Three Variants of the Generalized Ellipsoid Method

Petro Stetsyuk[1], Andreas Fischer[2], Olha Khomiak[1]

[1]V.M. Glushkov Institute of Cybernetics,
Kyiv, Ukraine
[2]Institute of Numerical Mathematics,
Technische Universität Dresden, Dresden, Germany

XXVII INTERNATIONAL CONFERENCE "AUTOMATION 2024"
Conference dedicated to memory
Academicians of NAS of Ukraine V.M. Kuntsevych and Yu.G. Kryvonos
Dnipro, November 20–22, 2024

Content

1. Some facts of the ellipsoid method history
2. The generalized ellipsoid method (GEM)
3. Three variants of the GEM
4. The algorithm **emshor** and its **Octave** implementation

Content

The ellipsoid method was proposed

1976 by **Yudin and Nemirovskii** as a method of successive cutting-plane [*],

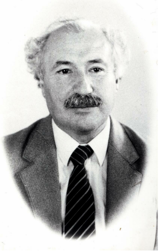
1977 by **Shor** as a variant of the method with space dilation in the direction of the subgradient [**].

* Yudin D.B. and Nemirovskii A.S. *Informational complexity and effective methods for the solution of convex extremal problems* // Ekonom. Mat. Metody, 12, No. 2 (1976).

** Shor N.Z. *Cut-off method with space extension in convex programming problems* // Cybernetics, 13, No. 1 (1977).

Yudin and Shor „from the banks of the Dnipro“

David Borisovich Yudin
born May 21, 1919
in Yekaterinoslav (today - Dnipro),
in 1941 graduated from
Dnepropetrovsk University

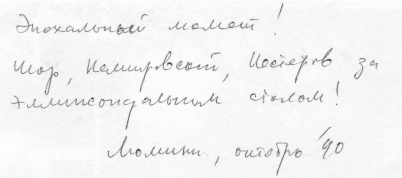


Naum Zuselevich Shor
born January 1, 1937
in Kyiv (city on the Dnipro),
in 1958 graduated from
Kyiv University

Oles Honchar Dnipro National University

Epochal moment!

N. Shor,
A. Nemirovski,
Y. Nesterov at the
ellipsoidal table!
October 1990



XI ISMP, Bonn, August 23–27, 1982

Fulkerson Prizes for the ellipsoid method:

1. Grötchel M., Lóvasz L., Schrijver A. (**1981**)
2. Khachiyan L. (**1979**), Yudin D., Nemirovski A. (**1976**)

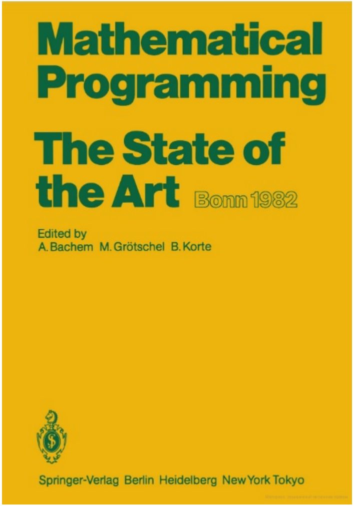
Shor's plenary report:

„Generalized gradient methods of nondifferentiable optimization employing space dilatation operations", published in [*]

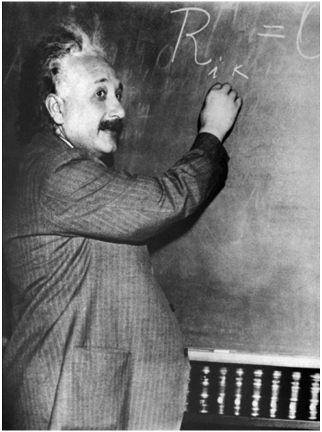
* MATHEMATICAL PROGRAMMING: THE STATE OF ART, BONN, 1982 / *Bachem A., Grötchel M., Korte B. (eds.)* – Berlin: Springer-Verlag, 1983. – 655 p.

