163
2000	1.27422415222862
2500	1.27422415251711
3000	1.27422415264495
3500	1.27422415271009
4000	1.27422415274670

Table 3: Norm of truncated matrix

As expected, these values are less then $\pi/\sqrt{6}$. Without further understanding of the limit process as the size of the matrix goes to infinity, it is difficult to assess whether or not convergence has actually taken place.

A plausible value, probably accurate to 6 to 8 digits is 1.274224153.

3.4 Global minimum of a noisy function

I proceed in three steps:

1. Bound the region where the minimum lies;
2. Using a global minimization algorithm, locate an approximation to the minimum, so that the function is convex in a neighborhood of this approximation;
3. Refine this approximation with Newton's method.

For the first part, a little experimentation shows that for $|x| \geq 3,\ |y| \geq 3$, we have

$$J_4(x,y) \geq \frac{18}{4} + e^{-1} - 4 \geq J(0,0) = 1 + \sin(60),$$

so the minimum lies in the rectangle $[-3,3] \times [-3,3]$.

For the global optimization, I used the DIRECT algorithm. Kelley [11] gives it as one that addresses the problem of locating the global minimum of a (possibly noisy) function. He mentions an implementation by Gablonsky [7]. Fortunately, this implementation is available on the Web, at `http://www4.ncsu.edu/eos/users/c/ctkelley/www/optimization_codes.html`.

All one has to do to use the code is write a function (in Fortran) implementing the cost function, provide bounds for the variables, and set a few parameters (tolerance, number of function evaluations). It was not difficult to obtain a value for the minimum of -3.3068686, reached at the point $(-0.0244033, 0.2106123)$. This actually used 73929 function evaluations, and 91 seconds on a Pentium II 366 PC.

Now one can use a local minimization method, after checking that the function is actually convex over the rectangle $-0.04 \leq x \leq 0.01$, $0.18 \leq y \leq 0.24$. Letting Maple solve for zeros of the gradient of our cost function improves the minimum to -3.30686864747527, reached at the point $x = -.0244030796943752, y = .210612427155356$. The computation was carried out with 15 digits, so x and y should be accurate to 10 digits, and the value of the minimum should be more accurate (see the discussion on this issue in problem 9).

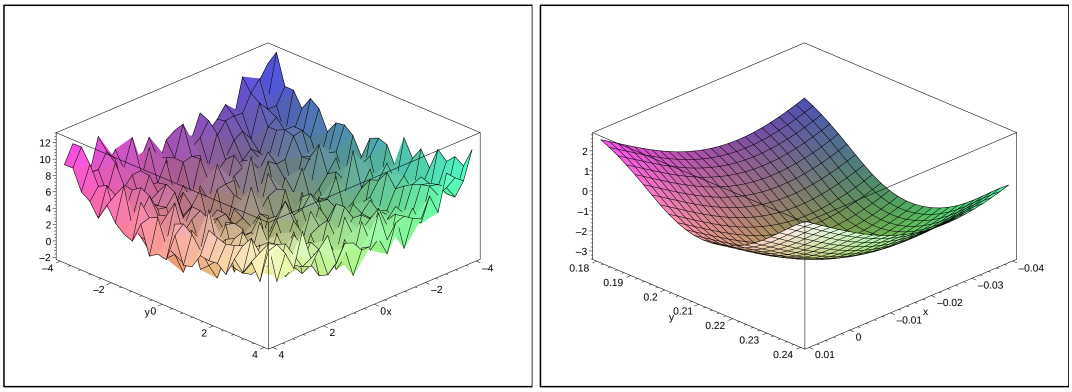

Figure 3: Cost function and a zoom near the global minimum

3.5 Complex best approximation

For this problem, I have had the good luck to find all the software I needed on the web, without necessarily understanding the theory behind it. I first downloaded the `coca` toolbox by B. Fischer and J. Modersitzki, at `http://www.math.mu-luebeck.de/workers/modersitzki/COCA/coca5.html`. This is a Matlab toolbox designed for solving exactly the problem at hand, to wit finding best approximations in the complex plane. It assumes the function to be approximated is analytic in the region of interest (which is the case here), so that the minimum is attained on the boundary.

