Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 207 (2005) 625–638

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Improved Lebesgue constants on the triangle

Wilhelm Heinrichs

Universität Duisburg-Essen, Ingenieurmathematik (FB 10), Universitätsstrasse 3, D-45117 Essen, Germany

Received 15 November 2004; received in revised form 31 January 2005; accepted 1 February 2005
Available online 17 March 2005

Abstract

New sets of points with improved Lebesgue constants in the triangle are calculated. Starting with the Fekete points a direct minimization process for the Lebesgue constant leads to better results. The points and corresponding quadrature weigths are explicitly given. It is quite surprising that the optimal points are not symmetric. The points along the boundary of the triangle are the 1D Gauss–Lobatto points. For all degrees, our points yield the smallest Lebesgue constants currently known. Numerical examples are presented, which show the improved interpolation properties of our nodes.
© 2005 Elsevier Inc. All rights reserved.

MSC: 65M60; 65M70; 65N35; 41A10

Keywords: Triangle; Lebesgue constants; Fekete points; Multivariate approximation

1. Introduction

Points which minimize the Lebesque constant are interesting since they yield optimal interpolation properties. Hence they are also important for the numerical solution of partial differential equations with spectral methods. The standard spectral methods (Galerkin, tau, collocation) are based on these points and corresponding stable quadrature formulas. Since the theoretical progress is very slow, it is necessary to do numerical calculations and experiments. For tensor-product domains as the square the 1D Gauss–Lobatto nodes can be easily extended to the 2D case. This does not mean that they are the best interpolation points but they have good interpolation properties and are efficient in the implementation of boundary conditions and the application of fast Fourier transforms (FFTs) in the Chebyshev case. The Lebesque points (which are know numerically for the interval) have a lower Lebesque constant and thus a tensor product of Lebesque points will have a lower Lebesque constant than the Gauss–Lobatto

E-mail address: wheinric@ing-math.uni-essen.de.

doi:10.1016/j.jcp.2005.02.002

nodes. But also on the square there may be a non-tensor-product set of points with a lower Lebesque constant. On non-tensor-product domains as the triangle the 1D points cannot be extended. In earlier approaches (see [5,6,8,9]) mapping techniques from the square to the triangle were used. This leads to the typical spectral accuracy but due to the corner singularity the condition number becomes large. Other approaches (see [1,2,7,11]) directly deal with finding optimal interpolation nodes on the triangle. Hesthaven [7] uses a second-order time evolution method to minimize an electrostatic energy function. Bos [1], Chen and Babuška [2] and Taylor et al. [11] investigate the Fekete points which maximize the determinant of the Vandermonde matrix. They were explicitly calculated in [11] and applied in [10] to spectral element methods. A good review on these techniques can be found in the book of Deville et al. [3]. Here we start our iteration with the Fekete points and then employ a direct minimization process for the Lebesgue constants. Since the Lebesque function is a significantly more complex function without analytic expressions for its derivatives it is necessary to start the iteration with a good initial guess obtained from easier functionals (Fekete, electrostatics). For the minimization we combine a steepest descent algorithm with a damped Newton method. In the Fekete approach the points along the boundary of the triangle are the 1D Gauss–Lobatto points (see the conjecture of Bos [1]). For calculating the quadrature weights we employ the Dubiner basis functions [4] which yield well conditioned matrices. In the first approach our iteration does not preserve any symmetry conditions. Hence we obtain unsymmetric distributions of nodes. Some quadrature weights (only belonging to the edge of the triangle) become negative and the usefulness of the schemes for numerical integration is doubtful. In a second approach the points follow some (but not all) symmetries of the triangle. The resulting Lebesgue constants are somewhat worse but still better than for the Fekete points. Some quadrature weights still remain negative. Anyhow, our approach produces points with the best known Lebesgue constants. The improved interpolation properties are demonstrated by two numerical examples.

The paper is organized as follows. First we introduce the Lebesgue constant on the triangle (Section 2). Then we describe in Section 3 our minimization procedure. The corresponding numerical results are presented in Section 4. Finally our results are summarized in a conclusion.

2. The Lebesgue constant

We consider an optimization process on the triangle T given by

$$T = \{(x,y): \ x \geqslant -1, \ y \geqslant -1, x+y \leqslant 0\}.$$

For triangles, the usual choice for interpolation is a triangular truncation of polynomials, spanned by the monomials

$$\{x^m y^n, \ m+n \leqslant N\}.$$

The polynomial subspace spanned by these monomials is denoted by $\mathbb{P}_N$ with dim $\mathbb{P}_N = (N+1)(N+2)/2$. Since monomials yield very bad condition numbers for the Vandermonde matrix, it is practically useful to choose Dubiner basis functions [4] given by

$$\left\{L_m\left(2\frac{1+x}{1-y}-1\right)(1-y)^m P_n^{2m+1,0}(y), \ m+n \leqslant N\right\}$$

with the Legendre polynomials L_m and the Jacobi polynomials $P_n^{2m+1,0}$. The Dubiner basis is orthogonal in the triangle and hence leads to a well-conditioned Vandermonde matrix. Taylor et al. [11] obtained for the Fekete points ($N \leqslant 19$) a MATLAB reported condition number less than 50. For given nodes

$$(x_{ij}, y_{ij}), \quad 0 \leqslant i+j \leqslant N,$$

and basis functions ϕ_{mn} the Vandermonde matrix $V = (v_{ij,mn})$ is given by

$$v_{ij,mn} = \phi_{mn}(x_{ij}, y_{ij}), \quad 0 \leqslant i + j \leqslant N, \ \ 0 \leqslant m + n \leqslant N.$$

The Vandermonde matrix represents the interpolation at the above nodes. Its inverse is denoted by $U = V^{-1} = (u_{mn,ij})$. Now, the Lebesgue constant λ_N on the triangle T can be written as

$$\lambda_N = \max_{(x,y)\in T} \left| \sum_{i+j=0}^{N} \left| \sum_{m+n=0}^{N} u_{mn,ij}\phi_{mn}(x, y) \right| \right|. \tag{1}$$

The maximum overall points in the triangle are approximated by the maximum over a grid of 2485 equally spaced points. Here we start a direct minimization process for the Lebesgue constant. After convergence, we further calculate the corresponding quadrature weights w_{ij} by solving the linear system

$$\sum_{i+j=0}^{N} w_{ij}\phi_{mn}(x_{ij}, y_{ij}) = \int \phi_{mn}\,\mathrm{d}T. \tag{2}$$

This requires inverting the Vandermonde matrix, so again it is important that this matrix is well conditioned. For increasing N it becomes necessary to use Dubiner basis functions.