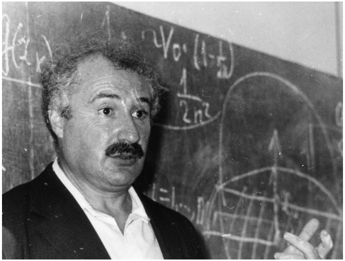
XI International Math. Programming Symposium

N. Shor in Bonn (1982)



Einstein writes a message to Shor





to denote space dilation operator as $R_\alpha(\xi)$

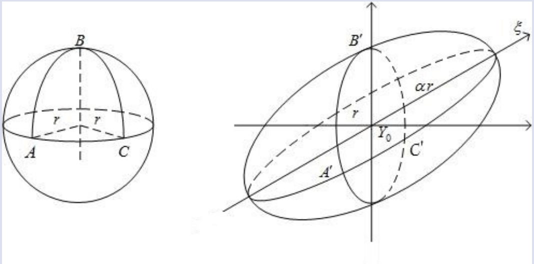
Shor's space dilation operator

Space dilation operator has the following form

$$R_\alpha(\xi) = I_n + (\alpha - 1)\xi\xi^T, \quad \text{where} \quad \alpha > 1.$$

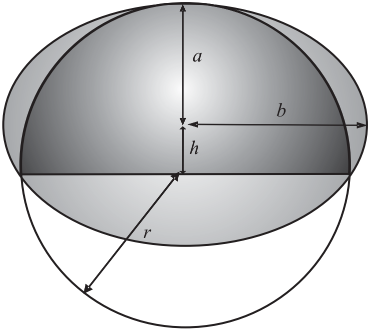
Here α is the coefficient of space dilation in the normed direction $\xi \in \mathbb{R}^n$, $\|\xi\| = 1$; I_n is the identity $n \times n$-matrix.

Example in $\mathbb{R}^3$: ball (left) is dilated to ellipsoid (right)



Content

The GEM: 1d-ellipsoid and its properties



The 1d-ellipsoid $\mathcal{E}_n$, containing half of ball S_n in $\mathbb{R}^n$, has parameters

$$b = \left(\alpha + \frac{1}{\alpha}\right)\frac{r}{2}, \quad h = \left(1 - \frac{1}{\alpha^2}\right)\frac{r}{2},$$

where $\alpha = \frac{b}{a}$ and r – radius of ball.

To transform $\mathcal{E}_n$ into a „new" ball we have to dilate the space with coefficient $\alpha = \frac{b}{a} > 1$.

The ratio of $\mathcal{E}_n$ volume to S_n volume equals

$$q_n(\alpha) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{b}{r}\right)^n = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n.$$

Why the GEM converges?

The ratio of $\mathcal{E}_n$ to S_n volumes equals

$$q_n(\alpha) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n .$$

If α is such that

$$\alpha + 1/\alpha < 2\sqrt[n]{\alpha},$$

then

$$q_n(\alpha) < 1.$$

The GEM converges on the volume

of the ellipsoid localizing searched point x^* with the rate of a geometric progression with the denominator $q_n(\alpha)$.

Basic problem for the GEM

Basic problem

Given a mapping $g := \mathbb{R}^n \Rightarrow \mathbb{R}^n$ and there is a point $x^* \in \mathbb{R}^n$ such that $g(x)^\top (x - x^*) \geq 0$ for all $x \neq x^*$.
For convenience, we will assume that x^* is unique.
It is required to find an approximation to the point x^*.

Applications of the GEM

1. Minimization of convex functions
2. Minimization of convex function on a ball
3. Constrained convex programming
4. Saddle point problems for convex-concave functions
5. Lagrangian bounds for nonconvex optimization problems

Algorithm of the GEM in B-form

Step 0. Choose $\alpha > 1$ such that $\alpha + 1/\alpha < 2\sqrt[n]{\alpha}$, $x_0 \in \mathbb{R}^n$, a non-singular matrix $B_0 \in \mathbb{R}^{n \times n}$ and $r_0 > 0$ so that $\left\| B_0^{-1}(x_0 - x^*) \right\| \le r_0$. Moreover, set $k := 0$

Step 1. Set $g_k: = g(x_k)$. If $g(x_k)=0$, then STOP: $k^*:=k$, $x^*:=x_k$.

Step 2. Compute

$$x_{k+1}:=x_k - h_k B_k \xi_k, \text{ where } \xi_k := \frac{B_k^\top g(x_k)}{\left\| B_k^\top g(x_k) \right\|}, \; h_k := \left(1 - \frac{1}{\alpha^2}\right) \frac{r_k}{2}.$$

Step 3. Update

$$B_{k+1} := B_k + \left(\frac{1}{\alpha} - 1\right)(B_k \xi_k)\, \xi_k^\top, \quad r_{k+1} := \frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right) r_k.$$

Step 4. Set $k := k + 1$ and go to Step 1.

