ELSEVIER

Journal of Computational and Applied Mathematics 72 (1996) 41–62

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

OPTAC: a portable software package for analyzing and comparing optimization methods by visualization

G.S. Androulakis *, M.N. Vrahatis

Department of Mathematics, University of Patras, GR-261.10 Patras, Greece

Received 30 July 1994; revised 27 October 1995

Abstract

A software package for analyzing and comparing optimization methods is presented. This package displays, using different colors, the regions of convergence to the minima of a given function for various optimization methods. It displays also the rate of their convergence as well as the regions of divergence of these methods. Moreover, this package gives quantitative information regarding the total convergence area in a specific domain for various minima.

Using OPTAC (OPTimization Analysis and Comparisons) we are able to "see" in a picture the advantages and disadvantages of any optimization method as well as to compare various methods in order to choose the proper method for a given class of problems. The OPTAC package is self-contained and conforms to the ANSI 1977 Fortran standards.

Keywords: Analyzing and comparing optimization methods; Software package; Algorithms analysis

AMS classification: 65 K10; 49 M37; 90C31; 68 N99

1. Introduction

There is a large variety of optimization algorithms for computing the extrema of a function:

$$f: \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}, \tag{1.1}$$

where $\mathbb{R}^n$ indicates the real Euclidean n-space.

These algorithms possess advantages and disadvantages and it is not always evident which one is proper for a given class of applications. Three principal problems may be associated with any iterative process for minimizing (1.1), namely:

(1) Is the algorithm well-defined, that is, can it be continued to a satisfactory end?

* Corresponding author. e-mail: gsa@math.upatras.gr.

SSDI 0377-0427(95)00244-8

(2) Do the iterates converge to a minimum?
(3) How economical is the entire operation?

In most instances only partial answers can be given to the above questions. Question (2) consists of three parts:

(2a) the convergence of the iterates to a limit (convergence or divergence),
(2b) the question of whether this limit is in fact a minimum and
(2c) the question of whether for a given initial guess the algorithm converges to the closest minimum.

Question (3) concerns the computational complexity of the algorithm and it includes the following subquestions:

(3a) What is the *cost* of any one step of the iterative process?
(3b) How *fast* does the sequence converge?
(3c) How *sensitive (stable)* is the process to changes of the operator or the initial data?

In this contribution we try to answer the above questions by providing experimental data concerning (2c), (3a) and (3c). The other questions can usually be addressed more satisfactorily by theory. To this end we give a software package for analyzing and visualizing the convergence behavior of optimization algorithms. This package displays, using different colors, the regions of convergence to the minima of a given function. It displays also the rate of their convergence as well as the regions of divergence of these methods. Moreover, this package gives quantitative measures for the terms "cost", "fast" and "sensitive". Using OPTAC we are able to "see" in a picture the advantages and disadvantages of any optimization method as well as to compare various methods in order to choose the proper one for a given class of problems.

In the next section we briefly present the algorithms which we shall analyze and compare. In Section 3, we outline how OPTAC accumulates information in a picture. A description of the software as well as an example of OPTAC usage are presented in Section 4. We finally end, in Sections 5 and 6 with some applications, a discussion for extracting information from an OPTAC picture and some concluding remarks.

2. The algorithms

In this section we shall briefly describe the algorithms which we shall analyze and compare here. The reason for this choice is that the following algorithms are well known and widely used. Our package, however, can be applied to any optimization method.

(a) *Armijo's modified steepest descent algorithm*

A well-known method in the class of steepest descent methods [3] for unconstrained minimization of functions (1.1) is *Armijo's method* or *the modified steepest descent algorithm* [1, 11]. In fact this method is a modification of the renowned Cauchy's method [3] and can be defined by the following sequence:

$$x^{k+1} = x^k - \eta_{m_k} \nabla f(x^k), \quad k = 0, 1, 2, \dots,$$

where $\nabla f(x)$ is the gradient of f at x, $\eta_m = \eta/2^{m-1}$, $m = 1, 2, \dots$, with η an arbitrary assigned positive number, usually taken to be 1, and m_k is the smallest positive integer for which

$$f\left(x^k - \eta_{m_k} \nabla f(x^k)\right) - f(x^k) \leqslant -\tfrac{1}{2}\eta_{m_k} \|\nabla f(x^k)\|^2.$$

(b) *Nonlinear conjugate gradient algorithms*

There is a class of methods called *nonlinear conjugate gradient methods*, as typified by the *Fletcher–Reeves* (FR) *algorithm* [5, 8] and the closely related *Polak–Ribiere* (PR) *algorithm* [5, 8]. For these methods the rate of their convergence for general nonlinear problems is linear [7, 5]. Conjugate gradient methods require storage of order only a few times n. On the other hand, these methods require derivative calculations as well as one-dimensional sub-minimization and they are very sensitive to rounding off errors.

The most known methods in this class are expressed by:

$$x^{k+1} = x^k + \alpha_k p^k, \quad k = 0, 1, 2, \ldots, \quad \text{where } \alpha_k \text{ minimizes } f(x^k + \alpha_k p^k),$$
$$p^0 = -\nabla f(x^0),$$
$$p^k = -\nabla f(x^k) + \beta^k p^{k-1},$$
$$\beta^k = \frac{\left(\nabla f(x^k) - \mu \nabla f(x^{k-1})\right)^{\mathrm{T}} \nabla f(x^k)}{\nabla f(x^{k-1})^{\mathrm{T}} \nabla f(x^{k-1})},$$

where for $\mu = 0$ we have the Fletcher–Reeves (FR) method [5, 3, 8] and for $\mu = 1$ we obtain the Polak–Ribiere (PR) method [5, 3, 8].

(c) *Variable metric methods*

Another efficient class of methods is known under the names *quasi-Newton* and *variable metric methods*, as typified by the *Davidon–Fletcher–Powell* (DFP) *algorithm* [5, 3, 8] or the closely related *Broyden– Fletcher–Goldfarb–Shanno* (BFGS) *algorithm*. The above-mentioned methods are very stable and they converge superlinearly [7]. On the other hand, these methods require storage of order n^2 and they require derivative calculations. Besides, they need one-dimensional sub-minimization.

The most known methods in this class are expressed by:

$$x^{k+1} = x^k - \alpha_k B_k \nabla f(x^k), \quad k = 0, 1, 2, \ldots,$$
$$B_{k+1} = B_k + \frac{r^k (r^k)^{\mathrm{T}}}{(r^k)^{\mathrm{T}} q^k} - \frac{B_k q^k (q^k)^{\mathrm{T}} B_k}{(q^k)^{\mathrm{T}} B_k q^k} + \mu (q^k)^{\mathrm{T}} B_k q^k u^k (u^k)^{\mathrm{T}},$$
$$u^k = \frac{r^k}{(r^k)^{\mathrm{T}} q^k} - \frac{B_k q^k}{(q^k)^{\mathrm{T}} B_k q^k},$$
$$r^k = x^{k+1} - x^k, \qquad q^k = \nabla f(x^{k+1}) - \nabla f(x^k),$$

where B_0 is an arbitrary symmetric and positive definite matrix, usually taken to be the identity matrix, and α_k is the optimal length in the direction $-B_k \nabla f(x^k)$. Now, for $\mu = 0$ we obtain the Davidon–Fletcher–Powell (DFP) method [5, 3, 8], while for $\mu = 1$ we get the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [5, 3, 8].

3. Accumulating information in an OPTAC picture

In this section we describe how the OPTAC package accumulates information in a picture.

OPTAC distinguishes the minima of the objective function by marking them with different colors. Each picture's element (pixel) corresponds to an initial guess of the analyzed algorithm and it is colored according to its convergence. Specifically, each one of them takes the color of the minimum to which the corresponding method converges, while, if for this initial guess the relative algorithm does not converge, after a specific number of iterations, this element is colored white. Thus, if a picture has for instance a fractal-like structure, then the respective algorithm is not stable since it is sensitive to small perturbations of the starting points.