3. The minimization procedure

We start our minimization procedure with the Fekete points which were explicitly calculated by Taylor et al. [11]. Since the method is sensitive to the initial condition for the grid points, it is important to initialize the algorithm with a good initial guess. Without such a good starting approximation, we were not able compute improved Lebesque constants. The main reason is that the Lebesque constant is a significantly more complex function without analytic expression for its derivatives. Hence, it is more difficult to obtain optimal solutions, thus making it necessary to start the iteration with near-optimal nodes from easier functionals (Fekete, electrostatics). Then we apply the steepest descent method to minimize the functional belonging to the Lebesgue constant. It is implemented in the following way. First we approximate the partial derivatives of the Lebesgue function with central finite differences, i.e., for each point (x_{ij}, y_{ij}) we calculate

$$\lambda_{N,x}(x_{ij}, y_{ij}) \cong (\lambda_N(x_{ij} + h, y_{ij}) - \lambda_N(x_{ij} - h, y_{ij}))/2h,$$
$$\lambda_{N,y}(x_{ij}, y_{ij}) \cong (\lambda_N(x_{ij}, y_{ij} + h) - \lambda_N(x_{ij}, y_{ij} - h))/2h,$$

with sufficiently small step size h. Here we choose $h = 1/512$. The method was not very sensitive to the choice of h. Also, larger step sizes of $h = 1/128$, $1/256$ yield similar results. Since N is relatively small, there is no specific choice in dependence of N required. Now, the steepest descent method iterates as follows:

$$x_{ij} = x_{ij} - \alpha_x\lambda_{N,x}(x_{ij}, y_{ij}), \tag{3}$$

$$y_{ij} = y_{ij} - \alpha_y\lambda_{N,y}(x_{ij}, y_{ij}), \tag{4}$$

with sufficiently small relaxation parameters α_x, α_y. In our numerical experiments, for $N \geqslant 6$, we made good experience with the choice

$$\alpha_x = 1/(N^2 - 26)\|\lambda_{N,x}\|, \tag{5}$$

$$\alpha_y = 1/(N^2 - 26)\|\lambda_{N,y}\|, \tag{6}$$

where $\|\cdot\|$ denotes the discrete L^2-norm. If the relaxation parameters are too small there is a very slow convergence. If they are too large some points leave the triangle. In our numerical experiments, we apply 1000 iterations of the steepest descent method followed by only a few iterations with a damped Newton method. Here, the main reduction of the Lebesque constant comes from the steepest descent method. The damped Newton method converges quite fast but only yields a slight reduction. For the Newton method second derivatives of the Lebesgue function are required which are once more approximated by central finite differences, i.e.,

$$\lambda_{N,xx}(x_{ij},y_{ij}) \cong (\lambda_N(x_{ij}-h,y_{ij}) - 2\lambda_N(x_{ij},y_{ij}) + \lambda_N(x_{ij}+h,y_{ij}))/h^2,$$
$$\lambda_{N,yy}(x_{ij},y_{ij}) \cong (\lambda_N(x_{ij},y_{ij}-h) - 2\lambda_N(x_{ij},y_{ij}) + \lambda_N(x_{ij},y_{ij}+h))/h^2,$$

with $h = 1/512$. Clearly, the Lebesque function is only piecewise smooth but the above finite difference schemes approximated the second derivatives well. Now, we are able to apply the decoupled Newton method to each component, i.e.,

Table 1
Lebesgue constants

N	Unsym.	Sym.	Taylor	CB L^2	Hesthaven	Ratio
6	3.67	3.87	4.17	3.79	4.07	1.11
9	5.58	5.89	6.80	5.92	6.92	1.19
12	7.12	7.59	8.03	10.08	12.52	1.11
15	8.41	9.25	9.97	–	29.69	1.17
18	10.08	11.86	13.5	–	–	1.31