That left me with the task of computing accurate approximation to the complex Γ function. Here I used a Matlab file from P. Godfrey, at `http://winnie.fit.edu/~gabdo/gamma.m` (described in[8]), based on an approximation due to Lanczos, that claims 15 digits accuracy. This is somewhat difficult to check, as I have no other means of computing the (complex) gamma function.

I show below some of the Matlab commands I used for solving this problem.

```
FUN.name='igamma';
BASIS.name      = 'monom';      % standard basis
BASIS.choice=[0:3];             % cubic polynomial
BASIS.C_dim     = length(BASIS.choice);
```

```
BOUNDARY.name    = 'gcircle';    % name of boundary
real_coef        = 1;
PARA.relative_error_bound = 1e-10;
PARA.stepsize             = 500;    % number of steps in COMPNORM.M
PARA.max_iterations       = 100;
[PARA] = coca(PARA);

PARA.error_norm
ans = 0.21433523459984
```

I also show, on figure 4, the error on the boundary of the unit circle.

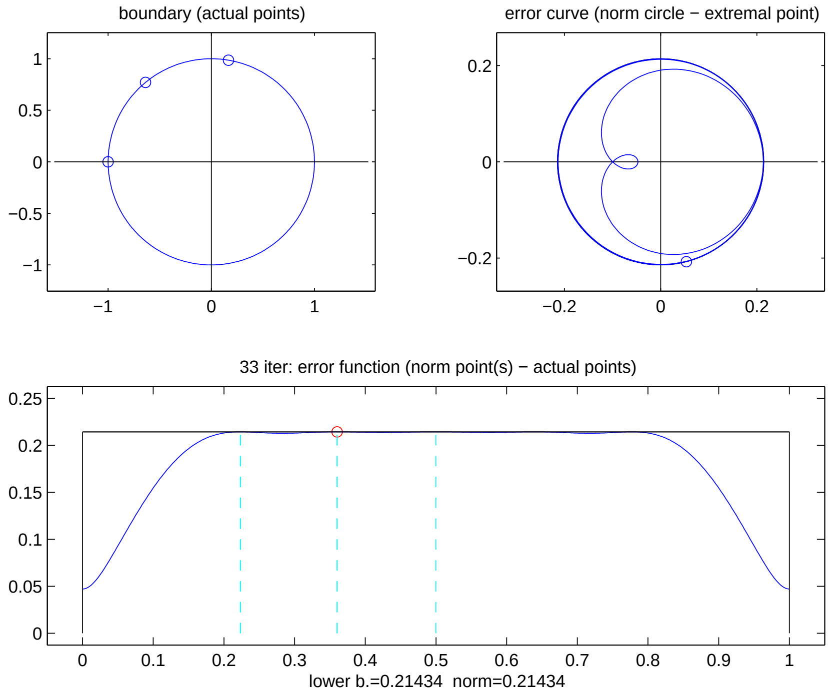

Figure 4: Error for the complex approximation problem

Accuracy is hard to estimate here mainly because of my lack of understanding of the method used.

3.6 Biased random walk on a lattice

We follow the exposition in Chapter 2 of Barber and Ninham [1].

Let $P^n_{k_1,k_2}$ be the probability that the flea is at point (k_1, k_2) after n steps, let $P_{k_1,k_2}(z) = \sum_{n=0}^{\infty} P^n_{k_1,k_2} z^n$ be its generating function, and let $G(\phi, z)$ be its Fourier transform (with $\phi = (\phi_1, \phi_2)$:

$$G(z\phi, z) = \sum_{k_1,k_2} e^{ik.\phi} P_{k_1,k_2}(z).$$

The probability that the flea returns to the origin at step n is P^n_{00}. According to the recurrence theorem of Feller [6], the probability that the flea ever returns to the origin is given by $1 - 1/u$, where $u = \sum_{n=0}^{\infty} P_{0,0}^n = P_{0,0}(1)$. Inverting the Fourier series, we end up with the expression

$$u = \frac{1}{4\pi^2} \int_{[-\pi,\pi]^2} G(\phi, 1)\, d\phi.$$

In order to compute G, we start by using the definition of the walk to obtain a recurrence relation among the probabilities $P_{k_1 k_2}^n$:

$$P_{k_1,k_2}^{n+1} = \frac{1}{4} P_{k_1,k_2-1}^n + \frac{1}{4} P_{k_1,k_2+1}^n + (\frac{1}{4} + \epsilon) P_{k_1+1,k_2}^n + (\frac{1}{4} - \epsilon) P_{k_1-1,k_2}^n, \tag{3.6.1}$$