Theorem

Let $x_k \in \mathbb{R}^n$ *and* $x_{k+1} \in \mathbb{R}^n$ *be generated by the algorithm of the GEM. Then,*

$$\|A_k(x_k - x^*)\| = \|B_k^{-1}(x_k - x^*)\| \le r_k, \quad k \in \mathbb{N}. \qquad (1)$$

The ratio of volumes of the ellipsoids $\mathcal{E}_{k+1}$ *and* $\mathcal{E}_k$ *does not depend on* k *and is equal to*

$$q_n(\alpha) := \frac{vol(\mathcal{E}_{k+1})}{vol(\mathcal{E}_k)} = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n < 1. \qquad (2)$$

Moreover, $x^* \in \mathcal{E}_k$ *for all* $k \in \mathbb{N}$.

$\mathcal{E}_k = \{x \mid \|B_k^{-1}(x_k - x)\| \le r_k\}$ *is ellipsoid with center* $x_k \in \mathbb{R}^n$.

Content

Three partial cases of the GEM

1. Minimum Volume Ellipsoid (Yudin-Nemirovski-Shor, 1976-1977)

$$\alpha = \alpha^* = \sqrt{\frac{n+1}{n-1}} \quad \text{and coefficient} \quad q_n(\alpha^*) \le \exp\left\{-\frac{1}{2n}\right\}.$$

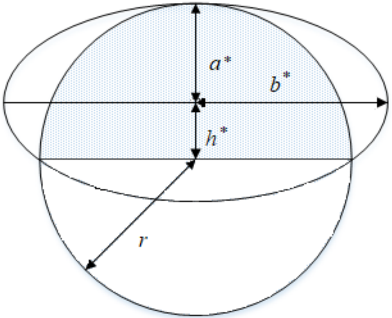
2. Approximate Ellipsoid (Stetsyuk, 2003)

$$\alpha = \alpha^{**} = \sqrt{1 + \frac{1}{n^2} + \frac{1}{n}} \quad \text{and} \quad q_n(\alpha^{**}) \le \exp\left\{-\frac{1}{2n} + \frac{1}{2n^2}\right\}.$$

3. Approximate Ellipsoid without square roots (2024)

$$\alpha = \alpha^{***} = 1 + \frac{1}{n} \quad \text{and} \quad q_n(\alpha^{***}) \le \exp\left\{-\frac{1}{2(n+1)}\right\}.$$

The minimum volume ellipsoid



$$a^* = \frac{n}{n+1} r$$

$$b^* = \frac{n}{\sqrt{n^2-1}} r$$

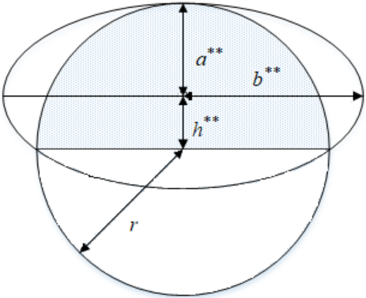
$$h^* = \frac{1}{n+1} r$$

$$\alpha^* = \frac{b^*}{a^*} = \sqrt{\frac{n+1}{n-1}} > 1$$

The ratio of $\mathcal{E}_n$ to S_n volumes equals

$$q_n(\alpha^*) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha^*}\left(\frac{b^*}{r}\right)^n = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n < \exp\left\{-\frac{1}{2n}\right\}$$

The approximate ellipsoid



$$a^{**} = r\left(1 - \frac{1}{n}\left(\sqrt{1+\frac{1}{n^2}} - \frac{1}{n}\right)\right)$$

$$b^{**} = r\sqrt{1+\frac{1}{n^2}}$$

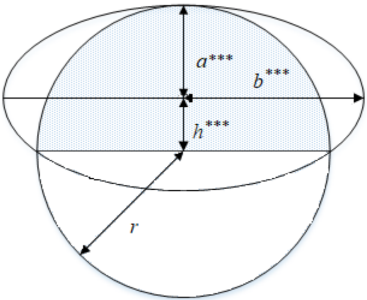
$$h^{**} = \frac{r}{n}\left(\sqrt{1+\frac{1}{n^2}} - \frac{1}{n}\right)$$

$$\alpha^{**} = \sqrt{1+\frac{1}{n^2}} + \frac{1}{n} > 1$$

The ratio of $\mathcal{E}_n$ to S_n volumes equals

$$q_n(\alpha^{**}) = \left(\sqrt{1+\frac{1}{n^2}}\right)\left(1+\frac{1}{n^2}\right)^{n/2} \leq \exp\left\{-\frac{1}{2n} + \frac{1}{2n^2}\right\}$$

