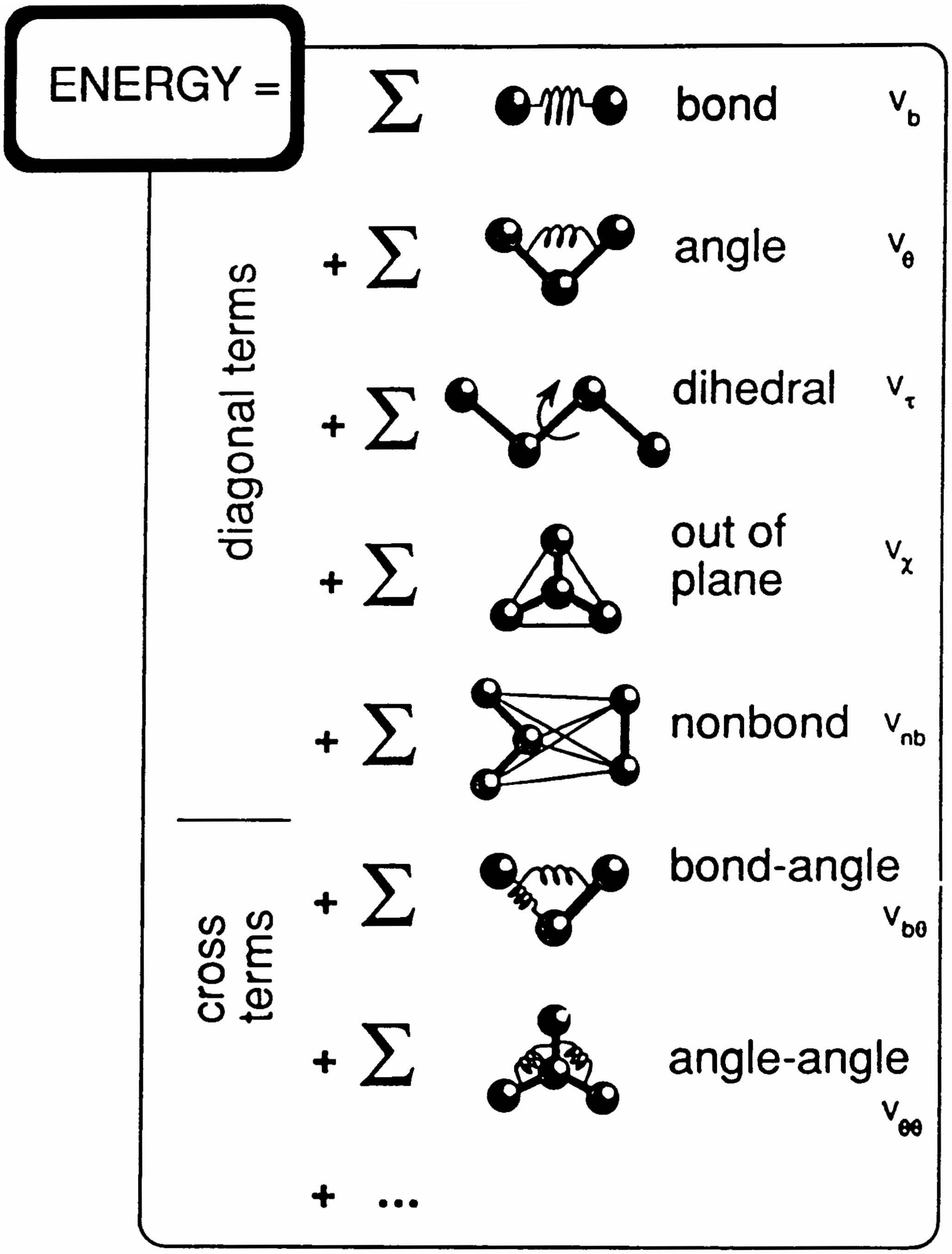


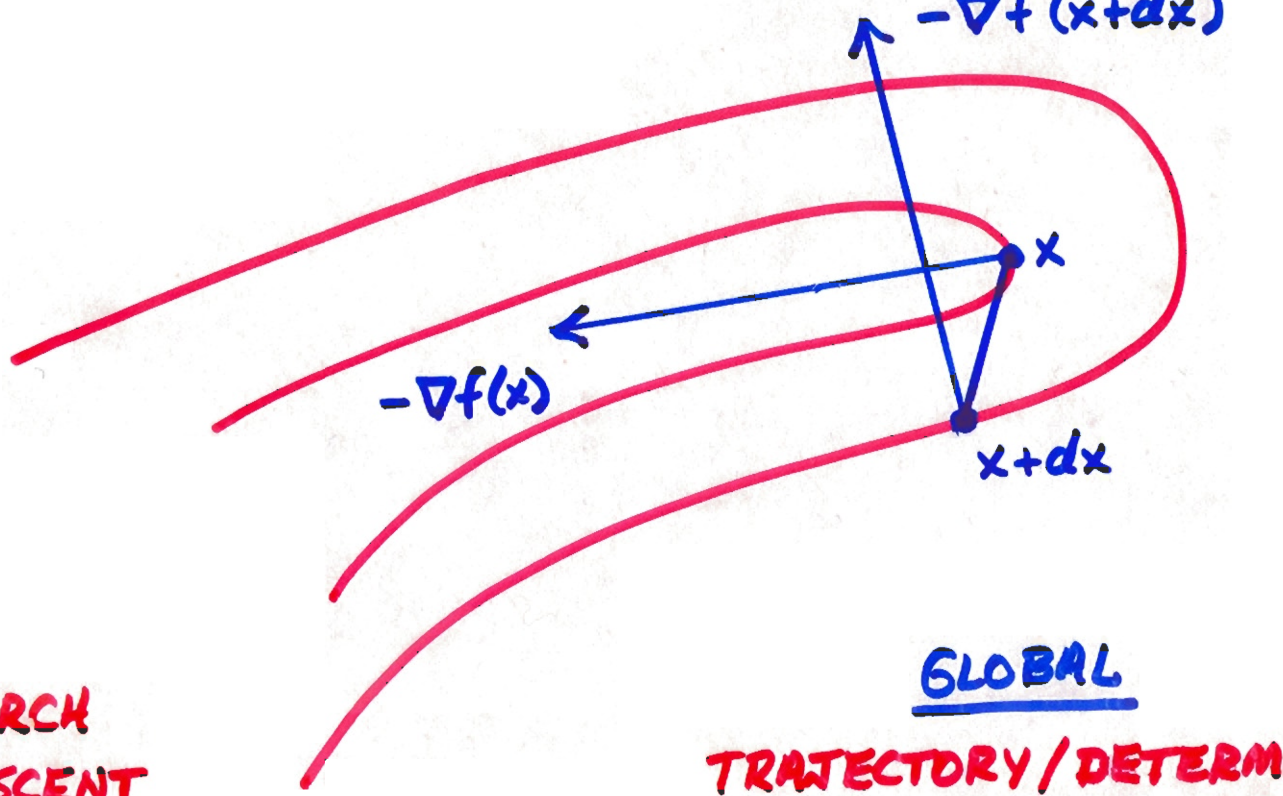
Figure 1 Schematic of molecular force field expression. Diagonal terms refer to interactions that can be expressed as a function of a single internal coordinate, whereas cross terms introduce coupled interactions involving two or more coordinates.

```
8   Ethane
1   C      0.000000    0.000000    0.000000     1     2     3     4     5
2   C      0.000000    0.000000    1.530000     1     1     6     7     8
3   H      1.037098    0.000000   -0.366645     5     1
4   H     -0.518502    0.898180   -0.366645     5     1
5   H     -0.518502   -0.898180   -0.366645     5     1
6   H      1.033151    0.090389    1.896645     5     2
7   H     -0.438247   -0.939953    1.896645     5     2
8   H     -0.594811    0.849572    1.896645     5     2


8   Ethane
1   C      1
2   C      1     1   1.53000
3   H      5     1   1.10000     2  109.4700
4   H      5     1   1.10000     2  109.4700     3 -120.0000      0
5   H      5     1   1.10000     2  109.4700     4 -120.0000      0
6   H      5     2   1.10000     1  109.4700     3    5.0000      0
7   H      5     2   1.10000     1  109.4700     3 -115.0000      0
8   H      5     2   1.10000     1  109.4700     3  125.0000      0


8   Ethane
1   C      1
2   C      1     1   1.53000
3   H      5     1   1.10000     2  109.4700
4   H      5     1   1.10000     2  109.4700     3  109.4700      1
5   H      5     1   1.10000     2  109.4700     3  109.4700     -1
6   H      5     2   1.10000     1  109.4700     3    5.0000      0
7   H      5     2   1.10000     1  109.4700     6  109.4700      1
8   H      5     2   1.10000     1  109.4700     6  109.4700     -1
```