then among the generating functions :

$$P_{k_1,k_2}(z) - z \left(\frac{1}{4} P_{k_1,k_2-1}(z) + \frac{1}{4} P_{k_1,k_2+1}(z) \right. \\ \left. + (\frac{1}{4} + \epsilon) P_{k_1+1,k_2}(z) + (\frac{1}{4} - \epsilon) P_{k_1-1,k_2}(z) \right) = \delta_{0,0}, \tag{3.6.2}$$

using the fact that the flea starts at $(0, 0)$.

Now multiply the last equation by $e^{ik\cdot\phi}$ and sum over all lattice points, giving the following expression for $G(\phi, z)$

$$G(\phi, z) = \frac{1}{1 - z\,(1/4 e^{i\phi_2} + 1/4 e^{-i\phi_2} + (1/4 + \epsilon) e^{-i\phi_1} + (1/4 - \epsilon) e^{i\phi_1})} \tag{3.6.3}$$

Eventually, we need to solve the equation

$$\frac{1}{4\pi^2} \int_{[-\pi,\pi]^2} \frac{1}{1 - (1/2 \cos\phi_2 + (1/4 + \epsilon) e^{-i\phi_1} + (1/4 - \epsilon) e^{i\phi_1})}\, d\phi_1\, d\phi_2 = 2$$

The ϕ_1 integral can be computed by the residue theorem, letting $z = e^{i\phi_1}$. Actually, Maple can perform the whole computation (though I do not know how to prove that the first root is the one inside the unit circle).

```
assume(epsilon>0,epsilon<1/4, phi_1,real);
g:=1/(1-1/2*cos(phi_2)-(1/4-epsilon)*exp(I*phi_2)
                      -(1/4+epsilon)*exp(-I*phi_2)):
g:=normal(1/(I*z)*subs({exp(I*phi_2)=z, exp(-I*phi_2)=1/z},g)):
racs:={solve(denom(g),z)}:
inint:=2*I*Pi*residue(g, z=racs[1]):
```

This gives

$$u = 1/\pi \int_{-\pi}^{\pi} \frac{d\phi_1}{\sqrt{3 - 4\cos\phi_1 + \cos^2\phi_1 + 16\epsilon^2}}$$

It turns out that this integral is a combination of elliptic integrals, that Maple can again evaluate exactly. I cannot resist showing the exact value :

$$u = 8/\pi \frac{1}{\sqrt{2 + 2\sqrt{1 - 16\,\epsilon^2} + 16\,\epsilon^2}} \left[2K \left(2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \right. \\ + F\left(\frac{\sqrt{2}}{2}\sqrt{\frac{3 - \sqrt{1 - 16\,\epsilon^2}}{2 - \sqrt{1 - 16\,\epsilon^2}}}, 2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \\ \left. + F\left(\frac{\sqrt{2}}{2}\sqrt{\frac{1 + \sqrt{1 - 16\,\epsilon^2}}{2 + \sqrt{1 - 16\,\epsilon^2}}}, 2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \right] \tag{3.6.4}$$

where K is the complete elliptic integral of the first kind, and F is the incomplete elliptic integral of the first kind.

Figure 5 shows the variation of u as a function of ϵ. As expected, u goes (slowly) to infinity when ϵ goes to 0 (this is the case of a symmetric walk, and the return probability is 1). The fact that the limit for $\epsilon = 1/4$ is non-zero is somewhat surprising... Maple solved the equation