Approximate ellipsoid without square roots



$$a^{***} = \frac{n}{n+1}\left(1 + \frac{1}{2n(n+1)}\right) r$$

$$b^{***} = \left(1 + \frac{1}{2n(n+1)}\right) r$$

$$h^{***} = \frac{1}{n+1}\left(1 - \frac{1}{2(n+1)}\right) r$$

$$\alpha^{***} = 1 + \frac{1}{n} > 1$$

The ratio of $\mathcal{E}_n$ to S_n volumes equals

$$q_n(\alpha^{***}) = \frac{n}{n+1}\left(1 + \frac{1}{2n(n+1)}\right)^n \leq \exp\left\{-\frac{1}{2(n+1)}\right\}$$

Comparison of methods (relative accuracy 10^{-10})

n	K_1	K_2	K_3	n	K_1	K_2	K_3
1	NaN	44	49	15	10354	10355	10366
2	177	179	183	20	**18414**	**18414**	18425
3	407	408	419	25	**28775**	**28775**	28787
4	**730**	**730**	741	30	**41439**	**41439**	41459
5	**1144**	**1144**	1156	40	**73676**	**73676**	73687
10	**4598**	**4598**	4609	50	**115122**	**115122**	115134

$$K_1 = \left\lceil \frac{-10n \ln 10}{\ln q_n(\alpha^*)} \right\rceil, K_2 = \left\lceil \frac{-10n \ln 10}{\ln q_n(\alpha^{**})} \right\rceil, K_3 = \left\lceil \frac{-10n \ln 10}{\ln q_n(\alpha^{***})} \right\rceil.$$

Content

Problem for algorithm emshor

Convex function minimization problem:

for a convex function $f(x)$, $x \in \mathbb{R}^n$, to find the point x^*_ε, subject to $f(x^*_\varepsilon) - f^* \le \varepsilon$, where $f^* = f(x^*)$, $\varepsilon > 0$.

Input parameters:

x_0 – starting point, $x_0 \in \mathbb{R}^n$;
r_0 – the radius of the ball localizing point $x^* \in \mathbb{R}^n$;
ε – parameter for stopping: x^*_ε: $f(x^*_\varepsilon) - f^* \le \varepsilon$.

Notation:

$g(x_k)$ – subgradient of $f(x)$ at point x_k.

The B-form of algorithm **emshor**(x_0, r_0, ε)

Step 0. Choose $x_0 \in \mathbb{R}^n$, $r_0 > 0$, $\varepsilon > 0$, $\|x_0 - x^*\| \le r_0$.

Set $B_0 := I_n \in \mathbb{R}^{n\times n}$ (denoting the identity matrix) and $k := 0$.

Step 1. If $\left\|B_k^\top g(x_k)\right\| r_k \le \varepsilon$, then STOP: $k^* := k$, $x_\varepsilon^* := x_k$.

Step 2. Compute

$$x_{k+1} := x_k - \frac{r_k}{n+1} B_k \xi_k, \quad \text{where} \quad \xi_k := \frac{B_k^\top g(x_k)}{\left\|B_k^\top g(x_k)\right\|}.$$

Step 3. Update

$$B_{k+1} := B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1\right)(B_k\xi_k)\,\xi_k^\top, \quad r_{k+1} := \frac{n}{\sqrt{n^2-1}} r_k.$$

Step 4. Set $k := k+1$ and go to Step 1.

Octave function **emshor**

```
# Input parameters:                                                #row00
# calcfg - name of the function for calculation of f and g         #row00a
# x0 - starting point, x0(1:n)                                     #row00b
# rad - the radius of the ball localizing the minimum point        #row00c
# epsf - parameter for stopping (accuracy)                         #row00d
# maxitn - parameter for stopping (maximal no. of iterations)      #row00e
# intp - printing interval (after each intp iterations)            #row00f
# Output parameters:                                               #row00g
# x - approximation of the minimum point, x(1:n)                   #row00h
# f - value of function f at the point x                           #row00i
# itn - number of iterations performed                             #row00j
# ist - exit code (1 = epsf, 4 = maxitn)                           #row00k

function[x,f,itn,ist] = emshor(calcfg,x0,rad,epsf,maxitn,intp); #row01
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0));          #row02
x=x0; radn=rad; B=eye(length(x));                                #row03
for (itn = 0:maxitn)                                             #row04
   [f,g1] = calcfg(x); g=B'*g1; dg=norm(g);                      #row05
   if(radn*dg < epsf) ist = 1;  return; endif                    #row06
   xi=(1.d0/dg)*g; dx = B * xi;                                  #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                              #row08
   B += (beta - 1) * (B * xi) * xi';                             #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);              #row10
   if(mod(itn,intp)==0)                                          #row11
     printf("itn %4d  f %14.6e\n",itn,f);                        #row12
   endif                                                         #row13
endfor                                                           #row14
ist = 4;                                                         #row15
endfunction #emshor                                              #row16
```