OPTIMIZATION: LOCAL vs. GLOBAL

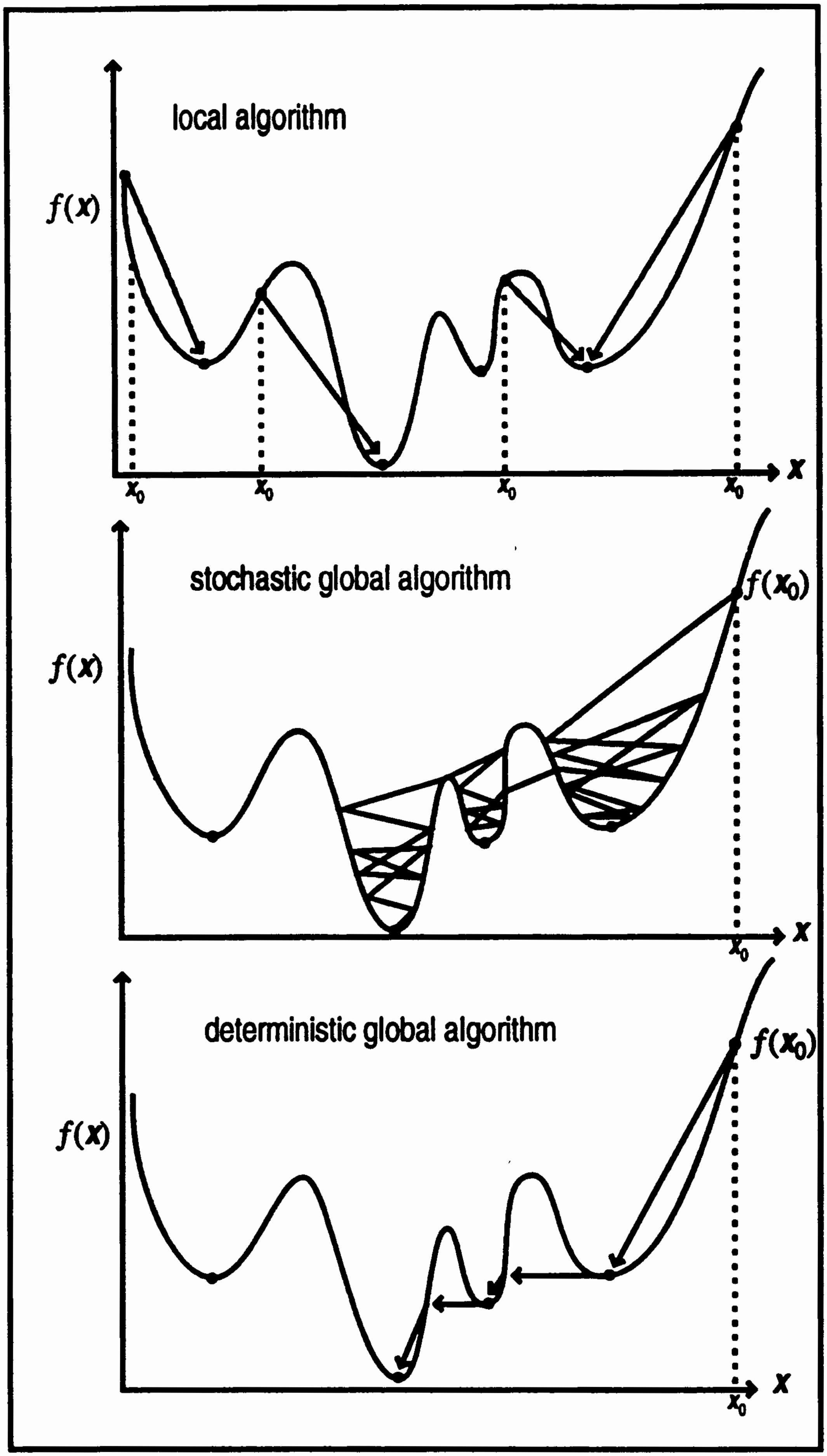


LOCAL

PATTERN SEARCH
STEEPEST DESCENT
NEWTON'S METHOD
VARIABLE METRIC

GLOBAL

TRAJECTORY/DETERMINISTIC
BREMMERMAN'S METHOD
CLUSTERING (TORN)
CONTROLLED RANDOM SEARCH
STOCHASTIC METHODS


local algorithm
f(x)
x
x0
x0
x0
x0
stochastic global algorithm
f(x)
f(x0)
x
x0
deterministic global algorithm
f(x)
f(x0)
x
x0

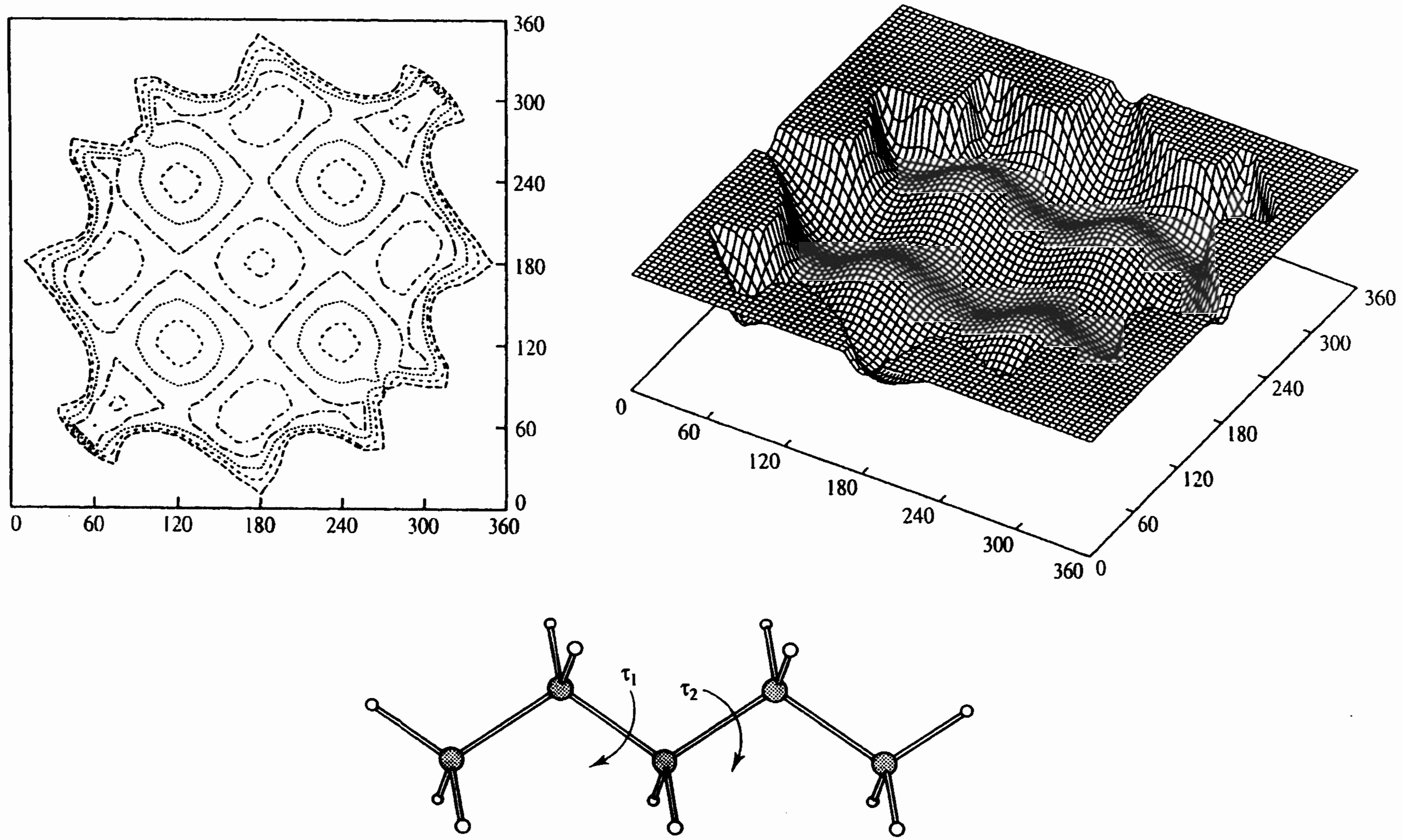


Fig. 5.1: Variation in the energy of pentane with the two torsion angles indicated and represented as a contour diagram and isometric plot. Only the lowest-energy regions are shown.