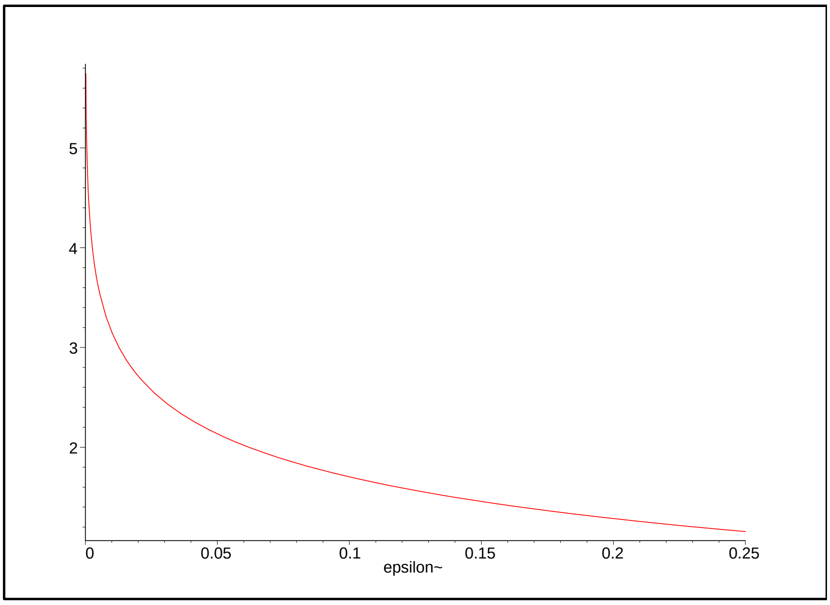

Figure 5: u as a function of ϵ, random walk problem

$u(\epsilon) = 2$ (it is clear that this function is monotone decreasing, so that there is only one solution), giving the solution 0.061913954473991. All digits are believed to be exact (based on a 40 digits computation).

3.7 Matrix inversion

I do not see any other way than numerically solving the system

$$Ax = e_1$$

and taking the first component of x. The matrix A is 20000×20000 with a bandwidth of 16384 and only 31 nonzero elements per line. It is very sparse, but the large bandwidth makes it impossible to use a sparse direct solver. I tried using UMFPACK, and even though reordering using reverse Cuthill-McKee gives the best results, factorization fails for lack of space.

I have resorted to using the conjugate gradient method, with incomplete Cholesky preconditioning, even though I do not know whether or not the matrix is positive definite (I actually do not think it is, but see below).

Here are the Matlab commands I used (the file `primes` contains a list of the first 20000 primes generated by Maple):

```
fid=fopen('primes'); primes=fscanf(fid, '%d,');
diags=2.^(0:14); diags=[-diags(15:-1:1),0, diags];
B=ones(20000,31); B(:,16)=primes;
A=spdiags(B, diags, 20000, 20000);
b=zeros(20000,1); b(1)=1;
R=cholinc(A,'0');
```

```
tol=1e-10; maxit=100;
[x,flag,relres,iter,resvec] = pcg(A,b,tol,maxit,R',R);

x(1)
ans =
  0.72507834626840
```

The conjugate gradient after 7 iterations, so the matrix must be "positive definite enough". It is difficult to estimate the accuracy of this solution, as I can see no way to estimate the condition number of this matrix. A good sign is the quick convergence of the conjugate gradient, and the fact that x_1 seems to be the largest component of x.

3.8 Heat equation

For definiteness, I will take the side $\{x = L\}$ $(L = 1)$ as the one with the temperature maintained at $d = 5$. I also denote the square by Ω, the side $\{x = L\}$ by Γ_d and the 3 other sides by Γ_0.

This problem cries out for a Fourier series solution. However, since the computations are somewhat involved, I started by using the PDE toolbox in Matlab to get a crude approximation to the answer. This way, I found that the time for the temperature at the center of the plate to reach 1 is between 0.42 and 0.43. This will provide a useful check on the semi-analytical solution.

The first step is to get rid of the inhomogeneous Dirichlet boundary condition on Γ_d. This can be done by a harmonic lifting : I solve for the function $u_0(x, y)$ solution of :

$$\begin{cases} -\Delta u_0 = 0 & \text{in } \Omega \\ u_0 = d & \text{on } \Gamma_d \\ u_0 = 0 & \text{on } \Gamma_0. \end{cases} \tag{3.8.1}$$

This can also be done by Fourier series: take $u_0(x, y) = \sum_{n=1}^{\infty} u_n(x) \sin n\pi \frac{(y + L)}{2L}$. One easily finds

$$u_n(x) = \mu_n \frac{\sinh n\pi \frac{(x+L)}{2L}}{\sinh n\pi}, \qquad \text{with } \mu_n = \begin{cases} \frac{4d}{n\pi} & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even.} \end{cases}$$