In the two-dimensional case we take a finite domain of starting points by taking a dense grid of the two unknowns which determine the coordinates of a pixel. In higher dimensions we take a finite domain of starting points of the two-dimensional subspace $\mathcal{E}^2$ of $\mathbb{R}^n$ spanned by $\{e^{\max}, e^{\min}\}$, where $e^{\max}$, $e^{\min}$ are the eigenvectors corresponding to the extreme eigenvalues of the Hessian of f at a minimum x^*, $\nabla^2 f(x^*)$. More specifically, we apply the algorithms for points

$$x = x^* + y \in \mathbb{R}^n, \quad \text{with} \quad y = c_1 e^{\max} + c_2 e^{\min}, \quad c_1, c_2 \in \mathbb{R}, \tag{3.1}$$

by taking a grid of the values of c_1 and c_2 which determine the coordinates of an OPTAC pixel. Now, since $\nabla^2 f$ is real and symmetric all eigenvalues and eigenvectors are real and the eigenvectors corresponding to distinct eigenvalues are orthogonal. Thus, the values of c_1 and c_2 taken by a grid into an orthogonal parallelepiped represent an orthogonal parallelepiped subset of $\mathcal{E}^2$.

The reason for the choice of this two-dimensional subspace is that it reveals useful information explained in the sequel. Studying the sensitivity of the solution to small changes, it is known that, in a sufficiently small neighborhood of x^*, the directions of the principal axes of the elliptical contours (n-dimensional ellipsoids) will be given by the eigenvectors of $\nabla^2 f$, while the lengths of the axes will be inversely proportional to the square roots of the corresponding eigenvalues. Thus, a variation along the eigenvector corresponding to the maximum eigenvalue will cause the largest change in f, while the eigenvector corresponding to the minimum eigenvalue gives the least sensitive direction. Thus, to study the behavior of optimization methods for various directions, including the "extreme ones", we apply the corresponding algorithms for starting values given by relation (3.1).

Of course, it is not necessary for OPTAC to know the coordinates of the minima beforehand, since it applies an algorithm for any finite domain of starting points and according to its convergence it saves the coordinates of the minima. The first minimum is colored Green, the second Red, the third Cyan, the fourth Gray and so on.

Furthermore, in order to display the rate of convergence (how fast the algorithm converges to a minimum for the specific initial point) we utilize color shades. The dark colors indicate rapid convergence while the light ones indicate slow convergence. To this end we utilize eight different shades per color. Thus, reading a picture, we are able to see the regions of rapid convergence. Moreover, by observing that the colored zones are separated we can easily answer the question of whether a minimum attracts the initial guesses close to it or not. This we call *method's reliability*.

Also, our package counts the *radius of convergence* of the algorithm to a specific minimum. The radius of convergence is the longest possible radius of the circle centered at the minimum which contains only starting points which converge to this minimum. OPTAC counts the radii of convergence for all the minima contained in the shot box. Of course, these results depend on the resolution of the screen. But the user can magnify various regions to verify his/her results.

Now, regarding the computational complexity of an algorithm the OPTAC performs quantitative measurements and gives the average of the number of iterations as well as the average of the number of function evaluations. Moreover, our package counts the total number of the initial guesses which converge to the closest minimum. Thus, estimating the above percentage we are able to declare the reliability of the algorithm.

4. Description of the software

The OPTAC package consists of a set of three subroutines, VIEW, FCOLZON and STATS, one of which is called by the user driver program. The user must only call the subroutine VIEW; all the other subroutines are evoked by VIEW.

The purpose of VIEW is to display the pieces of information obtained by the subroutines FCOLZON and STATS. Specifically, VIEW creates the pictures which give information, utilizing different colors, regarding the various minima and the rate of convergence to a minimum utilizing shades of the color of the shot minimum.

The subroutine FCOLZON (1) reads the data file MAINFILE.DAT which contains the coordinates of the initial points, the total number of the iterations required and the total number of the function evaluations required, (2) stores the number and the coordinates of the minima found in the shot domain, (3) computes all the radii of convergence, (4) calculates the total number of the initial guesses which converge to the closest minimum and computes the corresponding percentage, (5) scales the colors and shades relative to maximum and minimum values of the displayed information and (6) calculates the average of the total number of the iterations required for each starting point as well as the average of the total number of function evaluations required. The subroutine STATS displays the quantitative information which is computed by the subroutine FCOLZON.

OPTAC is evoked by the following FORTRAN statement:

CALL VIEW(n, m, armin, box, res, mainfile, rstat, reliab, totit, totfe, radii)

where:

- **n** is a positive integer input variable that defines the number of variables.
- **m** is a positive input variable that defines the maximum number of minima required. On output it determines the number of minima found.
- **armin** is an **m**×**n** matrix which in each row contains the coordinates of a minimum.
- **box** is the 2×2 matrix with elements $[a_{ij}]$ to specify the rectangular domain $[a_{11}, a_{21}] \times [a_{12}, a_{22}]$ of starting points.
- **res** is an input array of length 2 which contains the resolution of the monitor card (320 200 for VGA card with 256 colors).
- **mainfile** is the name of the above-described input data file MAINFILE.DAT.
- **rstat** is an output array of length (**m**×8) that specifies the total number of the initial guesses per shade and per color zone.
- **reliab** is a real output variable that specifies the percentage of the total number of the initial guesses which converge to the closest minimum.

totit is a real output variable that specifies the average of the total number of the iterations required.

totfe is a real output variable that specifies the average of the total number of the function evaluations required.

radii is an output array of length **m** which contains the radii of convergence to the minima contained in **box**.

The OPTAC package is self-contained and conforms to the ANSI 1977 Fortran standards. It contains about 1000 lines of code, 45% of which are comments. The total storage required for OPTAC is $m(13+n)+n+60$ locations, where m specifies the maximum number of minima required and n determines the dimension of the problem. OPTAC was tested on a PC IBM compatible with a variety of known problems. For the graphics portion of our package we have utilized the Microsoft Fortran 5.00 utilities FGRAPH.FD, FGRAPH.FI and the library GRAPHICS.LIB. The package is compatible with all the subsequent versions of the Microsoft Fortran for MS-DOS and Windows.

5. Applications

We have applied OPTAC to various well-known tested functions of various dimensions and have revealed useful information regarding the geometry of basins of attraction as well as quantitative measures for the terms "cost", "fast" and "sensitive".

The convergence criteria for all the algorithms were:

(i) $|f(x^{k+1}) - f(x^k)| \leqslant 10^{-8}$,

(ii) $\|\nabla f(x^{k+1})\|_2 \leqslant 10^{-4}$,

(iii) MIT = 2000, where MIT is the maximum number of iterations.

Here we exhibit OPTAC pictures and quantitative results for the following applications.

Application 5.1. *Complex function* ($z^3 - 1 = 0$), $n = 2$ [8]:

$$f(x_1,x_2) = (x_1^3 - 3x_1x_2^2 - 1)^2 + (3x_1^2x_2 - x_2^3)^2. \tag{5.1}$$

This function has three minima $x_1^* = (1,0)$, $x_2^* = (-\frac{1}{2}, \sqrt{3}/2)$, and $x_3^* = (-\frac{1}{2}, -\sqrt{3}/2)$, with $f(x_i^*) = 0$, $i = 1,2,3$.

Application 5.2. *Stenger function*, $n = 2$ [9]:

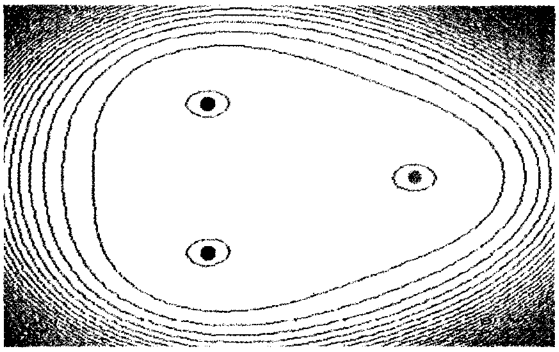
$$f(x_1,x_2) = (x_1^2 - 4x_2)^2 + (x_2^2 - 2x_1 + 4x_2)^2. \tag{5.2}$$

This function has two minima $x_1^* = (0,0)$ and $x_2^* = (1.695415, 0.7186082)$ with $f(x_i^*) = 0$, $i = 1,2$.