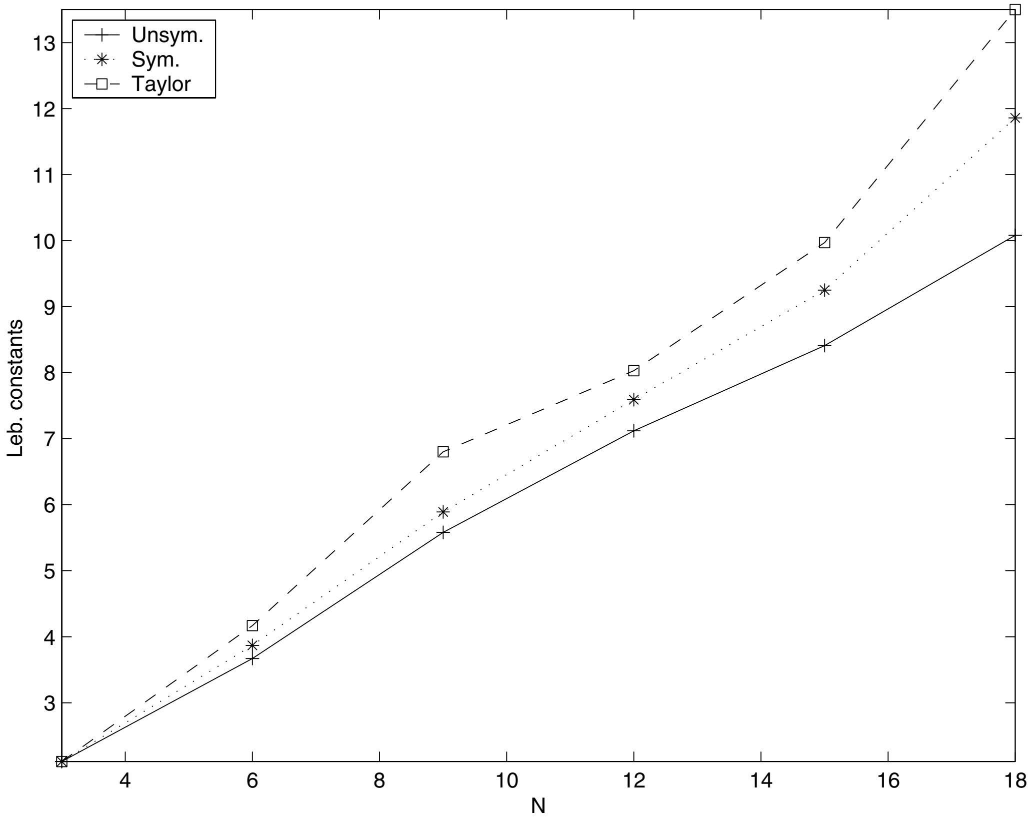

Fig. 1. Lebesgue constants.

$$x_{ij} = x_{ij} - \beta_x \lambda_{N,x}(x_{ij}, y_{ij}) / \lambda_{N,xx}(x_{ij}, y_{ij}), \tag{7}$$

$$y_{ij} = y_{ij} - \beta_y \lambda_{N,y}(x_{ij}, y_{ij}) / \lambda_{N,yy}(x_{ij}, y_{ij}), \tag{8}$$

where the parameters β_x, β_y are chosen in a suitable way. They are successively reduced until the Lebesgue constant becomes smaller. If the normalized residual reaches a lower bound of 10^{-5}, we stop the iteration. From numerical experiments, we observed a better performance of the above decoupled Newton approach instead of using the full 2D-Newton scheme.

4. Numerical results

In Table 1 and Fig. 1, we present our improved Lebesgue constants in compare to the results for the Fekete points obtained by Taylor et al. [11]. Also for small N, our Lebesque constants are better than Chen and Babuškas [2] optimal L^2-norm points and the points of Hesthaven [7]. Obviously, for increasing N the improvement becomes more significant. In Tables 6–8 we list our points for $N = 6, 9, 12$. Although the degrees of polynomials are too low for an asymptotic prediction it may be expected form Fig. 1 that the Lebesgue constants grow linearly in N. Our iteration does not preserve any symmetry. Hence we obtain unsymmetric distributions of points. This result is surprising and it hints that the actual Lebesque points

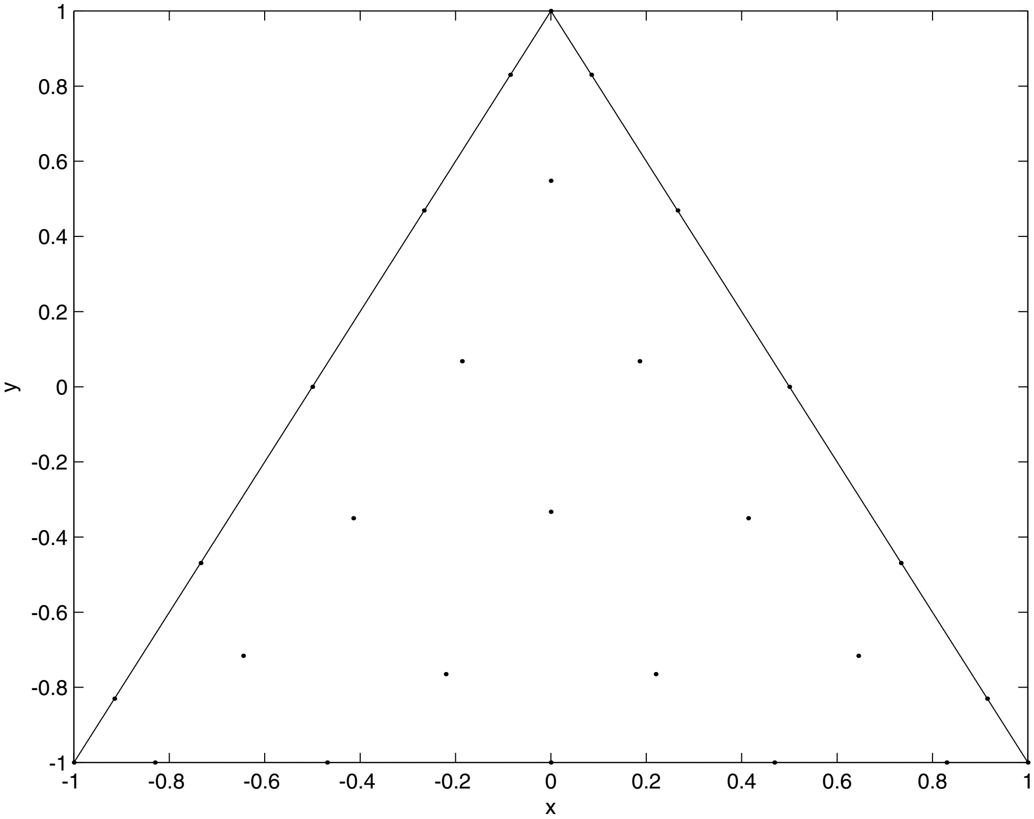

Fig. 2. Optimal points for $N = 6$.