For later use, note that

$$u_0(0, 0) = \sum_{n=0}^{\infty} (-1)^n \frac{4d}{(2n + 1)\pi} \frac{\sinh(2n + 1)\pi/2}{\sinh(2n + 1)\pi}. \tag{3.8.2}$$

Now, let $u = v + u_0$. Then v solves the problem:

$$\begin{cases} \frac{\partial v}{\partial t} - \Delta v = 0 & \text{in } \Omega \\ v = 0 & \text{on } \partial\Omega \\ v(x, y, 0) = -u_0(x, y) & \text{in } \Omega. \end{cases} \tag{3.8.3}$$

We search for v in the form $v(x, y, t) = \sum_{p,q} v_{pq}(t) \sin p\pi \frac{x + L}{2L} \sin q\pi \frac{y + L}{2L}$. v_{pq} satisfies the ODE :

$$v'_{pq}(t) + (p^2 + q^2)\pi^2/(4L^2)v_{pq}(t) = 0$$

with initial condition $v_{pq}(0) = -b_{pq}$, where b_{pq} is the Fourier coefficient of u_0 :

$$b_{pq} = \frac{2}{\pi} \mu_q \int_0^{\pi} \frac{\sinh q\pi t}{\sinh q\pi} \sin pt \, dt = -\frac{2}{\pi} \mu_q \frac{(-1)^p p}{p^2 + q^2}.$$

Eventually, we have

$$v(x,y,t)=\sum_{p,q}\frac{2}{\pi}\mu_q\frac{(-1)^p p}{p^2+q^2}e^{-(p^2+q^2)\pi^2/(4L^2)t}.$$

Putting everything together, we get the expression for the temperature at the center of the plate, as a function of time :

$$u(0,0,t)=\frac{20}{\pi}\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}\frac{\sinh(2n+1)\pi/2}{\sinh(2n+1)\pi} - \frac{40}{\pi^2}\sum_{q=0}^{\infty}\sum_{p=0}^{\infty}\frac{(-1)^{p+q}\,(2p+1)\,e^{-1/4\left((2p+1)^2+(2q+1)^2\right)\pi^2 t}}{(2q+1)\left((2p+1)^2+(2q+1)^2\right)} \tag{3.8.4}$$

I show, on figure 6, the evolution of the temperature at the center of the plate.

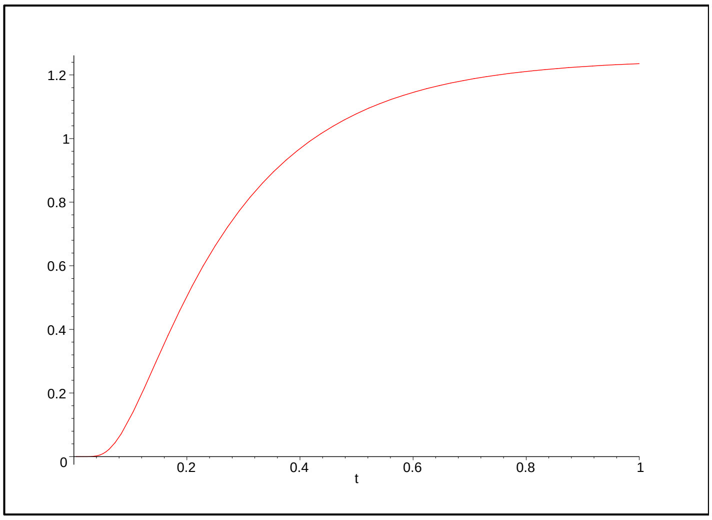

Figure 6: Evolution of the temperature at the center of the plate

Except for small values of t, the series above is rapidly convergent. It is easy to see that the solution is indeed between 0.42 and 0.43. Then, I used Maple to solve the equation $u(0,0,t)=1$ in this interval. The solution is $t=0.424011387033688$, truncated from a 20 digits approximation. This should give at least ten correct digits.

3.9 A parametric integral

As $\alpha>0$, the given integral exists. The given function is highly oscillatory, so I must again proceed in several stages.

I start with a visual examination of the function. Maple can fairly easily compute single precision accuracy approximations to the integral (more on that issue below), and I obtained the graph shown on the left part of figure 7. It is easier to compute the integral if one first takes the factor $2+\sin(10\alpha)$ out of the integral. Maple seems to be able to handle the singularity at $x=2$ without my needing to transform to a unbounded interval. It is visually apparent that the maximum is somewhat less than one. The right part of figure 7 shows a zoom of the graph on the interval $[1/2,1]$, and it seems that the maximum is just less than 0.8. More importantly, on this interval the given function is unimodal, and thus amenable to a local maximization algorithm.