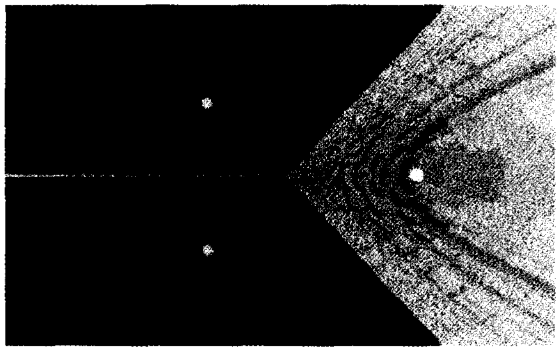
Application 5.3. *Himmelblau function*, $n = 2$ [2]:

$$f(x_1,x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2. \tag{5.3}$$

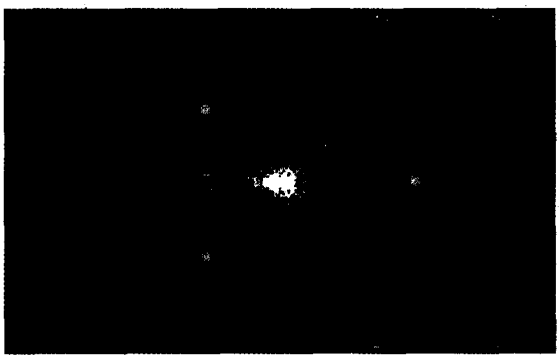
This function has four minima $x_1^* = (3, 2)$, $x_2^* = (-2.805118, 3.131312)$, $x_3^* = (3.584428, -1.848126)$ and $x_4^* = (-3.779310, -3.283186)$ with $f(x_i^*) = 0$, $i = 1,2,3,4$.