may not contain any symmetry. But here we do not claim that. We are only able to state that our method yields unsymmetric distributions of nodes. This does not mean that the really optimal Lebesque nodes are unsymmetric. Furthermore, we calculated, the corresponding weights where we observed, that some weights belonging to the edge of the triangle become small but negative. Hence the stability properties of the schemes for numerical integration is doubtful. Here, we do not optimize on barycentric coordinates. But we were interested to see how symmetry conditions influence the Lebesque constant. For this purpose we started an optimization process on the triangle

$$T^* = \{(x,y): \ -1 \leqslant y \leqslant 2x+1, -1 \leqslant x \leqslant 0; \ -1 \leqslant y \leqslant -2x+1, 0 \leqslant x \leqslant 1\},$$

which is symmetric with respect to the axis $x = 0$. We started the above optimization procedure only on coordinates $x_{ij} \leqslant 0$ and afterwards set

$$(x_{N-i-jj}, y_{N-i-jj}) = (-x_{ij}, y_{ij}) \quad \text{for } i = 0, \ldots, (N-j)/2, \ \ j = 0, \ldots, N.$$

From Table 1 and Fig. 1, we observe that the corresponding Lebesgue constants are larger than for the unsymmetric treatment but smaller than for the Fekete points. In Figs. 2–6, we plotted our points on the triangle T* for $N = 6, 9, 12, 15, 18$. In general, it can be expected that more symmetry leads to worse Lebesgue constants. This makes sense in terms of a constrained optimization should do worse than an unconstrained optimization.

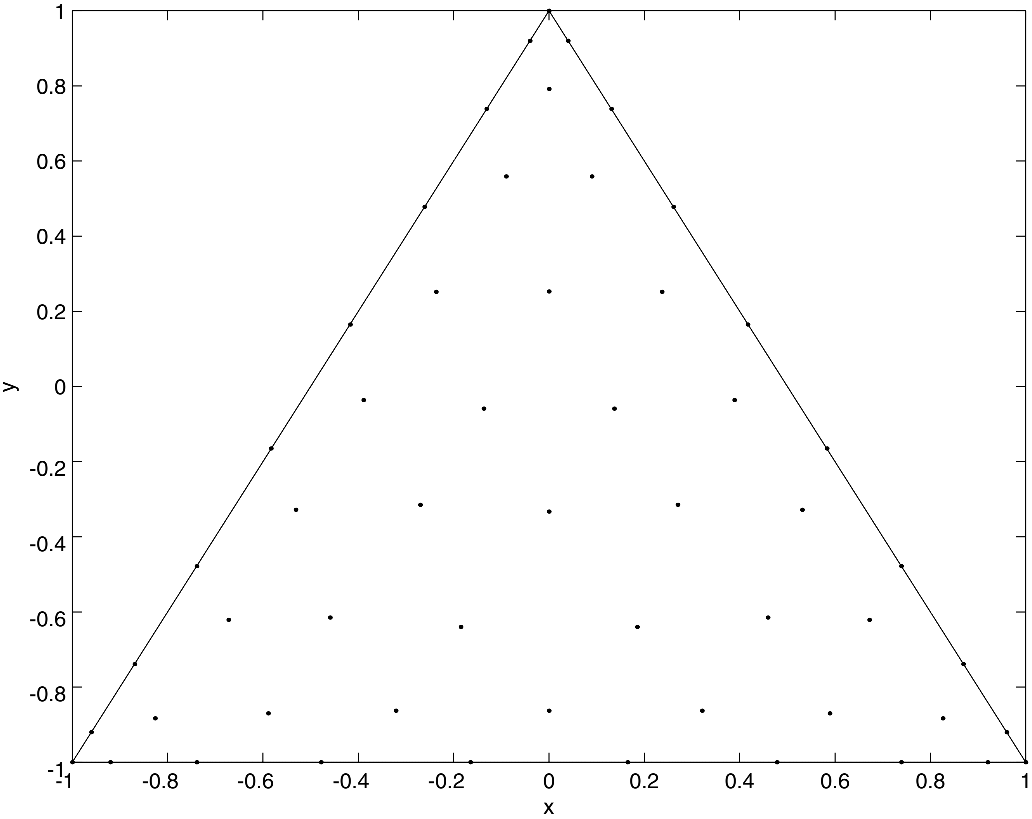

Fig. 3. Optimal points for $N = 9$.

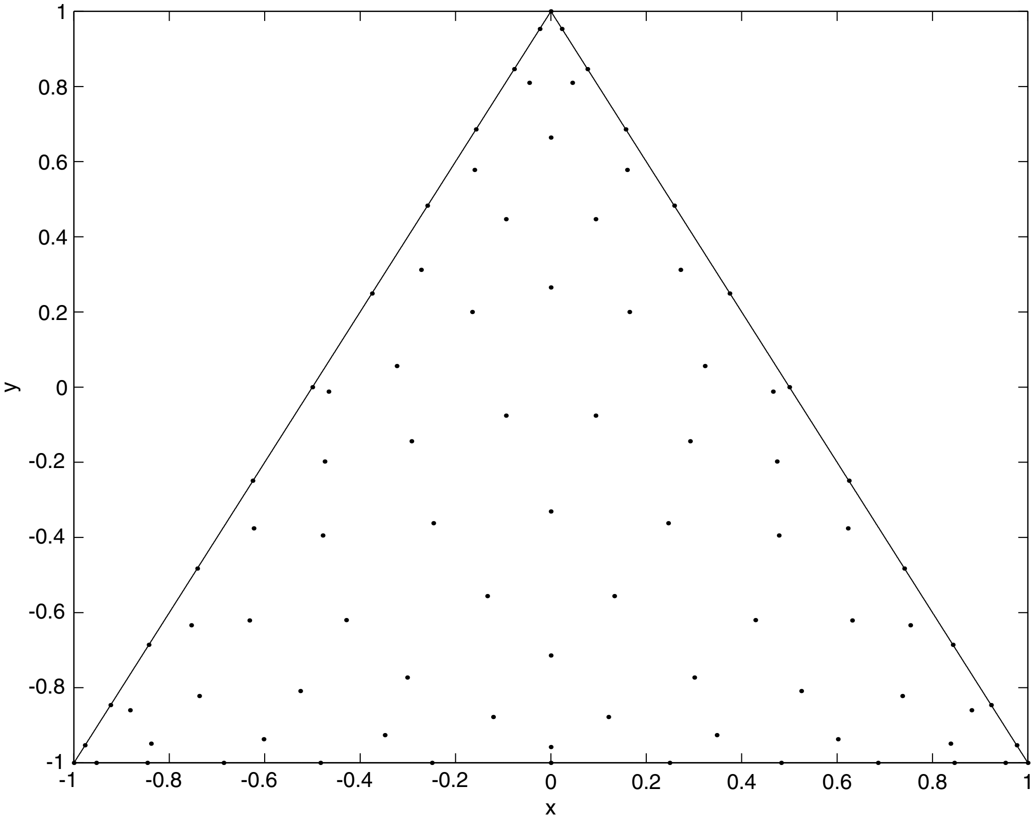

Fig. 4. Optimal points for $N = 12$.