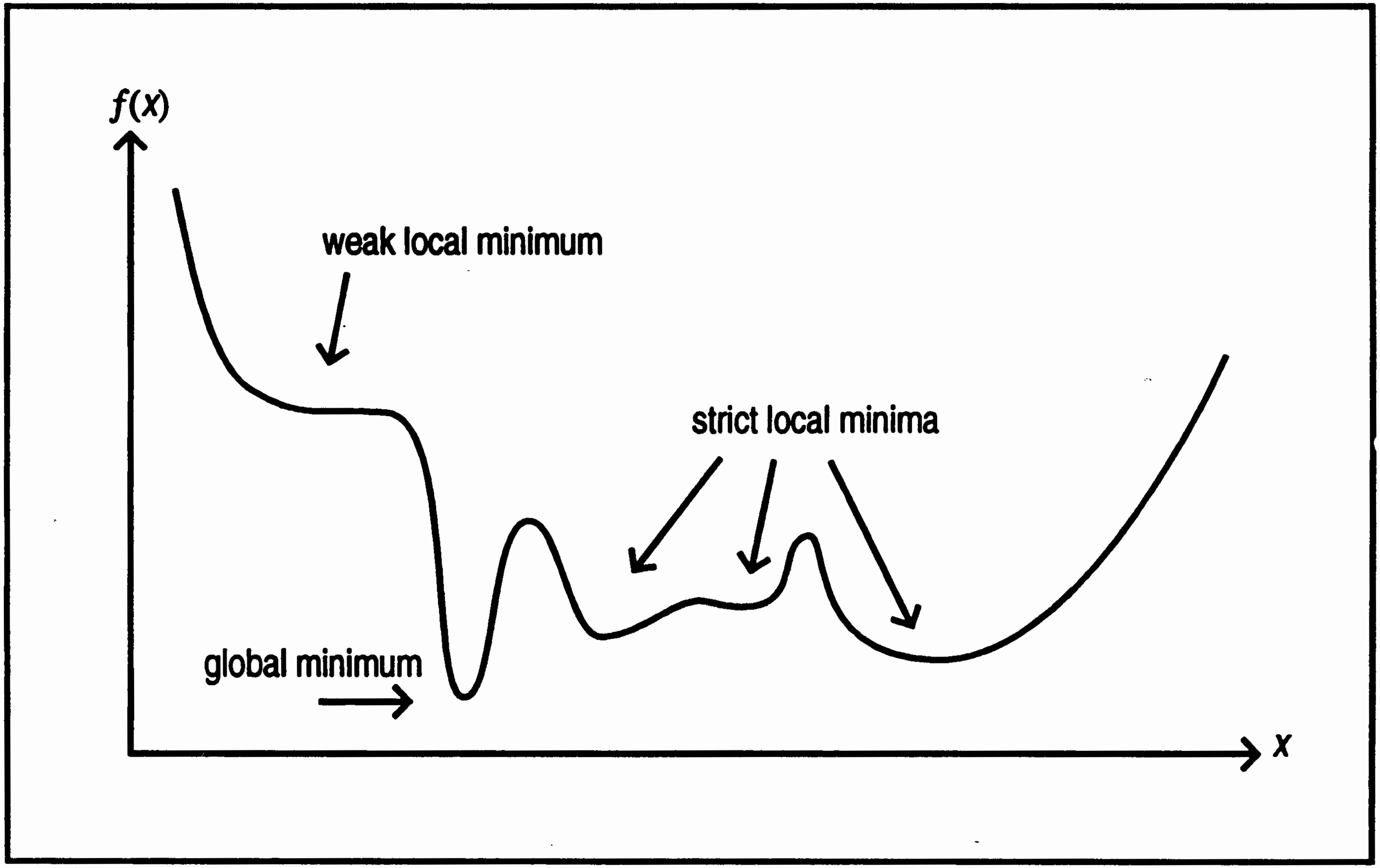
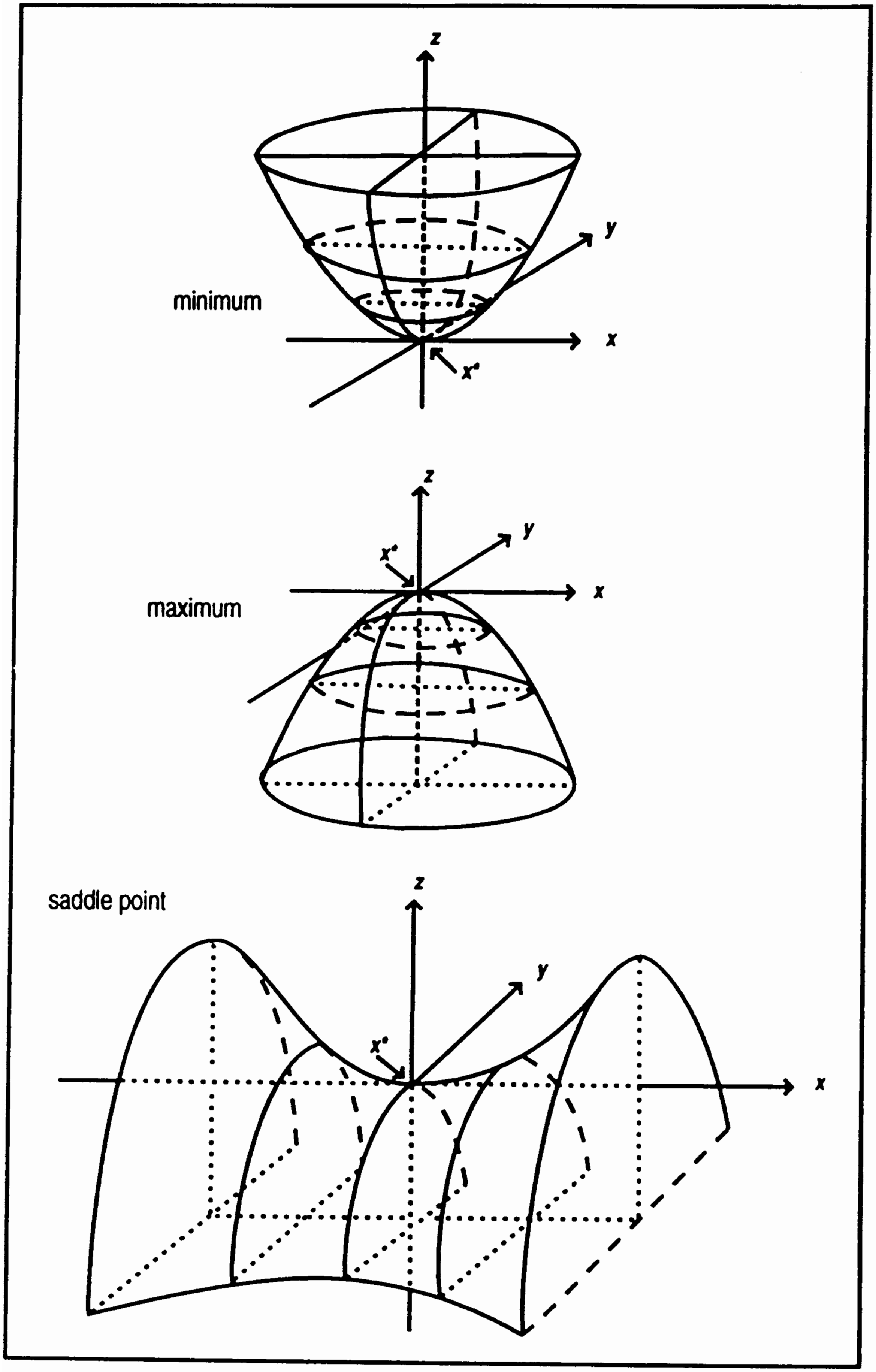


Figure 1 Types of minima.



Types of stationary points.