Contour Lines

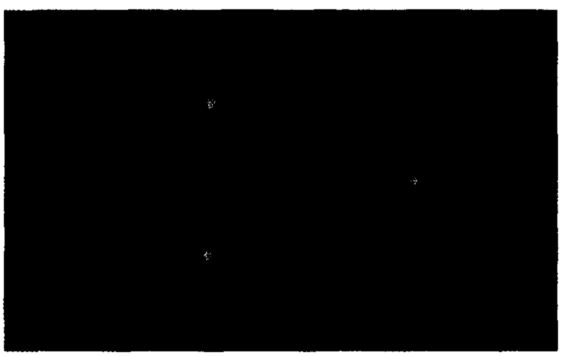


Armijo

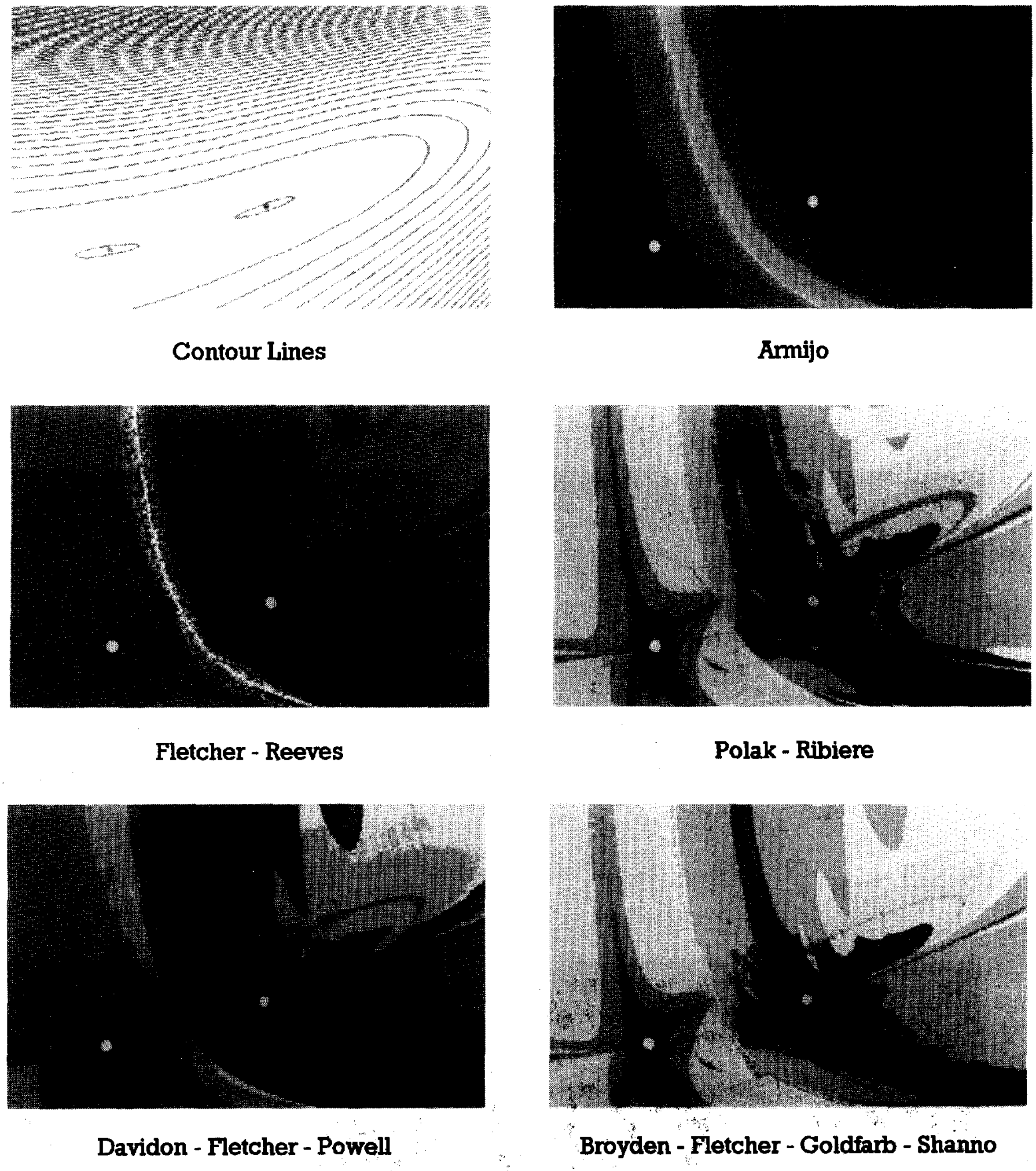


Fletcher - Reeves

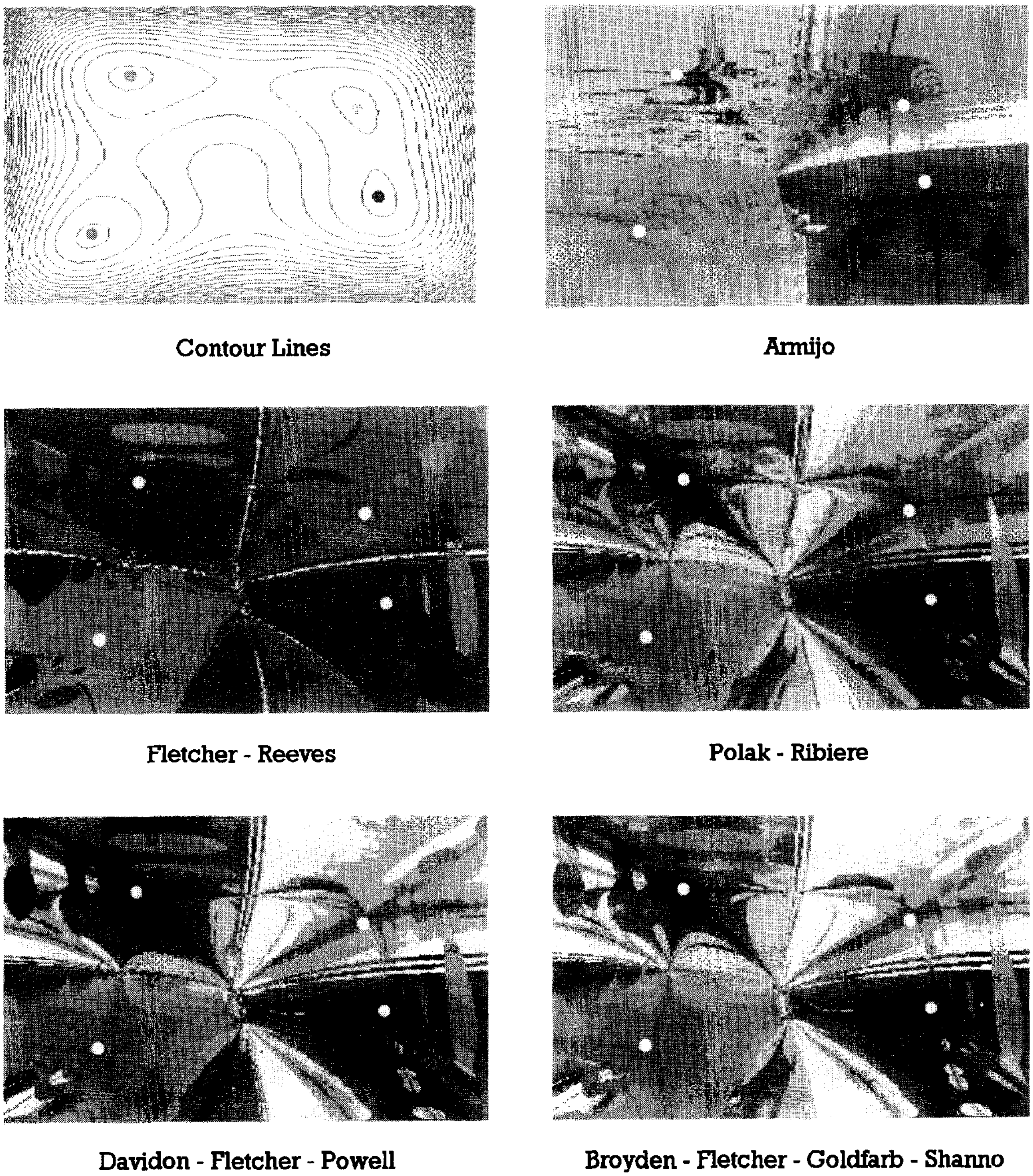
Polak - Ribiere



Application 5.1. Box $[-2, 2] \times [-2, 2]$.



Application 5.2. Box $[-1, 4] \times [-1, 4]$.



Application 5.3. Box $[-6, 6] \times [-6, 6]$.

Application 5.4. *Helical valley function*, $n=3$ [6]:

$$f(x_1,x_2,x_3)=100(x_3-10\theta(x_1,x_2))^2+100(\sqrt{x_1^2+x_2^2}-1)^2+x_3^2, \tag{5.4}$$

where

$$\theta(x_1,x_2)=\begin{cases}\dfrac{1}{2\pi}\arctan\left(\dfrac{x_2}{x_1}\right), & x_1>0,\\[2mm] \dfrac{1}{2\pi}\arctan\left(\dfrac{x_2}{x_1}\right)+0.5, & x_1<0.\end{cases} \tag{5.5}$$

This function has one minimum $x^*=(1,0,0)$ with $f(x^*)=0$. The eigenvalues of the $\nabla^2 f$ at x^* are $\lambda=(1.432763432230859,\ 200,\ 707.173154779458)$. The normalized eigenvector for the minimum eigenvalue, $e^{\min}$, and for the maximum, $e^{\max}$, are

$$e^{\min}=\begin{pmatrix}0.0000000000000000\\0.5330985587889368\\0.8460531464495351\end{pmatrix},\qquad e^{\max}=\begin{pmatrix}0.0000000000000000\\0.8460531464495351\\0.5330985587889368\end{pmatrix}.$$

Application 5.5. *Brown almost-linear function*, $n=3$ and $n=4$ [6]:

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{n} f_i^2(x_1,\ldots x_n), \tag{5.6}$$

where

$$f_i(x_1,\ldots,x_n)=x_i+\sum_{j=1}^{n}x_j-(n+1),\quad i=1,\ldots,n-1,$$

$$f_n(x_1,\ldots,x_n)=\left(\prod_{j=1}^{n}x_j\right)-1.$$

The minima of this function are $(a,\ldots,a^{1-n})$ where a satisfies the equation

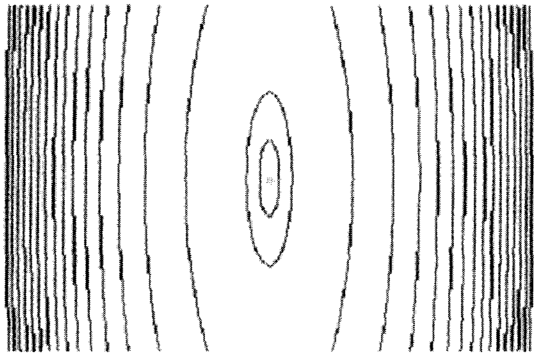
$$na^n-(n+1)a^{n-1}+1=0.$$

For $n=3$ the eigenvalues of the $\nabla^2 f$ at the considered minimum $x^*=(1,1,1)$ are $\lambda=(14-8\sqrt{3},\ 2, 14+8\sqrt{3})$. The normalized eigenvectors $e^{\min}$ and $e^{\max}$ are

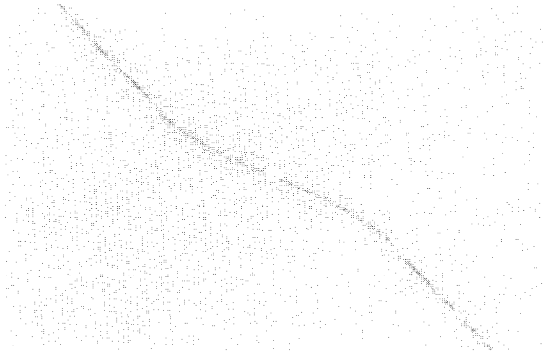
$$e^{\min}=\begin{pmatrix}-0.3250575836718656\\-0.3250575836718656\\0.8880738339771151\end{pmatrix},\qquad e^{\max}=\begin{pmatrix}0.6279630301995545\\0.6279630301995545\\0.4597008433809832\end{pmatrix}.$$

For $n=4$ the eigenvalues of the $\nabla^2 f$ at the considered minimum $x^*=(1,1,1,1)$ are $\lambda=(23-5\sqrt{21},\ 2,2,23+5\sqrt{3})$ while the normalized eigenvectors $e^{\min}$ and $e^{\max}$ are

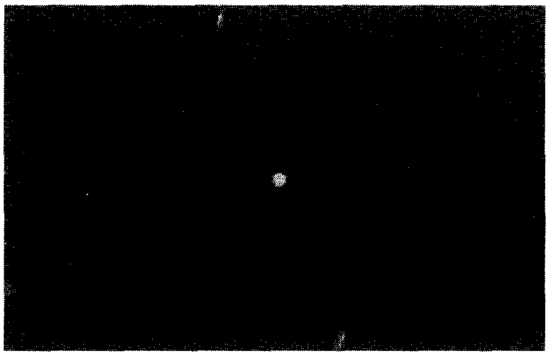
$$e^{\min}=\begin{pmatrix}-0.2399119037244083\\-0.2399119037244083\\-0.2399119037244083\\0.9095750850556481\end{pmatrix},\qquad e^{\max}=\begin{pmatrix}0.5251434202050548\\0.5251434202050548\\0.5251434202050548\\0.4155396065912507\end{pmatrix}.$$