Furthermore, it is interesting to know how sensitive the maximum is to the number of points or would it be better if the points were clustered near the edges. Here, we only consider the unsymmetric case. The maximum was calculated on a grid of $70 \times 71/2 = 2485$ equidistant points. In Table 2, we also give numerical results for $80 \times 81/2 = 3240$ and $90 \times 91/2 = 4095$ equidistant points. As it could be expected the maximum is increasing. This becomes more obvious if we take the maximum over a Chebyshev Gauss–Lobatto grid where the points are clustering near the three edges. We choose a Chebyshev grid with nodes (x_i, y_j) given by

$$y_j = \cos\left(j\frac{\pi}{M}\right), \quad j = 1, \ldots, M,$$
$$x_i = -\frac{1}{2}\cdot\left((y_j - 1)\cos\left((j-i)\frac{\pi}{j}\right) + y_j + 1\right), \quad i = 0, \ldots, j,$$

for $M = 69, 79, 89$. By adding the corner point $(-1, 1)$ we obtain grids with 2485, 3240 and 4095 Chebyshev points. From Table 2, we observe a strong increase in the maximum value which remains stable for increasing M. Obviously, the maximum is attained near the edges. A similar behavior can be observed for the Fekete points in Table 3. Furthermore, we checked the accuracy of the interpolation based on our unsymmetric nodes for two examples. They are given by

Example I

$$u(x,y) = (x+1)(y+1)(e^{x+y} - 1).$$

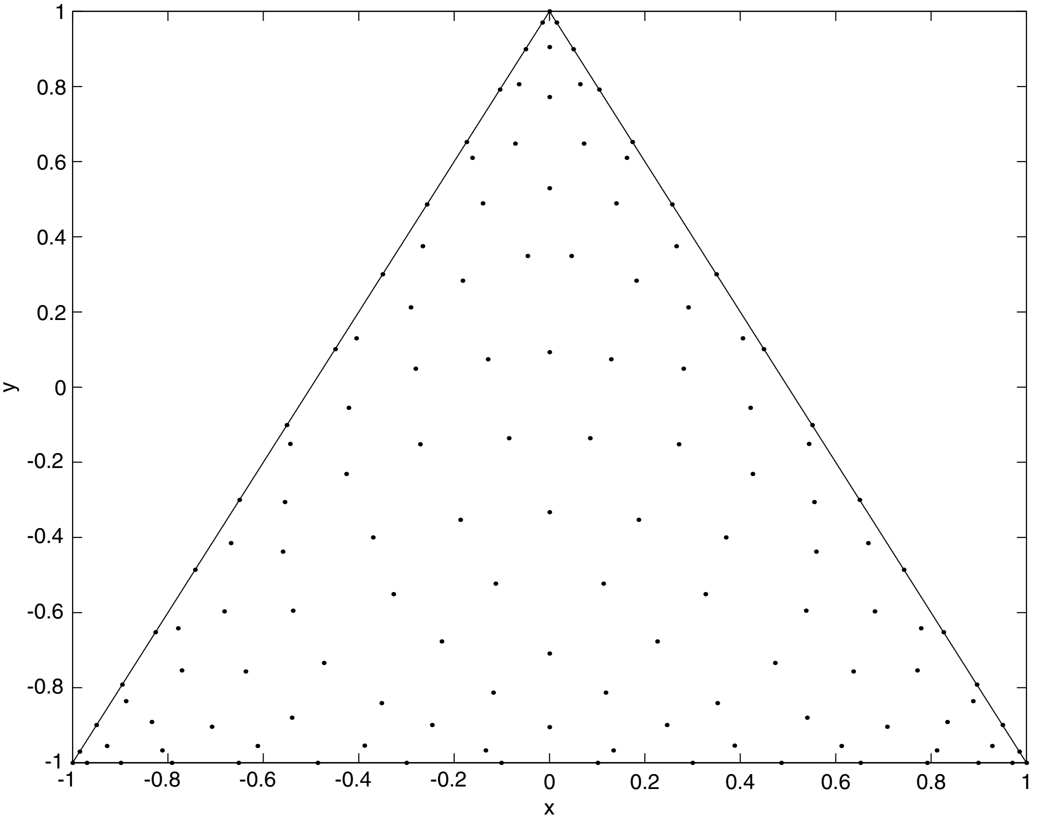

Fig. 5. Optimal points for $N = 15$.

Example II

$$u(x,y) = 10(x+1)(y+1)(\cosh(x+y)-1).$$

We calculated the discrete interpolation errors in the maximum norm over grids of 2485, 4095 equidistant points and 4095 Chebyshev points. In general, the errors are slightly increasing for increasing numbers of points or for the Chebyshev grid. From the numerical results in Tables 4 and 5 we observe a slight improvement by using our points. It is well known that the interpolation error can be estimated by the Lebesque constant as follows:

$$\|u - I_N u\|_\infty \leqslant (1 + \lambda_N)e_N(u),$$

where I_N denotes the interpolation operator and

$$e_N(u) = \inf\{\|u - p_N\|_\infty : p_N \in \mathbb{P}_N\}$$

the approximation error. Hence in order to compare error estimates for different nodes we have to calculate the ratio between $1 + \lambda_N$ for the Fekete points and our unsymmetric distribution of nodes. From Table 1 we obtain values in between 1.11 and 1.31. The ratio between the interpolation errors for the above examples with 4095 Chebyshev points is given in Tables 4 and 5 and it is even better than the behavior of the Lebesque constants. The improvement becomes obvious for increasing N with ratios of about 1.24–1.80.