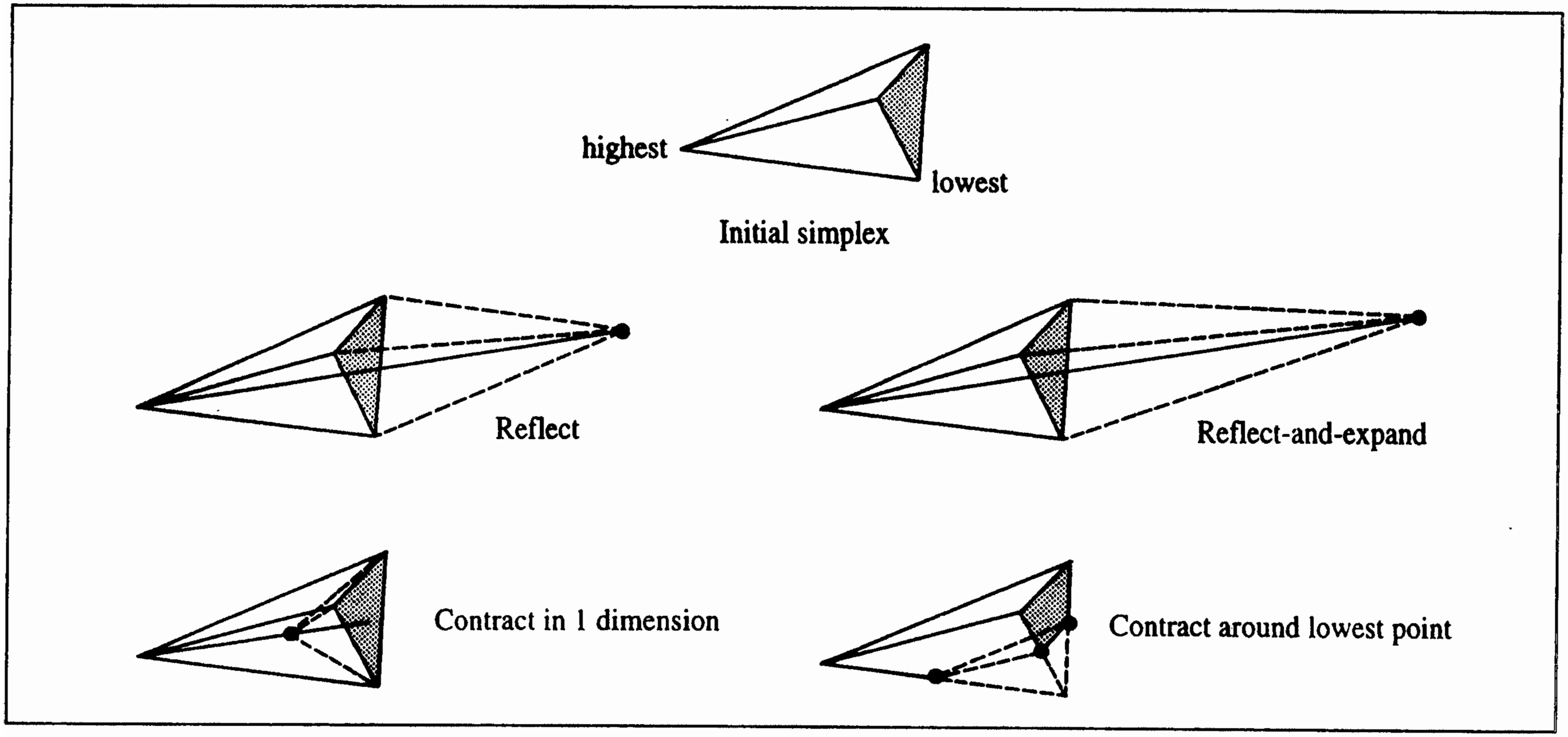
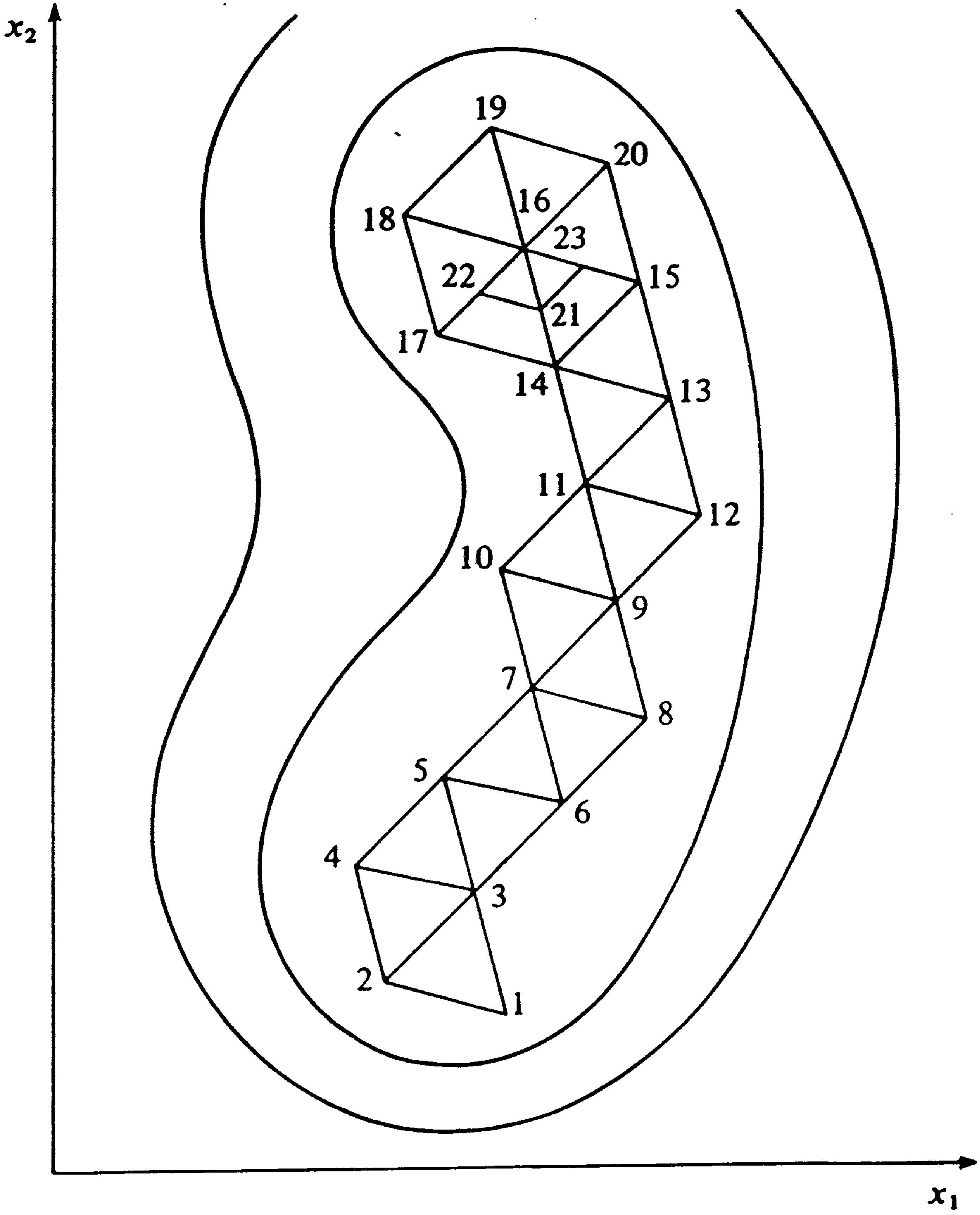
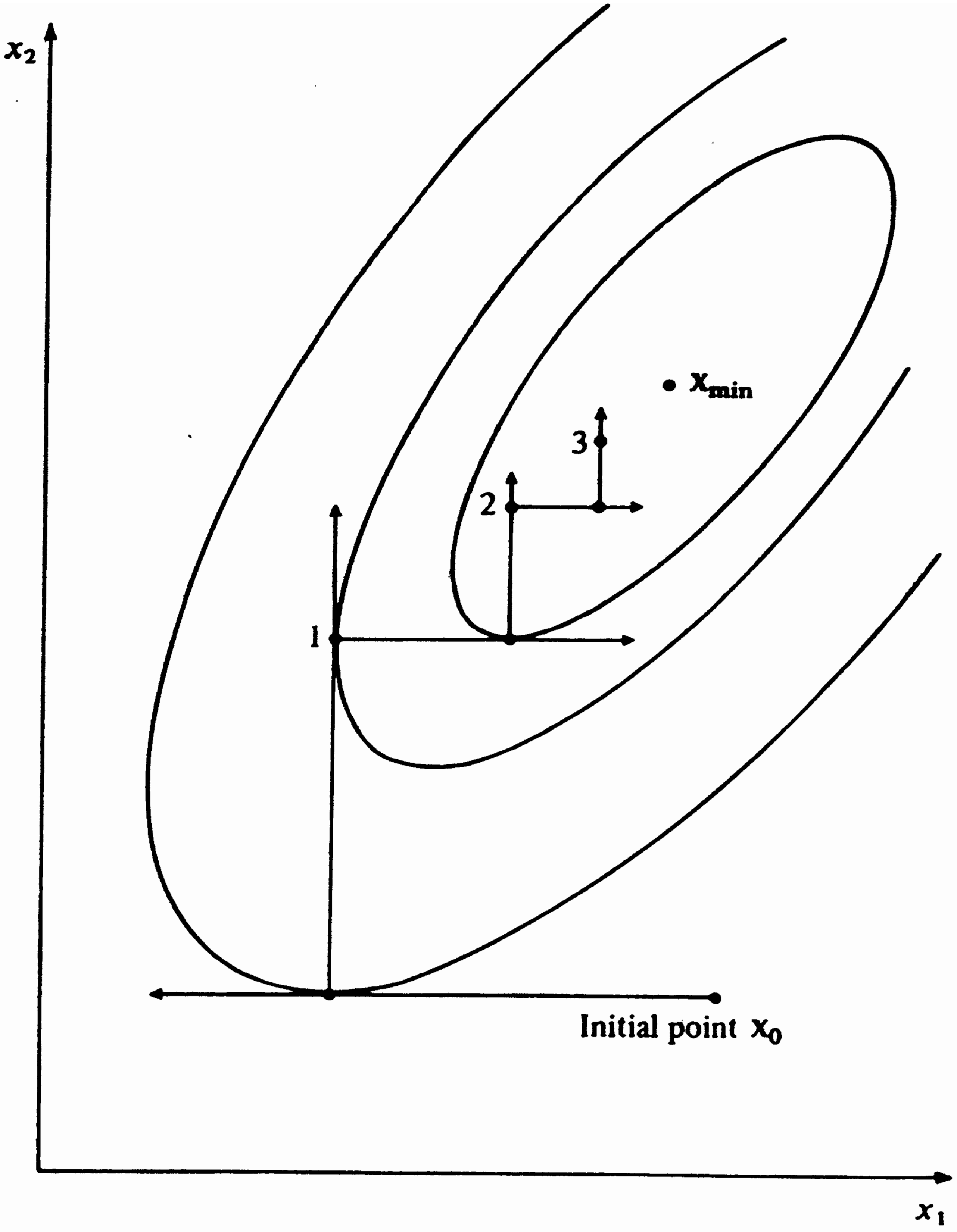


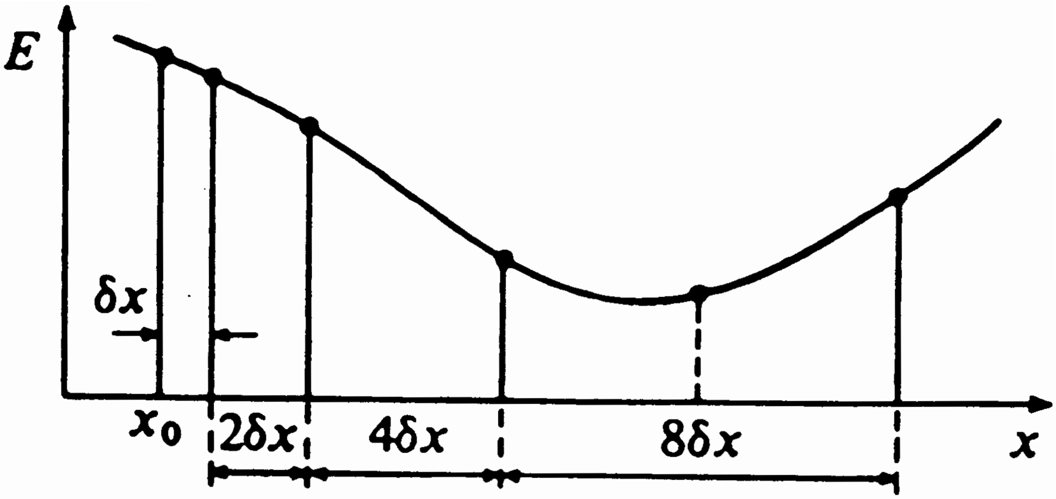
Fig. 5.4: The three basic moves permitted to the simplex algorithm (reflection, and its close relation reflect-and-expand; contract in one dimension and contract around the lowest point). (Figure adapted from Press W H, B P Flannery, S A Teukolsky and W T Vetterling 1992. Numerical Recipes in Fortran. *Cambridge, Cambridge University Press.)*



The simplex search technique.



One at a time search.



Davies, Swann, and Campey's search.