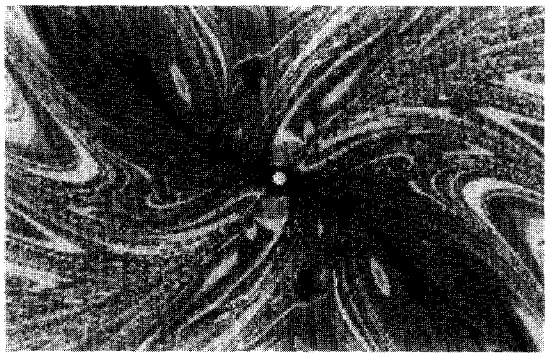
Contour Lines



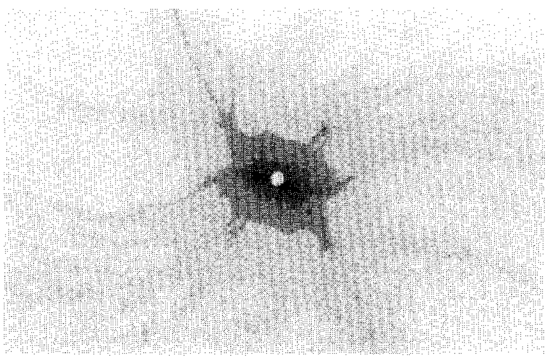
Armijo



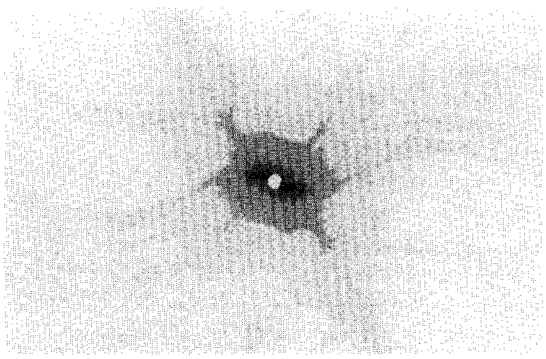
Fletcher - Reeves



Polak - Ribiere

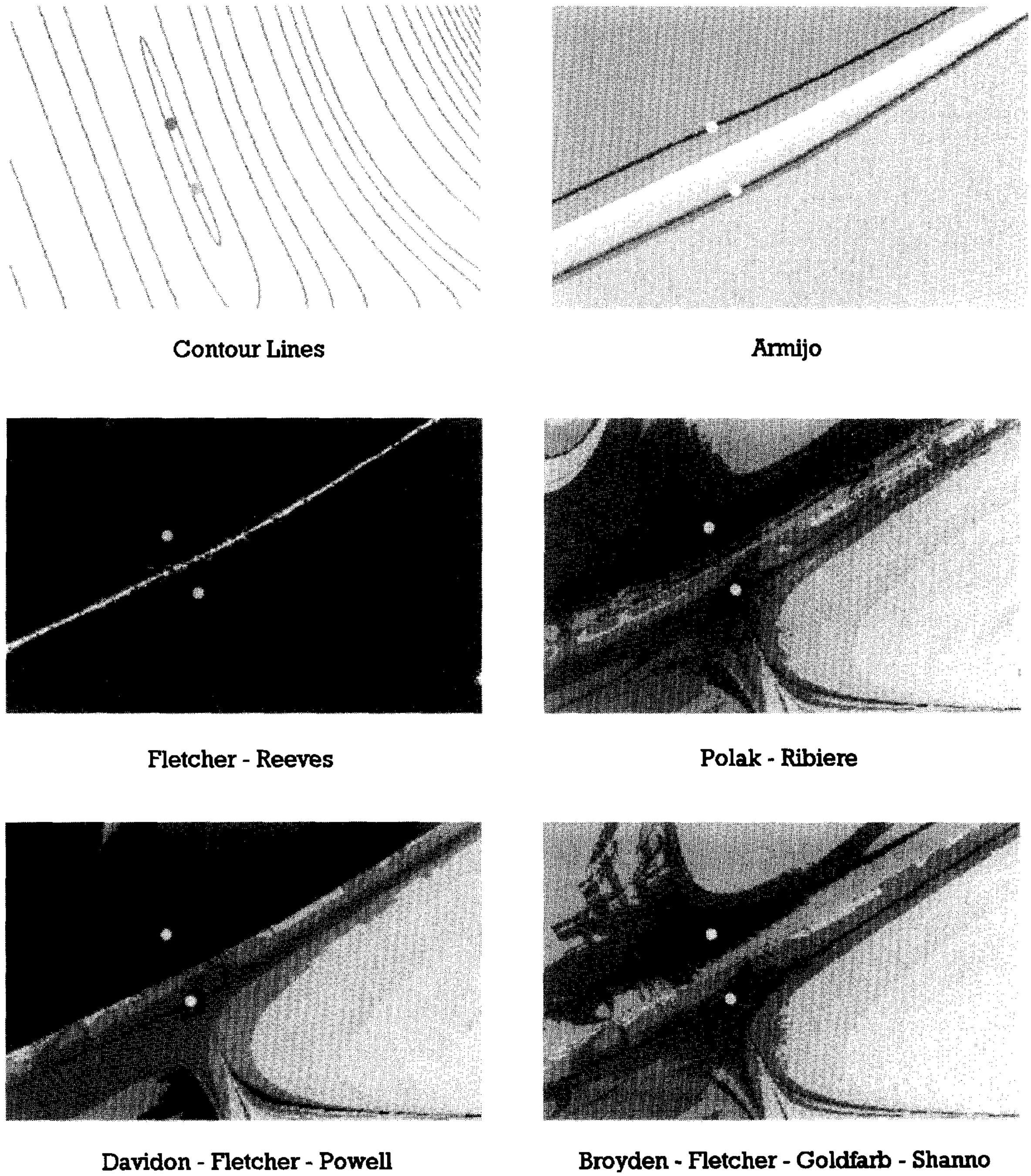


Davidon - Fletcher - Powell

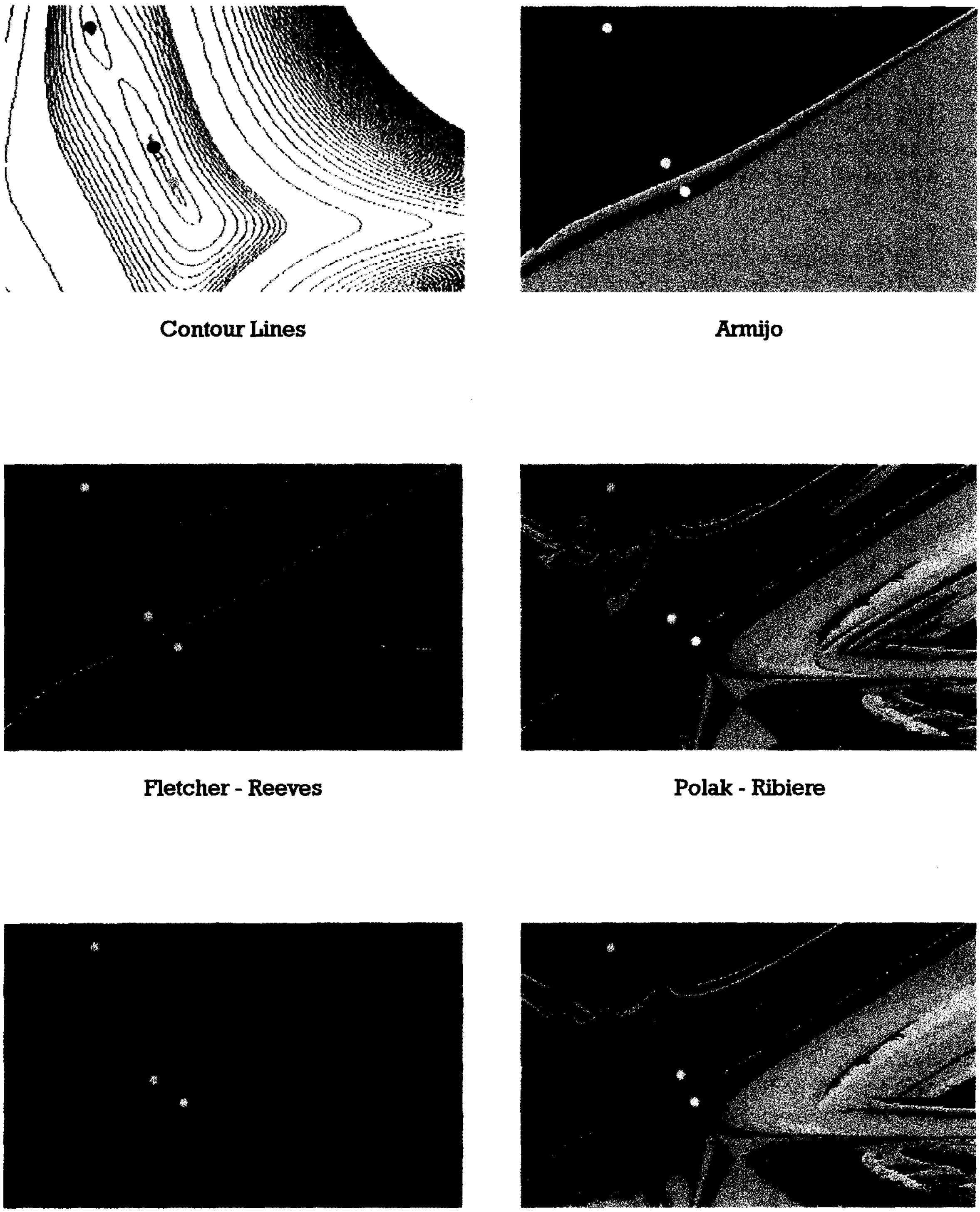


Broyden - Fletcher - Goldfarb - Shanno

Application 5.4. Box $[-0.6259, 0.6259] \times [-2.7583, 2.7583]$.