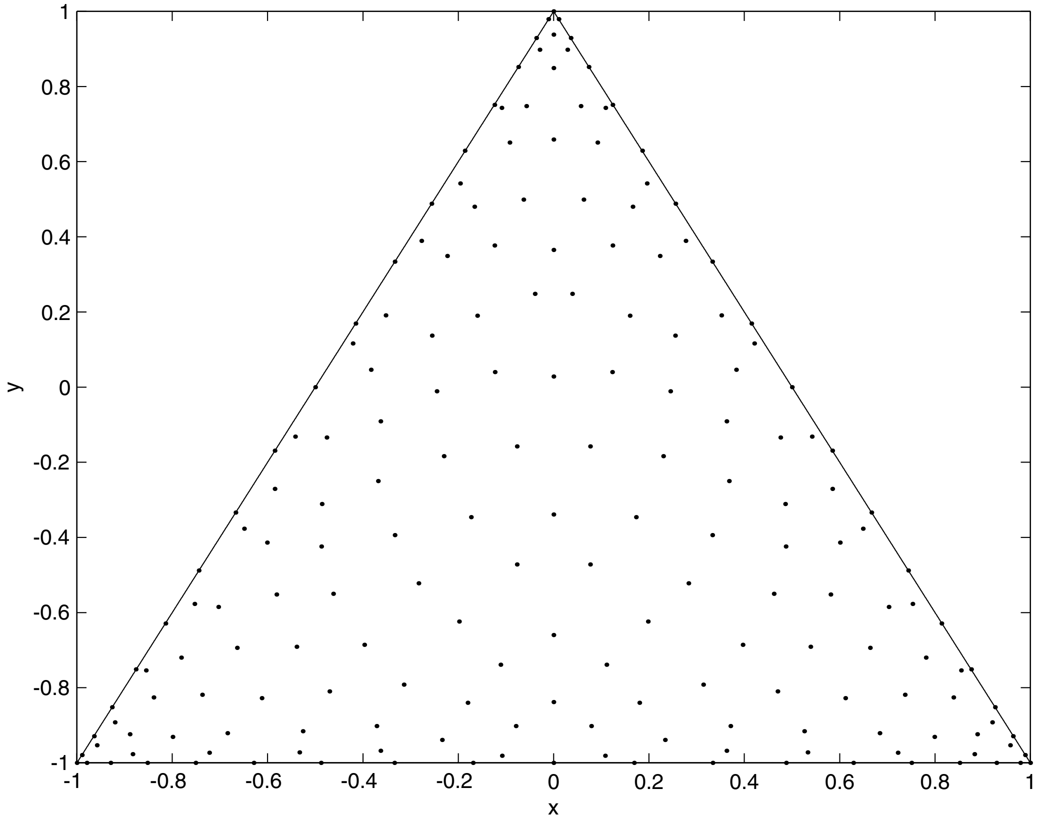

Fig. 6. Optimal points for $N = 18$.

Table 2
Lebesgue constants for unsym. points on equid. and Cheb. grids

N	2485 p.	3240 p.	4095 p.	2485 Ch.p.	3240 Ch.p.	4095 Ch.p.
6	3.67	3.69	3.68	3.69	3.69	3.68
9	5.58	5.58	5.60	5.59	5.59	5.58
12	7.12	7.41	7.49	7.42	7.48	7.50
15	8.41	8.71	8.83	8.73	8.82	8.73
18	10.08	11.76	12.77	13.86	13.48	13.78

Table 3
Lebesgue constants for Fekete points on equid. and Cheb. grids

N	2485 p.	3240 p.	4095 p.	2485 Ch.p.	3240 Ch.p.	4095 Ch.p.
6	4.10	4.17	4.16	4.15	4.17	4.16
9	6.80	6.80	6.77	6.77	6.79	6.80
12	8.03	9.64	9.61	9.61	9.57	9.67
15	9.97	9.91	9.98	9.93	9.98	9.90
18	13.5	13.76	14.26	14.66	14.69	14.71

Table 4
Interpolation errors for Example I

N	Present 2485 p.	Taylor 2485 p.	Present 4095 p.	Taylor 4095 p.	Present 4095 Ch.p.	Taylor 4095 Ch.p.	Ratio
12	1.64×10^{-12}	2.45×10^{-12}	1.70×10^{-12}	2.46×10^{-12}	1.72×10^{-12}	2.47×10^{-12}	1.44
15	5.72×10^{-15}	1.00×10^{-14}	6.08×10^{-15}	1.03×10^{-14}	6.25×10^{-15}	9.96×10^{-15}	1.60
18	1.28×10^{-14}	1.95×10^{-14}	1.29×10^{-14}	1.97×10^{-14}	1.28×10^{-14}	2.30×10^{-14}	1.80

Table 5
Interpolation errors for Example II

N	Present 2485 p.	Taylor 2485 p.	Present 4095 p.	Taylor 4095 p.	Present 4095 Ch.p.	Taylor 4095 Ch.p.	Ratio
12	4.00×10^{-11}	5.49×10^{-11}	4.32×10^{-11}	5.52×10^{-11}	4.37×10^{-11}	5.54×10^{-11}	1.27
15	2.96×10^{-14}	3.94×10^{-14}	2.96×10^{-14}	4.02×10^{-14}	3.09×10^{-14}	3.83×10^{-14}	1.24
18	7.71×10^{-14}	6.36×10^{-14}	6.33×10^{-14}	7.79×10^{-14}	6.28×10^{-14}	9.20×10^{-14}	1.46