1. Input data $x_{0,1}$, δx_1, ϵ, K, f. Set $k = 0$.
2. Set $k = k + 1$.
   Evaluate f at $x_{0,k}$ and $x_{1,k}$ to give $E_{0,k} = f(x_{0,k})$ and $E_{1,k} = f(x_{1,k})$.
3. Test for positive search direction.
   If $E_{1,k} < E_{0,k}$, set $p = +1$ and go to Step 5.
   Otherwise go to Step 4.
4. Test for negative search direction.
   Evaluate f at $x_{-1,k}$.
   If $E_{-1,k} < E_{0,k}$, set $p = -1$ and go to Step 5.
   Otherwise go to Step 8.
5. Search until the minimum is spanned.
   Either search the positive direction (for $p = +1$), or search the negative direction (for $p = -1$), until $E_{n,k} > E_{n-1,k}$.
6. Insert the extra function evaluation at the centre of the final interval.
   Evaluate f at $x_{n,k} - 2^{n-2} p \delta x_k$.
7. Determine the minimum of the fitted quadratic.
   If $E_{c,k} > E_{n-1,k}$ use equation 2.33 to give $x_{0,k+1}$.
   Otherwise use equation 2.34.
   If $2^{n-2}\delta x_k \leqslant \epsilon$ go to Step 9.
   Otherwise, set $\delta x_{k+1} = K\delta x_k$ and go to Step 2.
8. Determine the minimum of the quadratic fitted to $x_{-1,k}$, $x_{0,k}$, and $x_{1,k}$.

   $$\text{Set } x_{0,k+1} = x_{0,k} + \frac{\delta x_k (E_{-1,k} - E_{1,k}}{2(E_{1,k} - 2E_{0,k} + E_{-1,k})}$$

   If $\delta x_k \leqslant \epsilon$ go to Step 9.
   Otherwise, set $\delta x_{k+1} = K\delta x_k$, and go to Step 2.
9. Output data, either δx_k or $2^{n-2}\delta x_k$,

   $x_{\min} = x_{0,k+1}$,

   $E_{\min} = E_{0,k+1}$

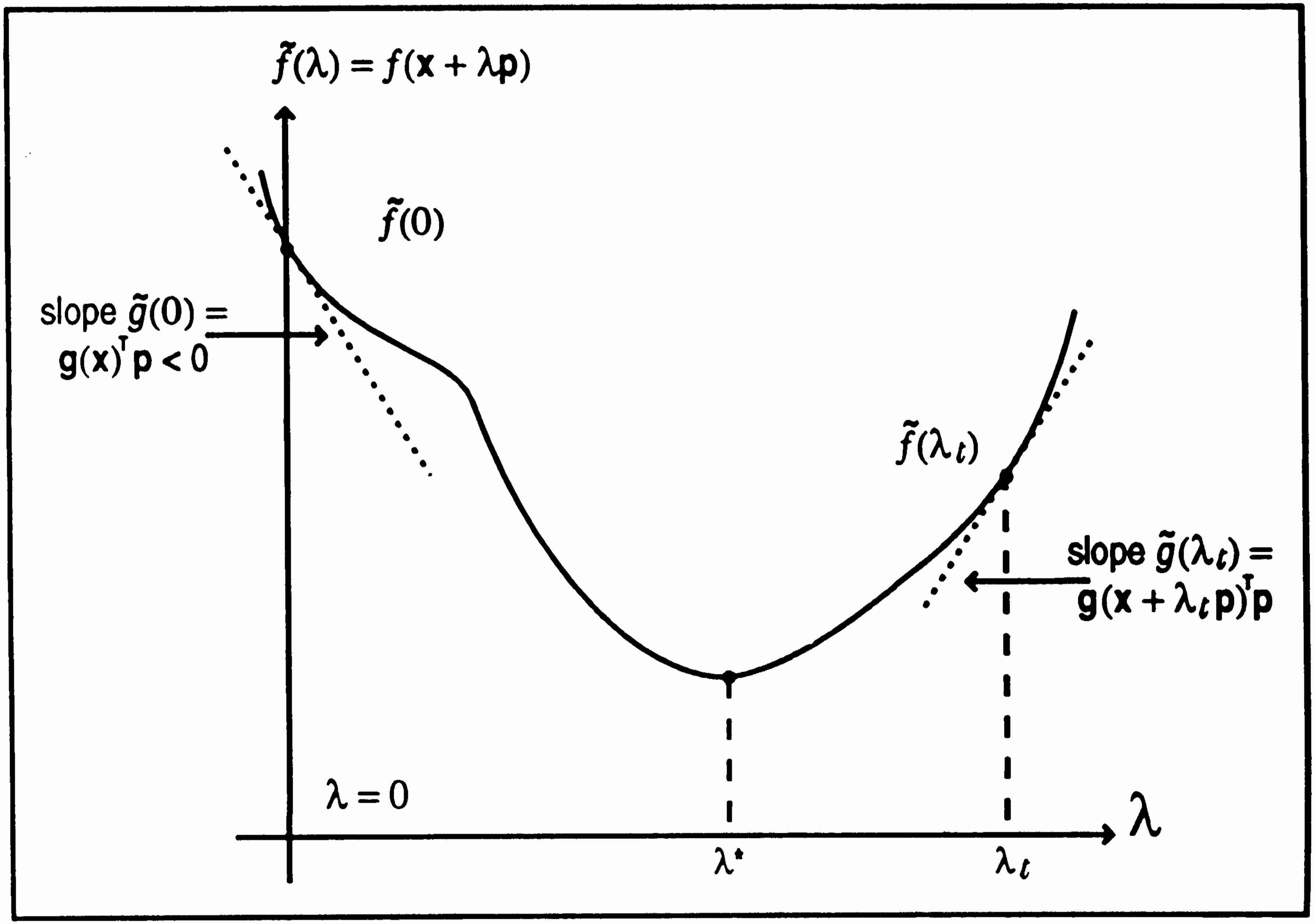

$\tilde{f}(\lambda) = f(\mathbf{x} + \lambda\mathbf{p})$
$\tilde{f}(0)$
slope $\tilde{g}(0) =$
$\mathbf{g}(\mathbf{x})^T\mathbf{p} < 0$
$\tilde{f}(\lambda_t)$
slope $\tilde{g}(\lambda_t) =$
$\mathbf{g}(\mathbf{x} + \lambda_t\mathbf{p})^T\mathbf{p}$
$\lambda = 0$
λ
λ^*
λ_t

Figure 9 A one-dimensional line search for $f\bar{(}\lambda) = f(\mathrm{x} + \lambda\mathrm{p})$ by cubic interpolation produces an approximate minimum along **p**, at $\lambda^*, f\bar{(}\lambda^*)$.