Appliction 5.5A. Box [0, 4.9774] × [0, 0.4746].



Application 5.5B. Box [−3.4313, 10.2938] × [−1.4278, −0.4759].

Application 5.6. *Extended Kearfott function,* $n = 4$ [4, 10]:

$$f(x_1,x_2,x_3,x_4) = (x_1^2 - x_2)^2 + (x_2^2 - x_3)^2 + (x_3^2 - x_4)^2 + (x_4^2 - x_1)^2. \tag{5.7}$$

This function has two minima $x_1^* = (1,1,1,1)$ and $x_2^* = (0,0,0,0)$ with $f(x_i^*) = 0$, $i = 1,2$. The eigenvalues of the $\nabla^2 f$ at the considered minimum $x^* = (1,1,1,1)$ are $\lambda = (2,10,10,18)$. The normalized eigenvector for the minimum eigenvalue is $e^{\min} = (\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2})$ and for the maximum is $e^{\max} = (-\frac{1}{2},\frac{1}{2},-\frac{1}{2},\frac{1}{2})$.

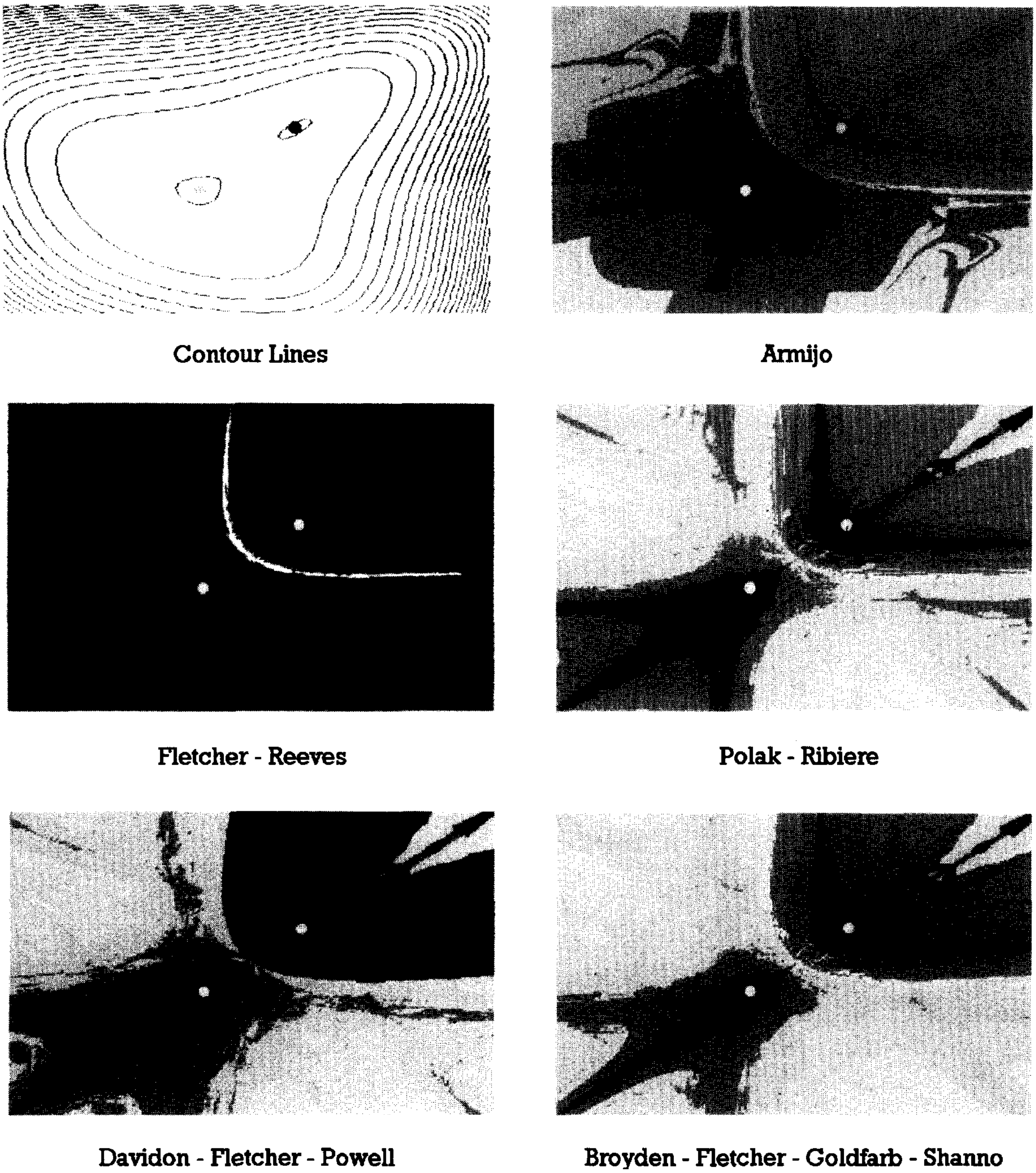
In the sequel we present the pictures gathered together for all the algorithms considered here as well as the corresponding quantitative measurements for each one of the above applications.

The first item in the pictures indicates the contours of constant values of f around the minima which are distinguished by different colors. The yellow spots indicate the locations of the minima. The reported parameters in the tables indicate:
- n dimension,
- $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ approximate minimum computed according to the above stopping criteria,
- the colors G (Green), R (Red), C (Cyan) and B (Black, Gray) which distinguish the minima,
- TOT the percentage of the starting points which converge to a specific minimum,
- RA the length of the radius of convergence,
- RE methods reliability (percentage),
- IT the total number of iterations required to obtain x^*,
- FE the total number of function evaluations (including derivatives).

The numbering from one to eight in the color zone expresses the percentages of the color shades starting from the darkest. Also, the averages reported in Tables 1–5 do not include failures by an algorithm.

Looking at the pictures of Application 5.1 we easily observe that the Fletcher–Reeves method does not converge for all the starting points (there is a white colored region at the center of the picture), while all the others do. The basins of attraction for all the methods (except Armijo's) have a fractal structure and they are sensitive to small changes of the starting points. Also, we observe that the picture of FR has dark color shades mostly. This does not mean that it possesses more rapid convergence relative to other methods, but that the starting points for which this scheme needs less function evaluations to converge than other starting points are densely distributed. For instance, looking at the minima colored by G (Green), R (Red), C (Cyan), and the corresponding shades (numbered from one to eight starting from the darkest), we observe that the basin of fastest convergence is formed by the Fletcher–Reeves algorithm since the first color shade occupies the highest percentage 94.5% of the total "green" area. The corresponding tables show that the most rapid on average methods are DFP and BFGS. Also, we easily observe that the most reliable method, the method that converges to the closest minimum, is Armijo's method while the worst is the Fletcher–Reeves as it is verified, by the corresponding tables, that its reliability is 44.05% while Armijo has 88.24%. Moreover, Armijo's method has the longest radii of convergence. Furthermore, we observe that the algorithms have different attractors since the green minimum attracts the highest percentage regarding Armijo's method while for all the other algorithms the green has the lowest percentage.

Applications 5.2 and 5.3 have different number of minima but they exhibit relative behavior like Application 5.1. In the corresponding pictures the shapes of the basins of attraction change but they



Application 5.6. Box $[0, 6] \times [-4, 0]$.