Table 6
Points and weights for $N = 6$

x	y	Weights
−1.0000000000	−1.0000000000	−0.0048224886
−0.8302291554	−1.0000000000	0.0304687335
−0.4688697196	−1.0000000000	0.0269823678
0.0000000000	−1.0000000000	0.0457765688
0.4688697196	−1.0000000000	0.0280944010
0.8302291554	−1.0000000000	0.0237130559
1.0000000000	−1.0000000000	−0.0001337281
−1.0000000000	−0.8302291554	0.0274721838
−0.7652509351	−0.7504601980	0.1168568092
−0.3716634059	−0.7270784196	0.1827228637
0.0771571795	−0.7394530985	0.1715141862
0.5202206843	−0.7578421807	0.1358555904
0.8302291554	−0.8302291554	0.0196485280
−1.0000000000	−0.4688697196	0.0295124820
−0.7251686126	−0.3504561918	0.1744737376
−0.2920792639	−0.3304823387	0.1178571302
0.0728865230	−0.3557412342	0.1564972947
0.4688697196	−0.4688697196	0.0349523174
−1.0000000000	0.0000000000	0.0428160453
−0.7420679319	0.0841263110	0.1829659071
−0.3292750199	0.0612978608	0.1837683005
0.0000000000	0.0000000000	0.0433703389
−1.0000000000	0.4688697196	0.0270875177
−0.7663604173	0.5375895923	0.1278322343
−0.4688697196	0.4688697196	0.0303304059
−1.0000000000	0.8302291554	0.0240449990
−0.8302291554	0.8302291554	0.0220546481
−1.0000000000	1.0000000000	−0.0017124304

5. Conclusion

Starting with the Fekete points we obtain improved Lebesgue constants on the triangle. Here, a minimization procedure based on a combination of the steepest descent algorithm with a damped Newton method is employed. The optimal points along the boundary of the triangle are the 1D Gauss–Lobatto points. We

Table 7
Points and weights for $N = 9$

x	y	Weights
−1.0000000000	−1.0000000000	0.0027614993
−0.9195339860	−1.0000000000	0.0010745157
−0.7387741816	−1.0000000000	0.0114670460
−0.4779256080	−1.0000000000	0.0111138988
−0.1652794130	−1.0000000000	0.0121590502
0.1652794130	−1.0000000000	0.0146081034
0.4779256080	−1.0000000000	0.0084917022
0.7387741816	−1.0000000000	0.0102722987
0.9195339860	−1.0000000000	−0.0003797630
1.0000000000	−1.0000000000	0.0029760753
−1.0000000000	−0.9195339860	0.0004443573
−0.8970170978	−0.8823000088	0.0375669118
−0.6584034284	−0.8623617857	0.0514402802
−0.3628452596	−0.8694908590	0.0650879241
−0.0549679730	−0.8644977271	0.0619762636
0.2446985666	−0.8650788533	0.0610740162
0.5201404749	−0.8792433132	0.0510004553
0.7953853302	−0.8983757215	0.0373311414
0.9195339860	−0.9195339860	−0.0002724360
−1.0000000000	−0.7387741816	0.0086996743
−0.8716313408	−0.6710778092	0.0441245636
−0.6272738975	−0.6337635383	0.0747087146
−0.3426436967	−0.6356519405	0.0741150243
−0.0360875403	−0.6158559450	0.0833512298
0.2595718532	−0.6291250229	0.0699109786
0.5291180820	−0.6679094700	0.0526021623
0.7387741816	−0.7387741816	0.0104743918
−1.0000000000	−0.4779256080	0.0122166226
−0.8671529699	−0.3735076157	0.0720805916
−0.6307765647	−0.3351681337	0.0800342681
−0.3365578062	−0.3478376149	0.1033485060
−0.0193134591	−0.3437245259	0.0829288036
0.2664760896	−0.3947759410	0.0595823395
0.4779256080	−0.4779256080	0.0104880205
−1.0000000000	−0.1652794130	0.0109371107
−0.8661309341	−0.0615062096	0.0567665379
−0.6121111450	−0.0430769119	0.0774924035
−0.3483137871	−0.0182856034	0.0865623953
−0.0673716394	−0.0612490064	0.0689137970
0.1652794130	−0.1652794130	0.0132626678
−1.0000000000	0.1652794130	0.0156110254
−0.8659060922	0.2374911740	0.0667956173
−0.6330776864	0.2504635954	0.0694730827
−0.3942499421	0.2623631727	0.0596988470
−0.1652794130	0.1652794130	0.0118984367
−1.0000000000	0.4779256080	0.0079094812
−0.8660518130	0.5156355674	0.0506963102
−0.6689999522	0.5391728608	0.0529739398
−0.4779256080	0.4779256080	0.0120256444
−1.0000000000	0.7387741816	0.0127695609
−0.8966337273	0.7892172212	0.0358131791
−0.7387741816	0.7387741816	0.0078155931
−1.0000000000	0.9195339860	−0.0010827624
−0.9195339860	0.9195339860	0.0021333781
−1.0000000000	1.0000000000	0.0026745228