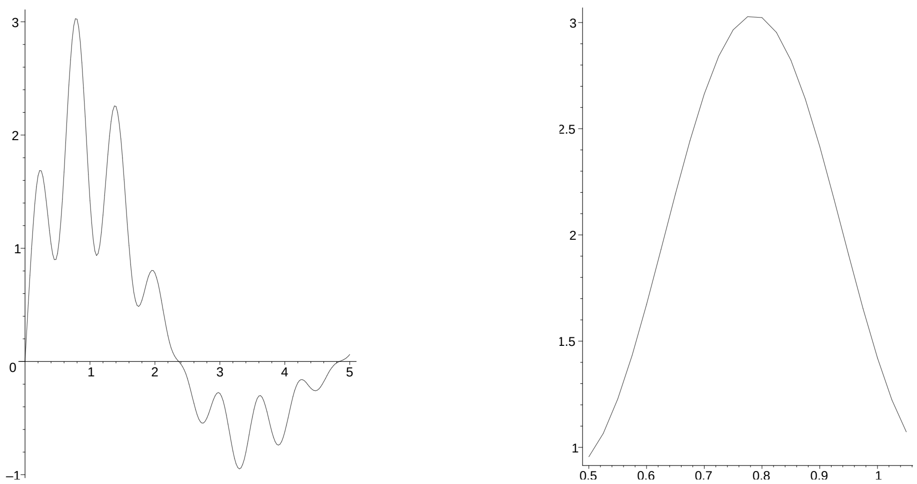

Figure 7: Graph of given integral as a function of α, left: on $[0, 5]$, right: zoom between 0.5 and 1

As it is more difficult (not to say more expensive) to compute the derivative of this function, I chose to use a minimization method that does not require derivatives. The most famous one is probably Brent's `localmin`, described in the book [2]. It is based on inverse parabolic interpolation, safeguarded by a golden search section. I have implemented the code as described in the Algol routine given on page 79 of [2].

Before giving the result, it is worth commenting on the attainable accuracy with this (or any other) method. The detailed discussion in [2] shows that we cannot hope to resolve the location of the minimum of a function f computed in finite precision to more than $2|f_0|\epsilon/(x_0^2 f_0'')^{1/2}$, where ϵ is the machine precision, x_0 is the location of the minimum, $f_0 = f(x_0)$, and $f_0'' = f''(x_0)$. If we want the minimum to 10 digits, we need to compute the function to (at least) 20 digits (a crude estimate of the second derivative is -106). This forces the use of Maple, which (in principle) can compute an integral to any given accuracy.

In practice, I found that Maple 6 can not handle the very stringent accuracy required, and that fortunately Maple 7 can. I used Brent's method on the interval $[0.78, 0.79]$ (I have checked by hand that the maximum is between these two values), and also on smaller intervals obtained by lower accuracy searches. The solution obtained (after 25 iterations, with a tolerance of 10^{-12}) is $\alpha_0 = .7859336744$, the last digit being not quite stable.

3.10 Brownian motion

For definiteness, let $L = 10$ be the length of the sides, and $l = 1$ be the length of the ends (I found it convenient to let the ration l/L vary, so as to check the computation). We denote the rectangle by R.

According to [9, thm 13.7 (5)], the probability that the particle will exit through one of the ends rather than one of the sides is given by the value at the center of the rectangle of the solution to the partial differential equation :

$$\begin{cases} -\Delta p = 0 & \text{in } R \\ p = 1 & \text{on } \{x = 0\} \cup \{x = L\} \\ p = 0 & \text{on } \{y = 0\} \cup \{y = l\} \end{cases} \tag{3.10.1}$$

This problem can easily be solved by Fourier series, as in question 3.8.