Table 1
Armijo method

		Color Zone												
		TOT	1	2	3	4	5	6	7	8	RA	RE	IT	FE
5.1	G	35.50	0.0	0.1	3.5	9.7	27.3	34.5	21.1	3.9	0.84	88.24	16	285
	R	32.25	0.1	2.0	5.6	16.2	30.5	27.3	16.7	1.6	0.85			
	C	32.25	0.1	2.0	5.6	16.2	30.5	27.3	16.7	1.6	0.85			
5.2	G	32.66	65.5	28.2	4.8	1.1	0.3	0.1	0.0	0.0	0.92	94.69	49	807
	R	67.34	0.4	73.3	17.5	6.8	1.6	0.4	0.1	0.0	0.93			
5.3	G	26.01	0.5	19.8	61.3	14.6	3.0	0.6	0.1	0.1	1.97	94.64	19	331
	R	26.20	0.1	1.1	9.9	51.6	33.3	3.9	0.1	0.1	2.79			
	C	24.25	0.2	2.7	29.0	53.7	10.9	3.1	0.4	0.1	2.41			
	B	23.54	0.1	0.9	6.2	25.6	43.1	19.8	4.0	0.4	2.21			
5.4	G	100.00	0.0	0.1	0.2	0.5	2.1	7.5	26.7	62.8	2.00	100.00	1426	27498
5.5	G	54.90	0.1	0.8	6.0	90.9	1.5	0.5	0.2	0.1	0.38	90.41	229	3324
$n=3$	R	31.59	0.4	1.5	15.8	77.6	2.8	1.2	0.5	0.1	0.41			
	C	13.46	0.1	0.1	0.8	5.1	23.7	45.0	24.0	1.3	1.42			
5.5	G	59.84	0.5	1.2	4.6	83.5	3.8	3.1	2.1	1.3	0.24	94.49	1094	16186
$n=4$	R	35.65	0.7	0.9	1.9	4.5	12.3	70.0	5.1	4.6	0.21			
5.6	G	67.68	1.9	16.0	49.8	29.8	2.4	0.1	0.0	0.0	0.71	82.32	16	215
	R	32.32	0.0	0.5	4.9	39.0	51.5	3.6	0.4	0.0	0.71			
Average			4.2	8.9	13.4	31.0	16.5	14.6	7.0	4.6	1.17	92.11	407.0	6949.4

Table 2
Fletcher-Reeves method

		Color zone TOT	1	2	3	4	5	6	7	8	RA	RE	IT	FE
5.1	G	29.14	94.5	3.1	1.2	0.6	0.3	0.3	0.1	0.0	0.17	44.05	59	976
	R	35.07	95.8	2.2	1.0	0.5	0.3	0.2	0.0	0.0	0.17			
	C	35.07	95.8	2.2	1.0	0.5	0.3	0.2	0.0	0.0	0.17			
5.2	G	50.33	94.9	2.7	1.3	0.6	0.3	0.2	0.0	0.0	0.55	74.78	64	860
	R	48.34	95.8	2.2	0.9	0.7	0.3	0.2	0.0	0.0	0.41			
5.3	G	31.72	97.8	1.3	0.4	0.2	0.1	0.1	0.0	0.0	0.21	70.37	45	635
	R	23.25	97.3	1.4	0.6	0.4	0.2	0.1	0.0	0.0	0.16			
	C	23.09	97.2	1.4	0.6	0.4	0.2	0.1	0.0	0.0	1.08			
	B	21.22	98.2	1.2	0.3	0.2	0.1	0.1	0.1	0.0	1.17			
5.4	G	100.00	99.4	0.4	0.1	0.1	0.0	0.0	0.0	0.0	2.00	100.00	40	518
5.5	G	44.51	98.4	0.9	0.3	0.2	0.1	0.1	0.0	0.0	0.34	69.90	42	535
$n = 3$	R	37.19	98.0	1.0	0.5	0.2	0.2	0.1	0.0	0.0	0.36			
	C	17.58	97.9	1.2	0.5	0.2	0.2	0.0	0.0	0.0	0.52			
5.5	G	58.10	98.4	0.6	0.3	0.2	0.1	0.1	0.1	0.1	0.19	92.76	48	574
$n = 4$	R	40.72	97.9	0.9	0.4	0.3	0.2	0.2	0.1	0.0	0.22			
5.6	G	73.04	98.8	0.5	0.2	0.2	0.1	0.1	0.0	0.0	0.30	75.50	32	393
	R	26.21	96.7	1.6	0.6	0.2	0.4	0.2	0.1	0.1	0.30			
Average			97.2	1.5	0.6	0.3	0.2	0.1	0.0	0.0	0.49	75.34	47.1	641.6

Table 3
Polak-Ribiere method

		Color Zone												
		TOT	1	2	3	4	5	6	7	8	RA	RE	IT	FE
5.1	G	29.96	7.5	34.8	25.3	20.0	9.2	2.4	0.8	0.1	0.17	46.21	7	110
	R	35.02	11.6	50.5	27.3	9.1	1.4	0.1	0.0	0.0	0.17			
	C	35.02	11.6	50.5	27.3	9.1	1.4	0.1	0.0	0.0	0.17			
5.2	G	52.24	0.8	8.8	20.0	40.4	15.0	14.2	0.9	0.0	0.55	74.94	8	110
	R	48.34	0.7	10.2	44.6	31.3	11.7	1.3	0.1	0.0	0.41			
5.3	G	32.51	1.9	36.0	43.9	16.8	1.3	0.1	0.0	0.0	0.21	72.33	7	104
	R	24.02	1.8	18.6	43.8	22.2	11.0	2.3	0.2	0.0	0.16			
	C	21.38	4.0	33.1	35.2	20.0	6.4	1.4	0.3	0.1	1.08			
	B	22.10	0.5	11.1	43.5	34.6	9.2	1.1	0.1	0.0	1.17			
5.4	G	100.00	10.0	50.5	17.5	9.5	5.9	3.8	2.5	0.3	2.00	100.00	41	527
5.5	G	47.54	1.5	16.3	28.1	33.6	15.1	4.7	0.5	0.2	0.38	71.37	9	117
$n = 3$	R	38.85	1.2	22.9	48.4	19.1	4.7	1.6	2.0	0.0	0.41			
	C	16.58	11.4	37.3	26.0	19.6	4.7	0.3	0.6	0.1	0.52			
5.5	G	60.76	1.3	12.5	25.7	17.8	15.3	16.5	9.9	1.0	0.27	93.74	8	113
$n = 4$	R	39.17	0.2	20.4	51.3	22.6	4.9	0.6	0.1	0.0	0.30			
5.6	G	73.17	0.5	12.0	21.7	41.1	19.2	5.3	0.2	0.1	0.30	76.13	10	125
	R	26.82	1.1	8.9	13.5	26.8	42.2	7.2	0.2	0.0	0.30			
Average			4.0	25.6	31.9	23.2	10.5	3.7	1.1	0.1	0.50	76.39	12.9	172.3