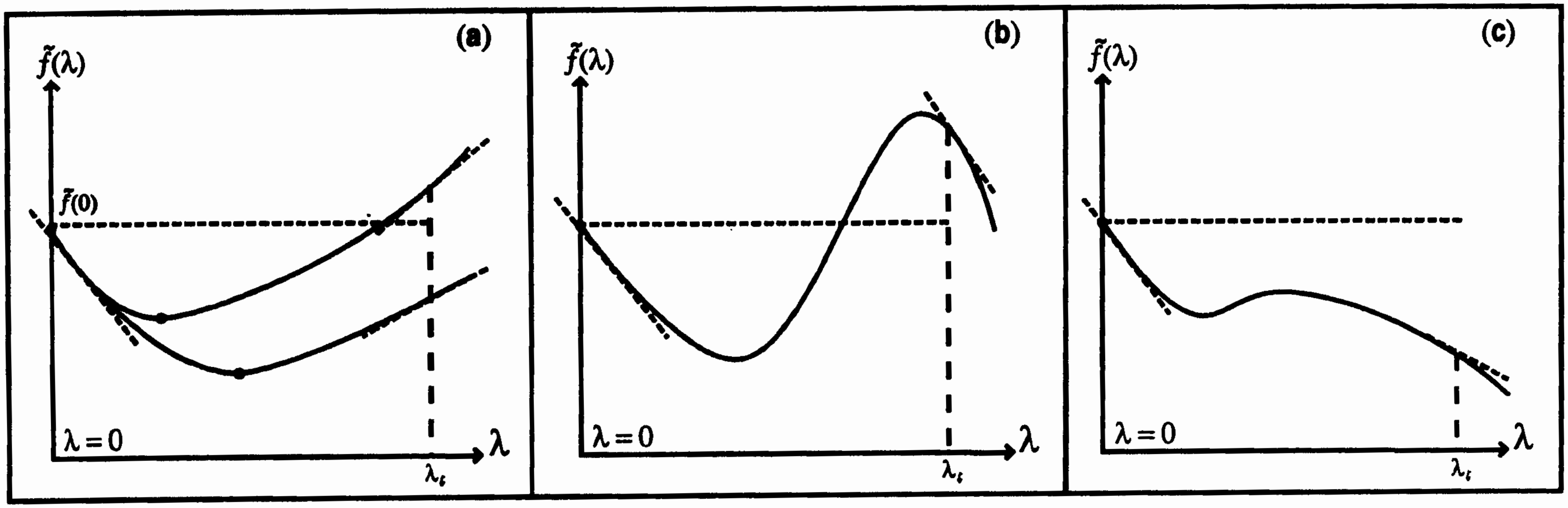
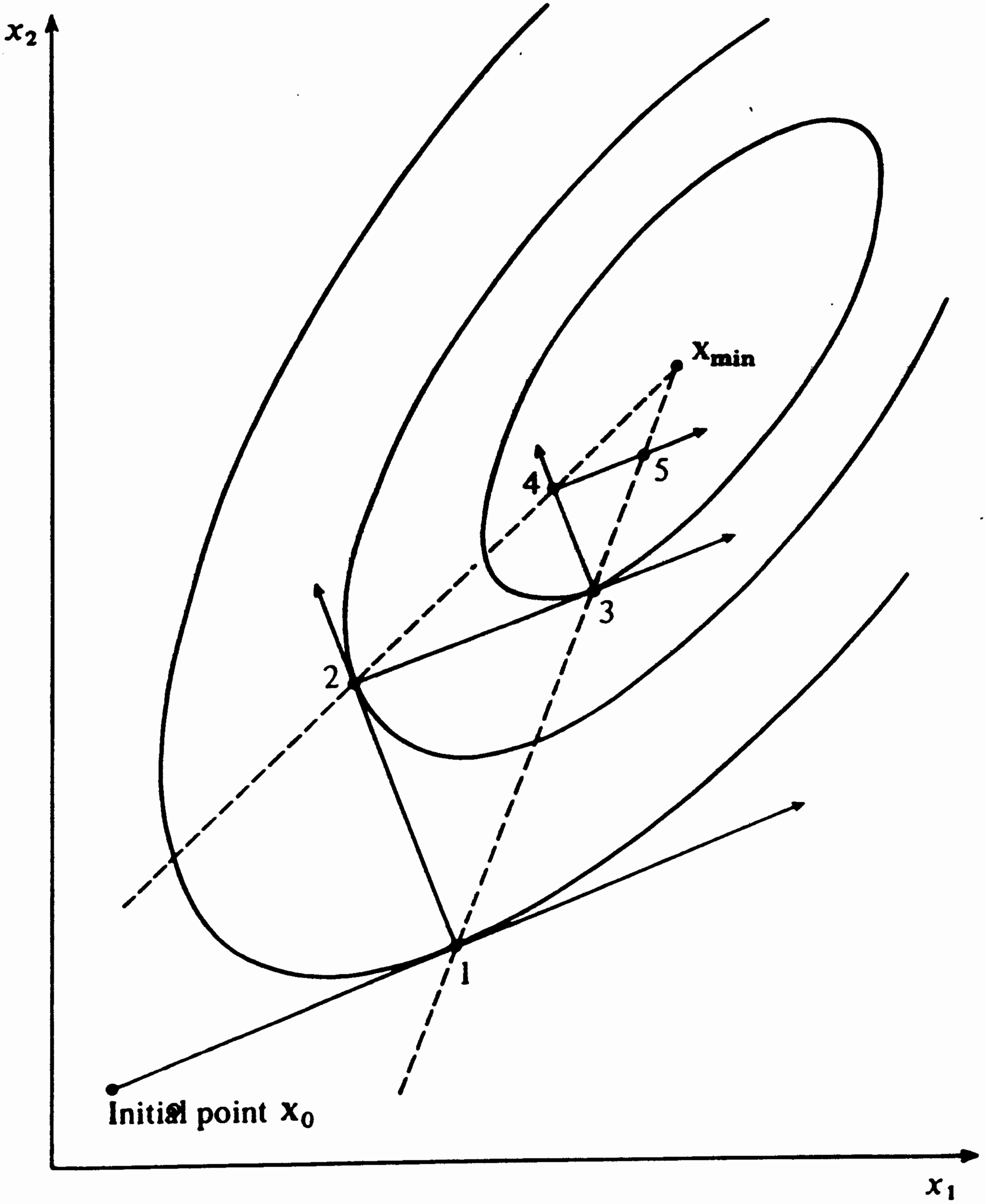
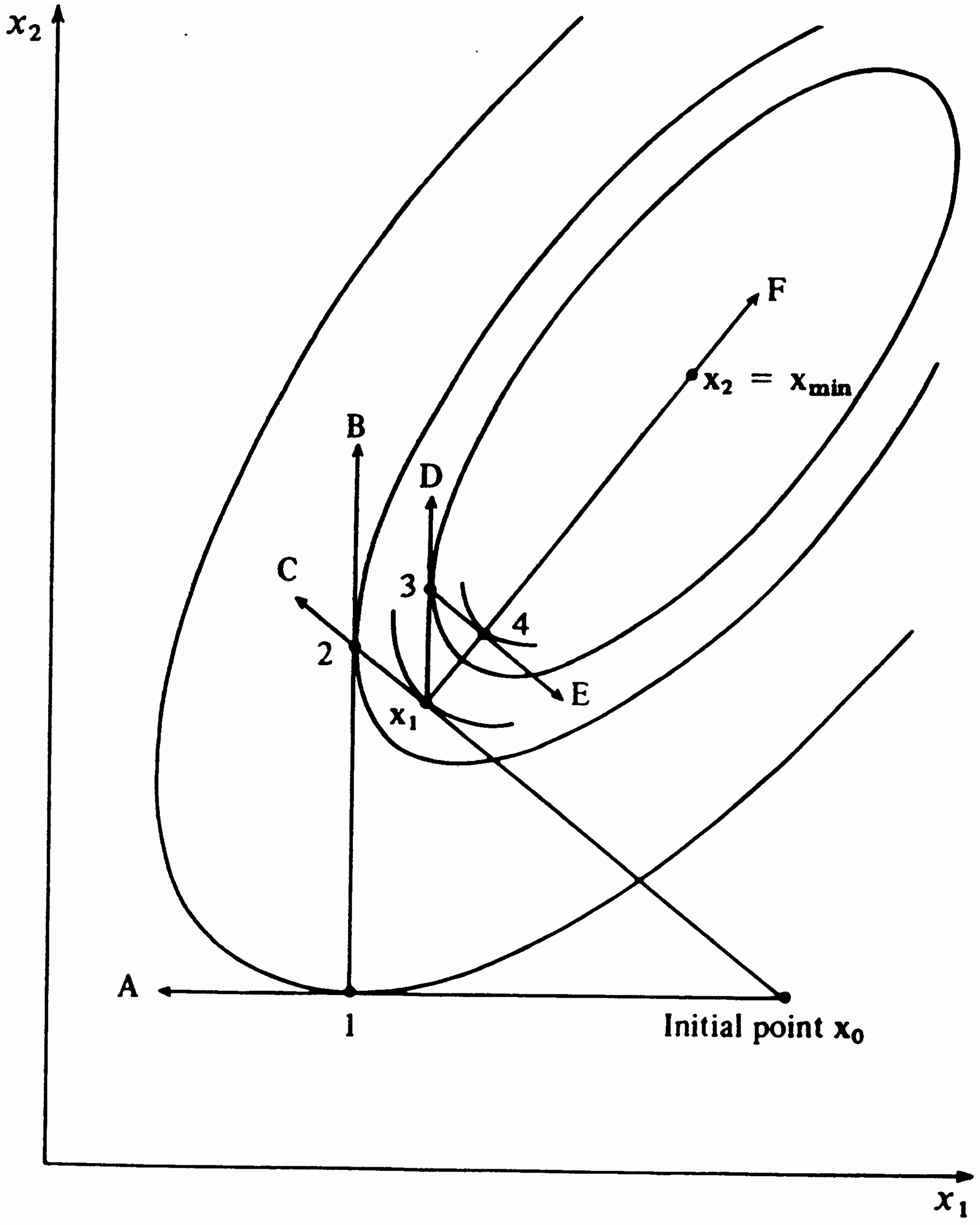


Figure 10 Possible situations in line search algorithms between the current and trial points: (a) The new slope is positive. (b) The new slope is negative but function value is greater. (c) The new slope is negative and function value is lower.



Steepest descent minimization.


x_2
F
$\mathbf{x}_2 = \mathbf{x}_{min}$
B
D
C
3
4
2
E
$\mathbf{x}_1$
A
1
Initial point $\mathbf{x}_0$
x_1

Search by the method of conjugate directions.

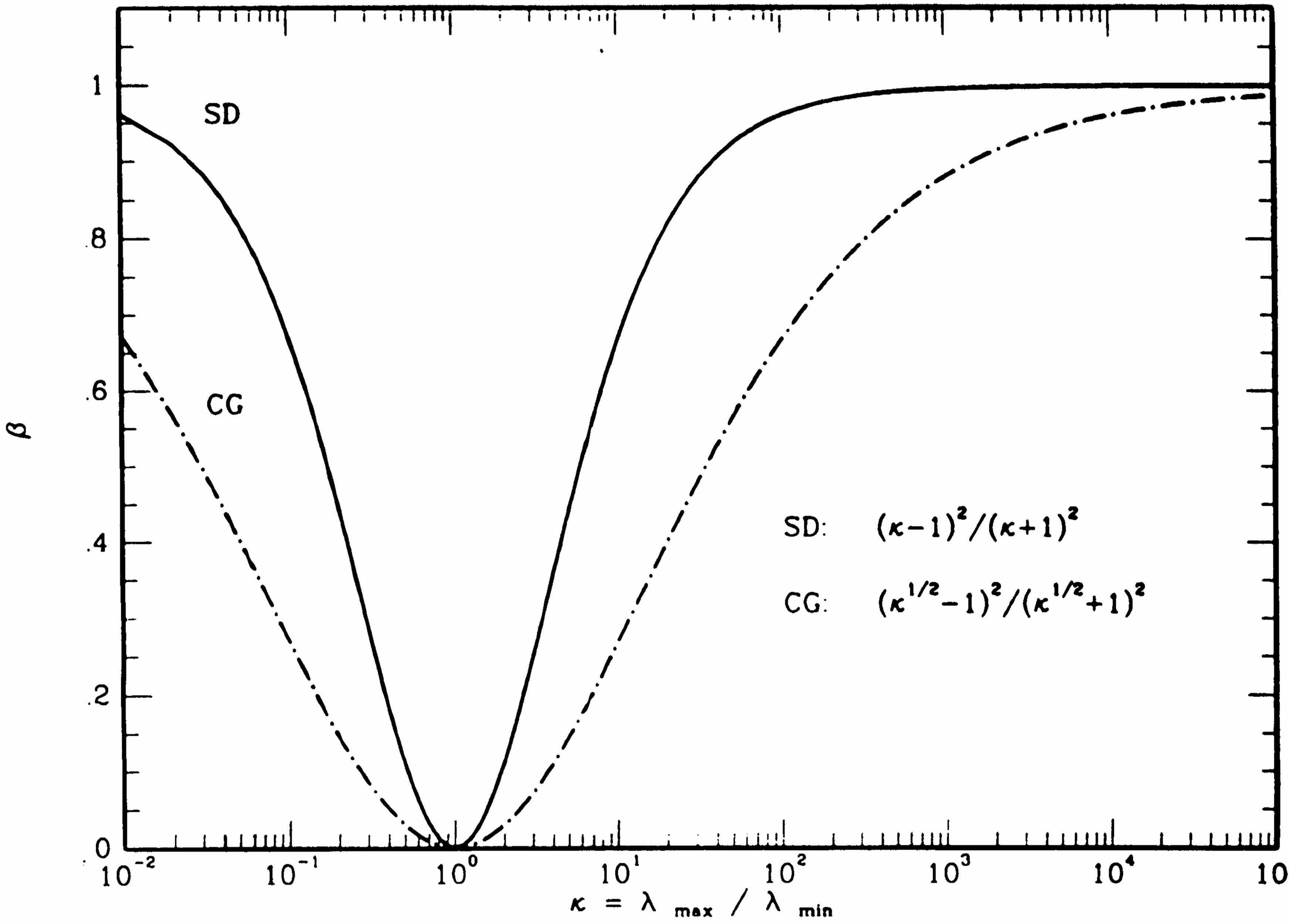


Figure 12 Steepest descent and conjugate gradient quantities that affect the convergence rate for quadratic functions (see text for the distinct context of these functions).