Table 8
Points and weights for $N = 12$

x	y	Weights
−1.0000000000	−1.0000000000	−0.0017352022
−0.9533098466	−1.0000000000	0.0033358747
−0.8463475646	−1.0000000000	0.0023824146
−0.6861884690	−1.0000000000	0.0023751816
−0.4829098210	−1.0000000000	0.0063893073
−0.2492869302	−1.0000000000	0.0035878762
0.0000000000	−1.0000000000	0.0054374279
0.2492869302	−1.0000000000	0.0060406611
0.4829098210	−1.0000000000	0.0026395147
0.6861884690	−1.0000000000	0.0053274724
0.8463475646	−1.0000000000	0.0010554009
0.9533098466	−1.0000000000	0.0033309399
1.0000000000	−1.0000000000	−0.0010571742
−1.0000000000	−0.9533098466	0.0047790873
−0.9219009766	−0.9201052124	0.0127137502
−0.7492410600	−0.9399006992	0.0215408918
−0.5103575968	−0.9403499904	0.0205159325
−0.4321198360	−0.8465677488	0.0339843788
−0.1880301145	−0.9337926179	0.0344257426
0.0964351963	−0.9229969309	0.0314638418
0.2778805235	−0.8459161672	0.0330129128
0.4031052378	−0.9439546054	0.0243542916
0.6856149142	−0.9425112148	0.0190415648
0.8436694782	−0.9234065040	0.0148735392
0.9533098466	−0.9533098466	0.0030533373
−1.0000000000	−0.8463475646	0.0002372244
−0.9483662493	−0.7345120001	0.0251087886
−0.8145728953	−0.8239213998	0.0249062914
−0.6394373688	−0.7997248691	0.0318890695
−0.1809726292	−0.7959761078	0.0380507052
−0.2155764744	−0.6143967746	0.0587582828
0.0137758327	−0.7496793837	0.0429848851
0.4746810969	−0.8275505928	0.0259093947
0.6223721207	−0.8264041680	0.0246608518
0.6948621890	−0.7530961006	0.0223996090
0.8463475646	−0.8463475646	0.0014624224
−1.0000000000	−0.6861884690	0.0026756561
−0.9556648041	−0.4646982593	0.0206122661
−0.8189960092	−0.6534744389	0.0245403089
−0.6960274740	−0.5390285091	0.0454742347
−0.4934232404	−0.6837532870	0.0428011926
0.2254956916	−0.6710567221	0.0431631723
0.1853242648	−0.5116174273	0.0427033385
0.4433755625	−0.5040138672	0.0264761380
0.4533492441	−0.6484211924	0.0317617345
0.6861884690	−0.6861884690	0.0034281861
−1.0000000000	−0.4829098210	0.0020434612
−0.8400054990	−0.4430292346	0.0389583911
−0.7542150701	−0.2365297128	0.0268033578
−0.4916084278	−0.4810104380	0.0624877852
−0.3304295394	−0.3310059092	0.0684694209
−0.0352531069	−0.4898915021	0.0583082123
0.0147568977	−0.2122611402	0.0385977876

Table 8 (*continued*)

x	y	Weights
0.2742071061	−0.4383085383	0.0285406100
0.4829098210	−0.4829098210	0.0030938709
−1.0000000000	−0.2492869302	0.0062394831
−0.9231919765	−0.1756908478	0.0363318914
−0.6799468025	0.1823935177	0.0445578163
−0.6215028749	−0.1769644322	0.0666397774
−0.5368826189	0.2234000013	0.0439960327
−0.1839259335	−0.1964618180	0.0662082809
0.1213585137	−0.1957914645	0.0311912442
0.2492869302	−0.2492869302	0.0068061152
−1.0000000000	0.0000000000	0.0037723120
−0.9284891809	0.1198691438	0.0344264079
−0.7867341316	−0.0479808067	0.0391854516
−0.4741667003	−0.0212720727	0.0578816516
−0.2344937875	0.0045453667	0.0343812742
−0.1792246758	0.1010930397	0.0351868559
0.0000000000	0.0000000000	0.0054066609
−1.0000000000	0.2492869302	0.0066156969
−0.9449945881	0.4482106621	0.0198623047
−0.8437483687	0.2971709638	0.0393975312
−0.4691753462	0.4203502209	0.0235415032
−0.4300368131	0.2722874703	0.0318177891
−0.2492869302	0.2492869302	0.0058791198
−1.0000000000	0.4829098210	0.0034088422
−0.8163193352	0.6408019800	0.0268674294
−0.7992366134	0.4449727193	0.0260154023
−0.6434964587	0.4652495937	0.0300989243
−0.4829098210	0.4829098210	0.0018301766
−1.0000000000	0.6861884690	0.0040370853
−0.9390291177	0.6811798646	0.0226135477
−0.7426299536	0.6901548707	0.0203013160
−0.6861884690	0.6861884690	0.0042070412
−1.0000000000	0.8463475646	0.0012022591
−0.9271329058	0.8502700028	0.0130808975
−0.8463475646	0.8463475646	0.0009141136
−1.0000000000	0.9533098466	0.0035984890
−0.9533098466	0.9533098466	0.0039075341
−1.0000000000	1.0000000000	−0.0015838735

obtain both symmetric and unsymmetric distributions of nodes. Some quadrature weights (only belonging to the edge of the triangle) become negative and hence the usefulness of our approach for numerical quadrature is doubtful. But for all degrees our points yield the smallest Lebesgue constants currently known. Numerical examples are presented which demonstrate the improved interpolation properties.

References

[1] L. Bos, Bounding the Lebesgue function for Lagrange interpolation in a simplex, J. Approx. Theory 38 (1983) 43–59.

[2] Q. Chen, I. Babuška, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Eng. 128 (1995) 405–417.

[3] M.O. Deville, P.F. Fischer, E.H. Mund, High-order Methods for Incompressible Fluid Flow, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2002.

[4] M. Dubiner, Spectral methods on triangles and other domains, J. Sci. Comput. 6 (1993) 345–390.

[5] W. Heinrichs, Spectral collocation on triangular elements, J. Comput. Phys. 145 (1998) 743–757.

[6] W. Heinrichs, B. Loch, Spectral schemes on triangular elements, J. Comput. Phys. 173 (2001) 279–301.

[7] J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation, SIAM J. Numer. Anal. 35 (1998) 655–676.

[8] G. Karniadakis, S.J. Sherwin, Spectral/hp Element Methods for CFD Numerical Mathematics and Computation, Oxford University Press, London, 1999.

[9] C. Mavriplis, A. Van Rosendale, Triangular spectral elements for incompressible fluid flow, ICASE Rep. (1993) 93–100.

[10] R. Pasquetti, F. Rapetti, Spectral element methods on triangles and quadrilaterals: comparisons and applications, J. Comput. Phys. 198 (2004) 349–362.

[11] M.A. Taylor, B.A. Wingate, R.E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal. 38 (2000) 1707–1720.