Table 4
Davidon-Fletcher-Powell method

		Color Zone TOT	1	2	3	4	5	6	7	8	RA	RE	IT	FE
5.1	G	29.98	99.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.17	46.90	5	92
	R	35.01	6.8	44.0	35.5	11.3	2.1	0.3	0.0	0.0	0.17			
	C	35.01	10.5	46.5	33.2	8.5	1.2	0.1	0.0	0.0	0.17			
5.2	G	50.29	8.2	50.7	31.5	9.6	0.1	0.0	0.0	0.0	0.55	76.83	7	90
	R	49.71	12.11	77.0	10.6	0.2	0.0	0.0	0.0	0.0	0.37			
5.3	G	32.86	0.6	9.7	39.2	31.5	14.4	4.4	0.2	0.0	0.34	72.48	6	83
	R	24.21	3.6	37.5	46.3	11.6	1.0	0.1	0.0	0.0	0.16			
	C	20.93	2.7	23.6	39.1	27.3	6.4	0.9	0.1	0.0	1.08			
	B	22.01	0.5	11.0	50.3	30.3	7.5	0.3	0.0	0.0	1.17			
5.4	G	100.00	0.2	1.1	4.4	9.4	23.3	30.8	23.9	6.9	2.00	100.00	13	157
5.5	G	46.54	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.38	76.09	8	103
$n = 3$	R	36.30	96.3	3.7	0.0	0.0	0.0	0.0	0.0	0.0	0.42			
	C	17.16	2.2	24.8	38.9	15.6	14.2	6.3	0.1	0.1	0.86			
5.5	G	59.81	1.6	23.5	21.7	15.9	18.0	15.5	3.8	0.0	0.28	95.59	8	97
$n = 4$	R	40.18	82.3	17.6	0.1	0.0	0.0	0.0	0.0	0.0	0.30			
5.6	G	72.19	1.5	8.6	27.2	37.1	25.0	0.7	0.0	0.0	0.30	77.13	10	127
	R	27.81	14.5	81.2	4.2	0.1	0.0	0.0	0.0	0.0	0.30			
Average			26.1	27.1	22.5	12.3	6.7	3.5	1.7	0.4	0.53	77.86	8.1	107.0

Table 5
Broyden-Fletcher-Goldfarb-Shanno method

		Color zone												
		TOT	1	2	3	4	5	6	7	8	RA	RE	IT	FE
5.1	G	29.95	2.7	32.0	30.6	23.1	9.2	1.7	0.5	0.1	0.17	46.29	6	92
	R	35.02	9.2	42.8	35.1	11.3	1.3	0.2	0.0	0.0	0.17			
	C	35.02	9.2	42.8	35.1	11.3	1.3	0.2	0.0	0.0	0.17			
5.2	G	50.53	0.5	8.0	16.0	39.6	15.7	17.0	3.1	0.1	0.55	76.44	7	90
	R	49.47	0.5	6.2	33.5	34.5	23.4	1.7	0.1	0.0	0.41			
5.3	G	32.80	0.5	6.9	26.8	38.8	20.1	6.1	0.7	0.1	0.21	72.46	6	83
	R	24.22	1.3	14.5	48.3	23.5	10.9	1.3	0.2	0.0	0.16			
	C	20.90	2.6	19.6	41.2	26.9	8.2	1.2	0.3	0.0	1.08			
	B	22.08	0.4	6.6	44.0	38.9	9.1	1.0	0.1	0.0	1.17			
5.4	G	100.00	0.2	0.7	4.4	7.8	16.7	32.6	26.0	11.6	2.00	100.00	12	157
5.5	G	46.83	2.4	12.8	24.0	30.9	23.6	5.8	0.2	0.3	0.38	76.26	8	102
$n = 3$	R	35.42	1.0	16.8	45.2	27.5	4.9	1.7	1.6	1.3	0.41			
	C	17.75	6.3	33.7	30.4	16.4	10.3	1.6	0.7	0.5	0.52			
5.5	G	59.88	1.3	14.6	26.2	14.1	13.8	13.9	13.4	2.7	0.28	95.26	8	96
$n = 4$	R	40.11	0.3	13.3	32.8	38.1	12.3	2.3	0.6	0.2	0.30			
5.7	G	72.71	0.8	4.7	15.9	28.4	33.9	15.5	0.6	0.0	0.30	76.59	10	126
	R	27.29	0.2	3.7	20.7	47.9	23.2	3.6	0.6	0.1	0.30			
Average			2.3	16.5	30.0	27.0	14.0	6.3	2.9	1.0	0.50	77.61	8.1	106.6

retain the fractal structure, except for Armijo's method. Only the percentages of the quantitative information are different.

By increasing the dimension and taking starting points determined by relation (3.1) with $-0.6259 \leqslant c_1 \leqslant 0.6259$, $-2.7583 \leqslant c_2 \leqslant 2.7583$ applied to the function of Application 5.4 we obtain various nested basins of attractions which correspond to different number of function evaluations (different color shades). The most sensitive behavior is exhibited by the PR algorithm. For this application the most expensive method regarding the function evaluations is Armijo's method. The DFP and BFGS have similar quantitative measurements. In this case the contours of constant values of f around the minima are drawn which lie on the considered subspace.

The function of Application 5.5 for $n=3$ has three minima lying in the two-dimensional subspace spanned by the extreme eigenvectors of $\nabla^2 f$ at $x^* = (1,1,1)$. The behavior of the methods is similar to the two-dimensional cases. Also, the PR, DFP and BFGS methods have a similar fractal structure but the DFP and BFGS are slightly better on the quantitative averages than PR. The most expensive method is Armijo's algorithm but on the other hand it exhibits the most stable behavior.

By increasing the dimension of the function of Application 5.5 by taking $n = 4$, we have two real minima lying in the two-dimensional subspace spanned by the extreme eigenvectors of $\nabla^2 f$ at $x^* = (1,1,1,1)$. The behavior of the methods are similar as in the three-dimensional case since the PR, DFP and BFGS methods have a similar fractal structure while the DFP and BFGS are slightly better on the quantitative averages than PR. Also, Armijo's method and FR do not converge for all the considered starting points. It is worth noticing here that the quantitative measurements on average were dramatically increased for Armijo's method and slightly for FR while, on the contrary, they were decreased for all the other algorithms. On the other hand, Armijo's algorithm exhibits the most stable behavior.

Similar results are obtained for Application 5.6 for $n=4$. In this case the PR was the best method. Also the FR algorithm does not converge for all the considered starting points while all the other methods do.

All the considered applications indicate that, regarding the quantitative measures, Armijo's reliability (RE) possesses, on average, the highest percentage. Also this method has the longest radii (RA) of convergence. On the other hand, Armijo's method is the most expensive one since it presents the highest average of number of iterations (IT) and function evaluations (FE). The FR algorithm does not always converge while the rest of the methods are not stable but they are rapidly convergent methods.

6. Concluding remarks

We have presented a software package for analyzing and visualizing the convergence behavior of optimization algorithms. Given a finite domain of starting points the package determines basins of attraction and relative convergence rates. Using this package, we are able to trace the proper algorithm for a given class of problems which possess similar objective functions (neural nets, economical problems, etc.) by checking only one of these problems.

OPTAC gives quantitative measures for the terms "cost", "fast" and "sensitive". The information accumulated by OPTAC can be useful when studying a particular optimization problem. Also, it can likely help researchers doing applications that require global optimization.

We have seen the geometry of basins of attraction for well-known and widely used unconstrained optimization methods and we have given pictorial evidence that these basins of attraction have apparently a fractal (or "chaotic") structure.

The OPTAC package can be applied to any dimension by taking a finite domain of starting points of the two-dimensional subspace of $\mathbb{R}^n$ spanned by the eigenvectors corresponding to the extreme eigenvalues of the Hessian of f at a minimum x^*. This two-dimensional subspace reveals useful information when studying the behavior of optimization methods for various directions including the "extreme ones".

Acknowledgements

We thank the referee for his helpful and constructive comments. Partial support by a grant from the PENED program of the Ministry of Industry Energy and Technology of Greece, Gen. Secr. Res. Tech. is acknowledged.

References

[1] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific J. Math.* **16** (1966) 1–3.

[2] C.A. Botsaris, A curvilinear optimization method based upon iterative estimation of the eigensystem of the Hessian matrix, *J. Math. Anal. Appl.* **63** (1978) 396–411.

[3] J.E. Dennis, Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[4] B. Kearfott, An efficient degree–computation method for a generalized method of bisection, *Numer. Math.* **32** (1979) 109–127.

[5] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, (Addison-Wesley, Reading, MA, 1973).

[6] B.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization, *ACM Trans. Math. Software* **7** (1981) 17–41.

[7] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[8] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press, New York, 1992).

[9] F. Stenger, Computing the topological degree of a mapping in $\mathbb{R}^n$, *Numer. Math.* **25** (1975) 23–38.

[10] M.N. Vrahatis, Solving systems of nonlinear equations using the nonzero value of the topological degree, *ACM Trans. Math. Software* **14** (1988) 312–329.

[11] M.N. Vrahatis, G.S. Androulakis and G.E. Manoussakis, A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.*, 197 (1996) 586–607.