Octave function **emshor** (comments)

```
# Input parameters:
# calcfg - name of the function for calculation of f and g
# x0 - starting point, x0(1:n)
# rad - the radius of the ball localizing the minimum point
# epsf - parameter for stopping (accuracy)
# maxitn - parameter for stopping (maximum of iterations)
# intp - printing interval (after each intp iterations)
# Output parameters:
# x - approximation of the minimum point, x(1:n)
# f - value of function f at the point x
# itn - number of iterations performed
# ist - exit code (1 = epsf, 4 = maxitn)
```

Octave function **emshor** (program code)

```
function[x,f,itn,ist]
                 = emshor(calcfg,x0,rad,epsf,maxitn,intp);    #row01
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0));     #row02
x=x0; radn=rad; B=eye(length(x));                            #row03
for (itn = 0:maxitn)                                         #row04
   [f,g1] = calcfg(x); g=B'*g1; dg=norm(g);                  #row05
   if(radn*dg < epsf) ist = 1;  return; endif                #row06
   xi=(1.d0/dg)*g; dx = B * xi;                              #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                          #row08
   B += (beta - 1) * (B * xi) * xi';                         #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);          #row10
   if(mod(itn,intp)==0)                                      #row11
     printf("itn %4d  f %14.6e\n",itn,f);                    #row12
   endif                                                     #row13
endfor                                                       #row14
ist = 4;                                                     #row15
endfunction #emshor                                          #row16
```

Computational experiments for ravine function

The results for minimization function
$f(x) = \sum_{i=1}^{n} 2^{i-1} |x_i - 1|$ by program **emshor**

$r_0 = 5$									
	$\varepsilon = 10^{-3}$			$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-9}$		
n	itn	$f(x_{itn})$	$time$	itn	$f(x_{itn})$	$time$	itn	$f(x_{itn})$	$time$
5	519	6.1e-06	0.05	873	1.1e-07	0.12	1201	1.2e-10	0.15
10	2484	8.7e-05	0.29	3829	7.2e-08	0.48	5295	8.4e-11	0.62
15	6541	6.5e-05	0.64	9612	2.0e-09	0.91	12733	4.0e-11	1.19
20	**13097**	4.8e-05	1.28	**18722**	8.9e-09	1.83	**23426**	2.2e-11	2.20
$r_0 = 500$									
	$\varepsilon = 10^{-3}$			$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-9}$		
n	itn	$f(x_{itn})$	$time$	itn	$f(x_{itn})$	$time$	itn	$f(x_{itn})$	$time$
5	747	1.1e-05	0.11	1080	1.6e-07	0.09	1392	1.6e-10	0.12
10	3429	9.0e-05	0.30	4810	9.3e-08	0.43	6197	7.9e-11	0.56
15	8615	5.6e-05	0.79	11692	1.1e-08	1.04	14796	1.6e-11	1.32
20	**16772**	4.8e-06	1.48	**22404**	4.0e-08	1.95	**27072**	3.1e-11	2.33

Conclusion

The Emshor program finds precise approximations to the minimizer of the ravine convex function.

The number of iterations grows slightly faster than n^2 does.

References–1

- Н.З. Шор. *Метод отсечения с растяжением пространства для решения задач выпуклого программирования.* Кибернетика. 1977. 1. С. 94–95.
- П.И. Стецюк. *Приближенный метод эллипсоидов.* Кибернетика и системный анализ. 2003. № 3. С. 141–146.
- П.И. Стецюк, А.В. Фесюк, О.Н. Хомяк. *Обобщенный метод эллипсоидов.* Кибернетика и системный анализ. 2018. Т. 54, 4. С. 70–80.

References–2

- P.I. Stetsyuk, A. Fischer, O. Khomyak. *The generalized ellipsoid method and its implementation.* In: Jácimovíc, M., Khachay, M., Malkova, V., Posypkin, M. (eds.) Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science, vol. 1145, pp. 355–370, Springer, Cham (2020).
- A. Fischer, O. Khomyak, P. Stetsyuk. *The ellipsoid method and computational aspects.* Commun. Optim. Theory, No. 21, 2023, pp. 1–14.

Thanks

Volkswagen Foundation
(grant No 97 775)

Questions?

THANK YOU
FOR YOUR ATTENTION!

e-mail: stetsyukp@gmail.com