Algorithm [A1]: Basic Descent

- Supply an initial guess $\mathbf{x}_0$.
- For $k = 0, 1, 2, \ldots$, until convergence
  1. Test $\mathbf{x}_k$ for convergence.
  2. Calculate a search direction $\mathbf{p}_k$.
  3. Determine an appropriate step length λ_k (or modified step $\mathbf{s}_k$).
  4. Set $\mathbf{x}_{k+1}$ to $\mathbf{x}_k + \lambda_k \mathbf{p}_k$ (or $\mathbf{x}_k + \mathbf{s}_k$).

Algorithm [A4]: Modified Newton

- For $k = 0, 1, 2, \ldots$, until convergence, given $\mathbf{x}_0$,

1. Test $\mathbf{x}_k$ for convergence.
2. Compute a descent direction $\mathbf{p}_k$ so that

   $$\|H_k \mathbf{p}_k + \mathbf{g}_k\| \leq \eta_k \|\mathbf{g}_k\|,$$

   where η_k controls the accuracy of the solution and some symmetric matrix $\overline{H}_k$ may approximate H_k.
3. Compute a step length λ so that for $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda \mathbf{p}_k$,

   $$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \alpha \lambda \mathbf{g}_k{}^T \mathbf{p},$$

   $$|\mathbf{g}_{k+1}{}^T \mathbf{p}_k| \leq \beta |\mathbf{g}_k{}^T \mathbf{p}_k|\,,$$

   with $0 < \alpha < \beta < 1$.
4. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda \mathbf{p}_k$.

Algorithm [A2]: CG Method to Solve $A\mathbf{x} = -\mathbf{b}$

```
1. Set $\mathbf{r}_0 = -(A\mathbf{x}_0 + \mathbf{b})$, $\mathbf{d}_0 = \mathbf{r}_0$.
2. For $k = 0,1,2, \ldots$, until $\mathbf{r}$ is sufficiently small, compute
       $\lambda_k = \mathbf{r}_k^T\mathbf{r}_k/\mathbf{d}_k^T A\mathbf{d}_k$
       $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k$
       $\mathbf{r}_{k+1} = \mathbf{r}_k - \lambda_k A\mathbf{d}_k$
       $\beta_k = \mathbf{r}_{k+1}^T\mathbf{r}_{k+1}/\mathbf{r}_k^T\mathbf{r}_k$
       $\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{d}_k$.
```

Algorithm [A3]: PCG Method to Solve $A\mathbf{x} = -\mathbf{b}$

```
1. Set $\mathbf{r}_0 = -(A\mathbf{x}_0 + \mathbf{b})$, $\mathbf{d}_0 = M^{-1}\mathbf{r}_0$.
2. For $k = 0,1,2, \ldots$, until $\mathbf{r}$ is sufficiently small, compute
       $\lambda_k = \mathbf{r}_k^T(M^{-1}\mathbf{r}_k)/\mathbf{d}_k^T A\mathbf{d}_k$
       $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k$
       $\mathbf{r}_{k+1} = \mathbf{r}_k - \lambda_k A\mathbf{d}_k$
       $\beta_k = \mathbf{r}_{k+1}^T(M^{-1}\mathbf{r}_{k+1})/\mathbf{r}_k^T(M^{-1}\mathbf{r}_k)$
       $\mathbf{d}_{k+1} = (M^{-1}\mathbf{r}_{k+1}) + \beta_k\mathbf{d}_k$.
```

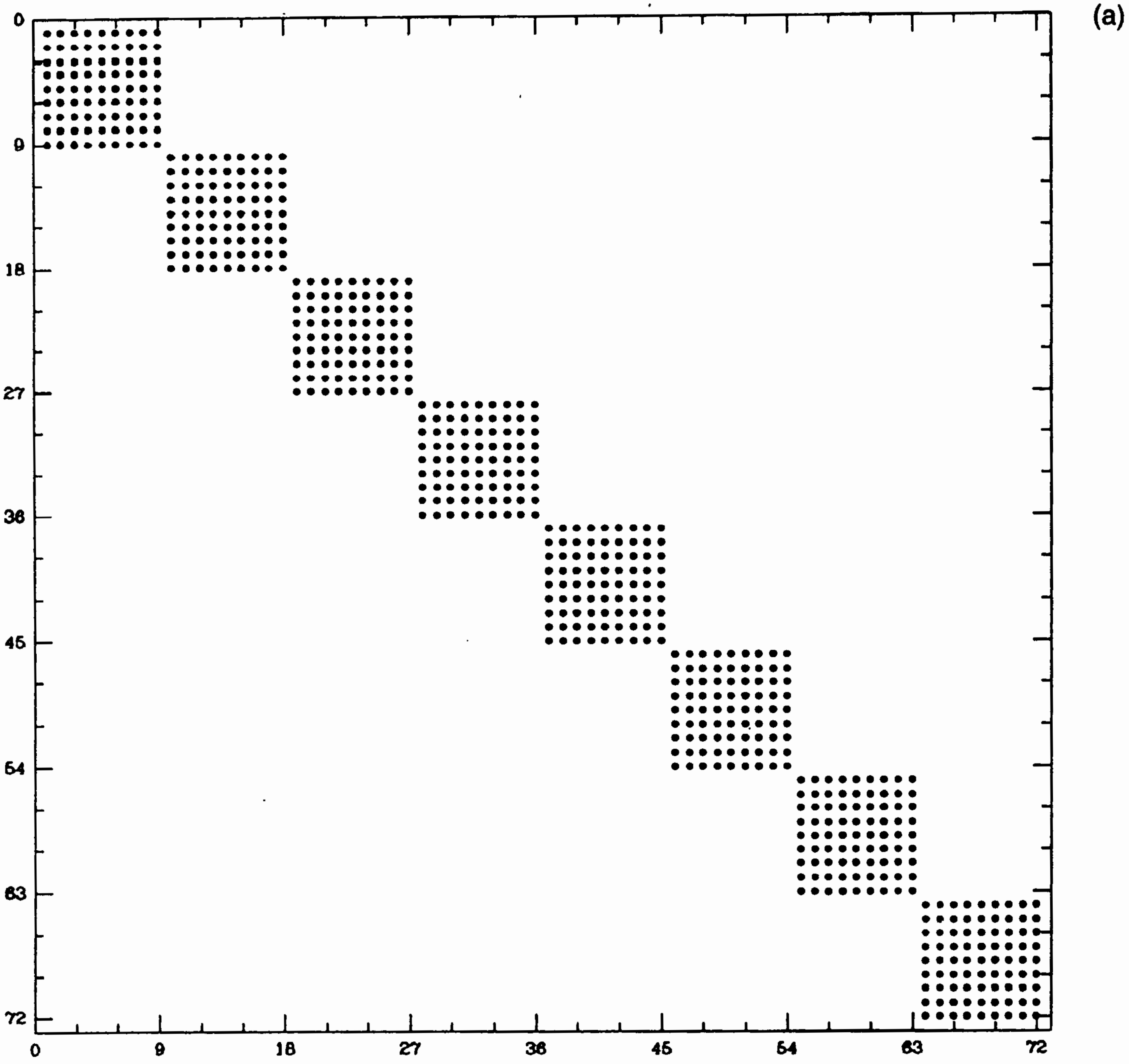


Figure 2 Sample matrix patterns for (a) block diagonal and (b—e) sparse unstructured. Pattern (b) corresponds to the Hessian approximation (preconditioner) for a potential energy model from the local energy terms (bond length, bond angle, and dihedral angle terms), and (c) is a reordered matrix pattern that reduces fill-in during the factorization. Pattern (d) comes from a molecular dynamics simulation of supercoiled DNA[36] and describes pairs of points along a ribbonlike model of the duplex that come in close contact during the dynamics trajectory; pattern (e) is the associated reordered structure that reduces fill-in.