Let $p(x,y) = \sum_{n=0}^{\infty} p_n(x) \sin\left(\frac{n\pi y}{l}\right)$, then p_n satisfies the ordinary differential equation

$$p_n''(x) - \frac{n^2\pi^2}{l^2} p_n(x) = 0,$$

so p_n has the form

$$p_n(x) = A_n \sinh\left(\frac{n\pi x}{l}\right) + B_n \cosh\left(\frac{n\pi x}{l}\right),$$

and we can determine A_n and B_n by using the boundary conditions on the ends:

- On the left end $x = 0$: $\sum_{n=0}^{\infty} B_n \sin\left(\frac{n\pi y}{l}\right) = 1$,
- On the right end $x = L$: $\sum_{n=0}^{\infty} \left(A_n \sinh\left(\frac{n\pi L}{l}\right) + B_n \cosh\left(\frac{n\pi L}{l}\right)\right) \sin\left(\frac{n\pi y}{l}\right) = 1$.

By expanding the right hand side (a constant) in a series of sines, we find the coefficients A_n and B_n :

- If n is even, $A_n = B_n = 0$,
- if n is odd, $A_n = 4/n\pi \left(\frac{1 - \cosh n\pi L/l}{\sinh n\pi L/l}\right), \quad B_n = 4/n\pi$.

Eventually, the probability we seek is given by the sum of the series :

$$p(0,0) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{2n+1} \frac{1}{\cosh(2n+1)\dfrac{\pi L}{2l}}. \tag{3.10.2}$$

I have plotted the value of $p(0,0)$ as a function of the ratio $r = L/l$ in figure 8. As expected, the probability is 1 if r goes to zero, and goes to 0 if r goes to infinity.

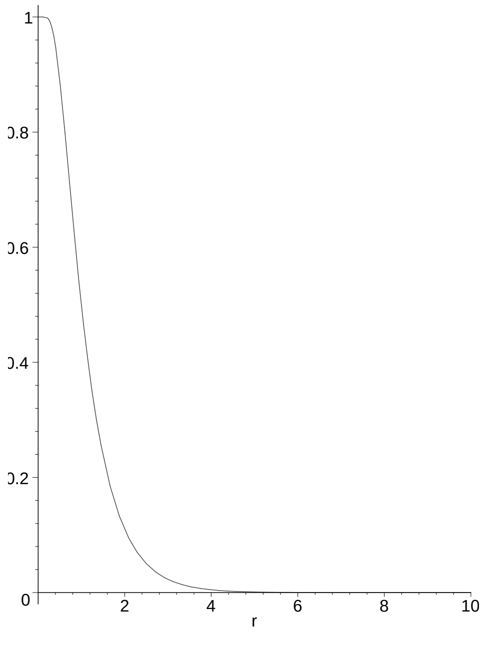

Figure 8: Probability of exit along the ends as a function of aspect ratio

For $r = 10$, as required, Maple gives the value of the sum as $0.3837587979\,10^{-6}$, with all digits thought to be correct.

Acknowledegements

I wish to thanks all the authors of the packages I have used. This work would not have been possible without them. I have also benefitted from conversations with P. Joly, V. Martin, B. Salvy and S. Tordeux.

References

[1] Michael N. Barber and B. W. Ninham. *Random and Restricted Walks – Theory and Applications*. Gordon and Breach, 1970.

[2] Richard P. Brent. *Algorithms for Minimization Without Derivatives*. Prentice Hall Series in Automatic Computation. Prentice Hall, 1973.

[3] Carey Bunks, Jean-Philippe Chancelier, Franois Delebecque, and Claude Gomez, editors. *Engineering and Scientific Computing with Scilab*. Birkhauser, 1999.

[4] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. On the Lambert W function. *Adv. Comp. Math.*, 1996.

[5] N. J. A. Sloane (Editor). The On-Line Encyclopedia of Integer Sequences. published electronically at `http://www.research.att.com/~njas/sequences/`, 2002.

[6] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley and Sons, 2nd edition, 1957.

[7] J. M. Gablonsky. An implementation of the DIRECT algorithm. Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, August 1998.

[8] Paul Godfrey. A note on the computation of the convergent Lanczos complex gamma approximation. published electronically at `http://winnie.fit.edu/~gabdo/gamma.txt`, 2001.

[9] G. R. Grimmet and D. R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, 2nd edition, 1992.

[10] Fréderic Hecht and Eric Saltel. Emc2 un logiciel d'edition de maillages et de contours bidimensionnels. Technical Report RT-118, INRIA, 1990.

[11] C. T. Kelley. *Iterative methods for optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.