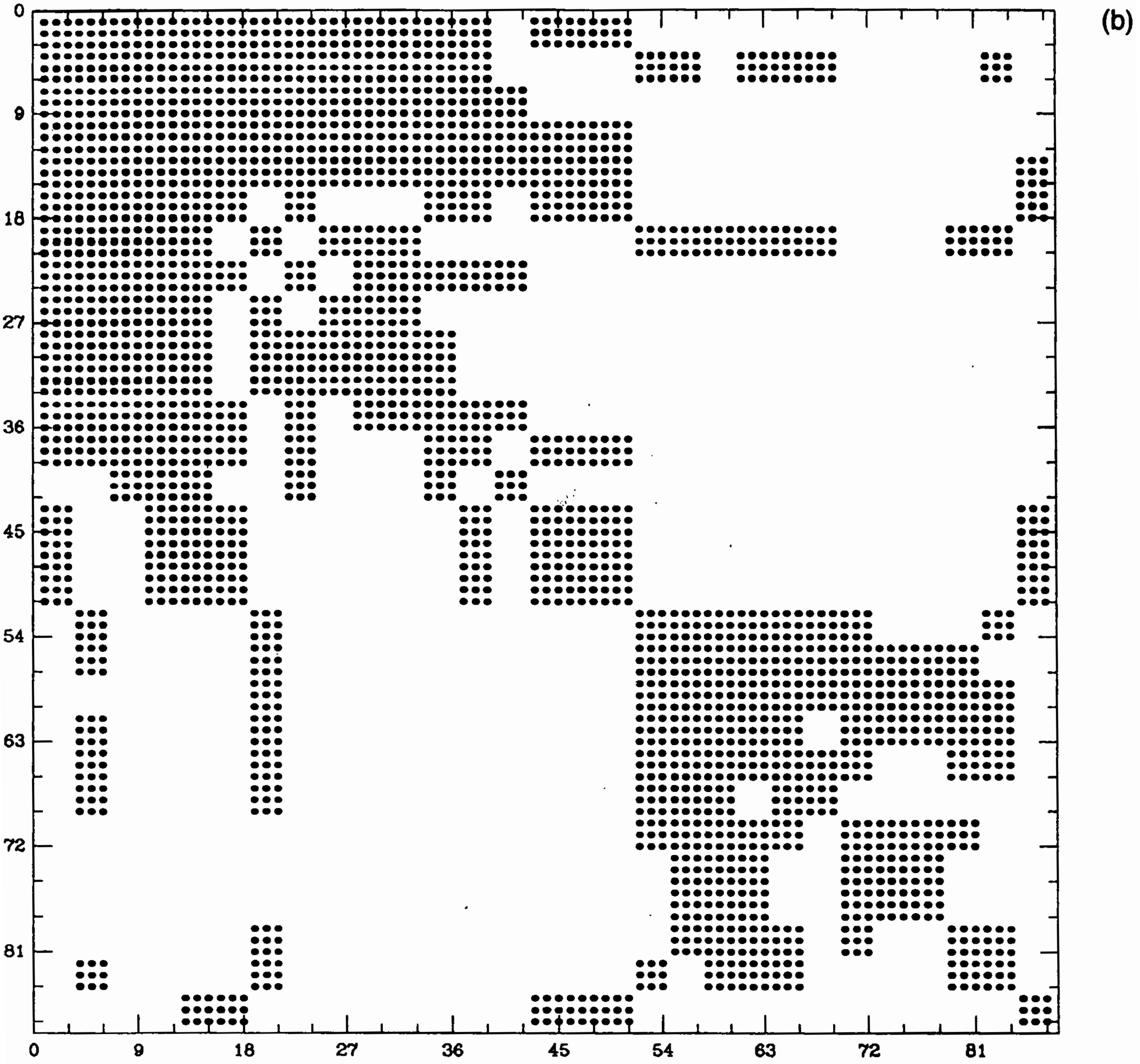


Figure 2 *(continued)*

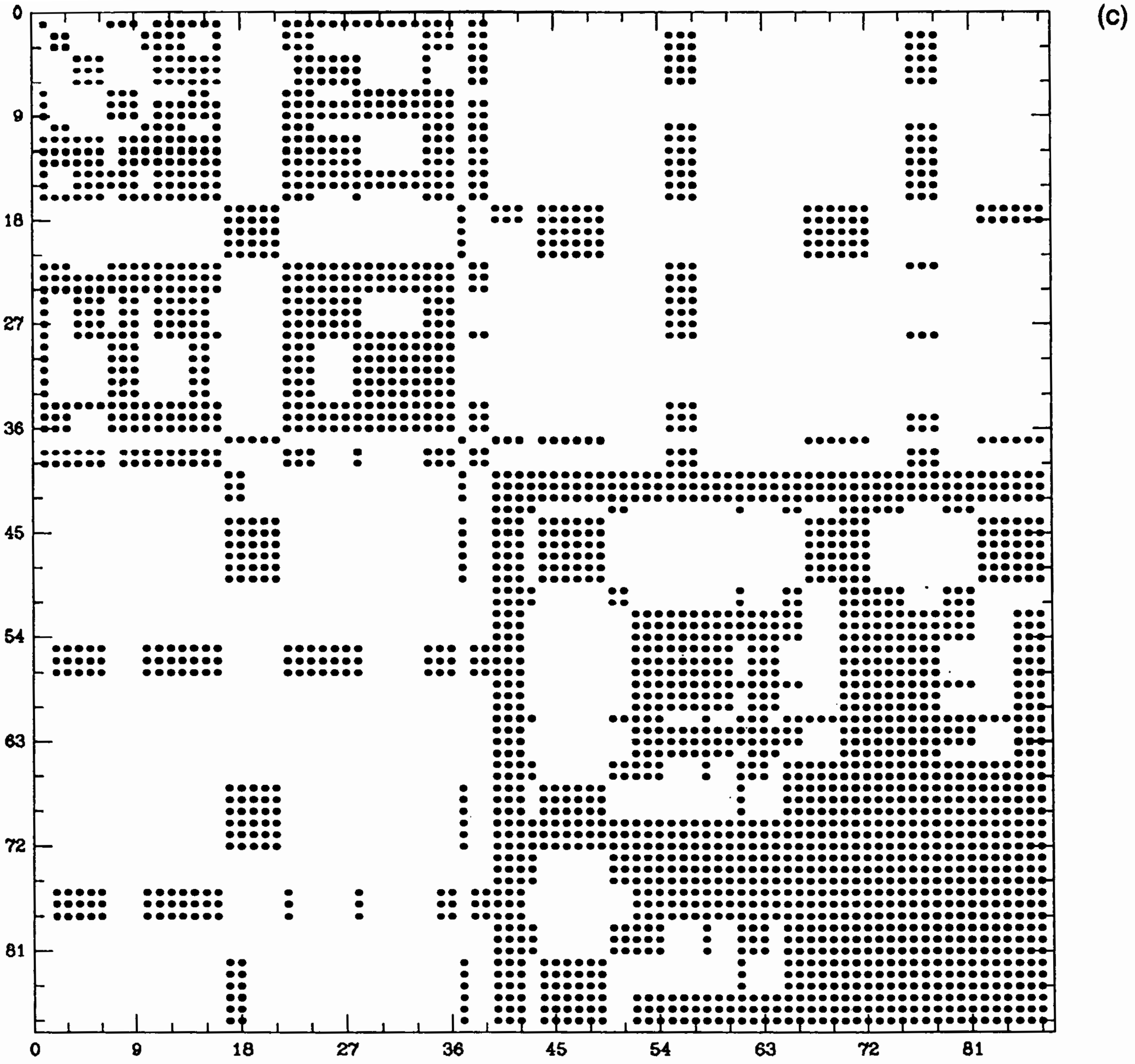
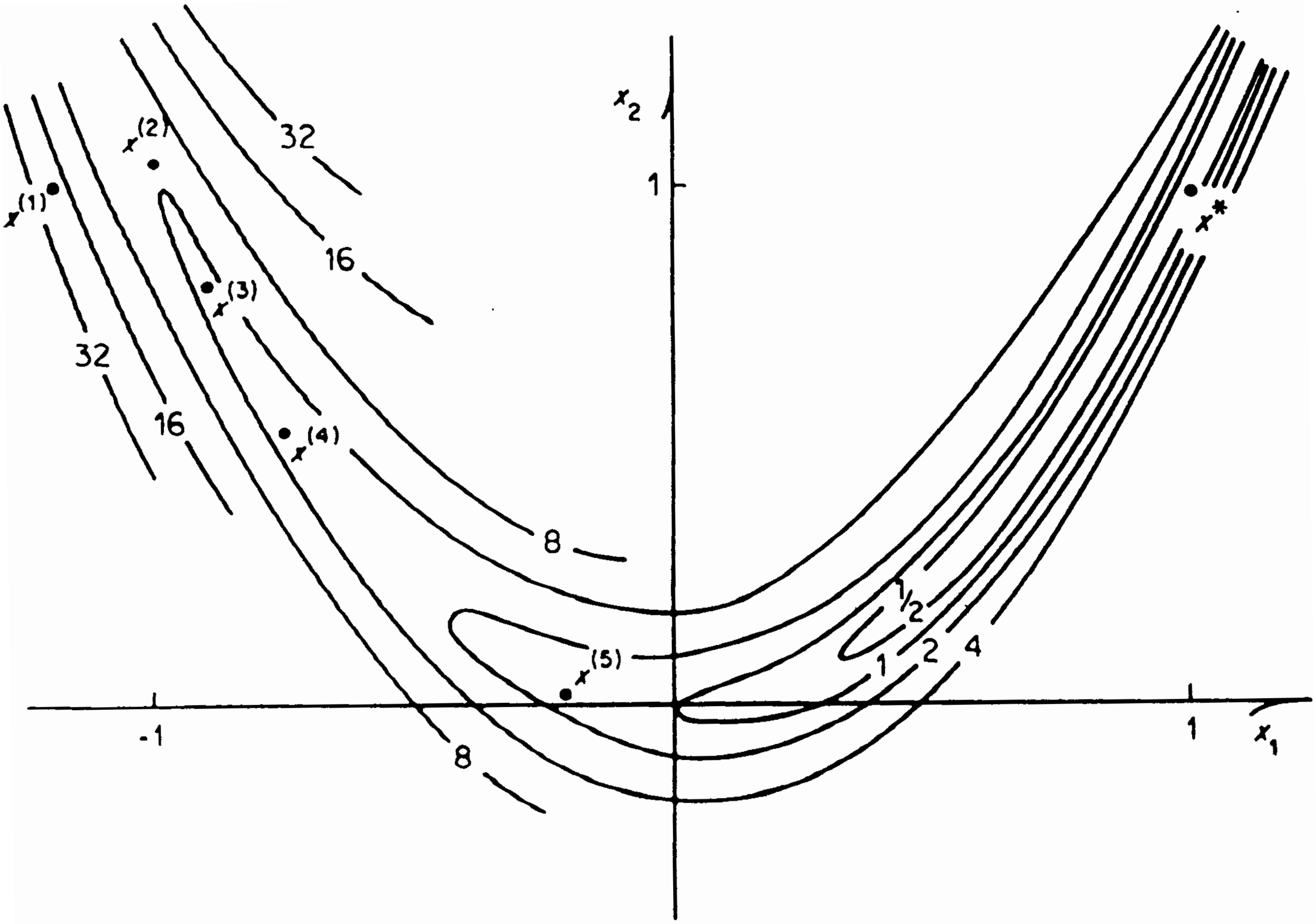


Figure 2 *(continued)*



Contours for Rosenbrock's function

Numerical Example I: Rosenbrock Minimization

Rosenbrock's function is often used as a minimization test problem, because its minimum lies at the base of a "banana-shaped valley" and can be difficult to locate. This function is defined for even integers n as the sum

$$f(\mathbf{x}) = \sum_{j=1,3,5,\ldots,n-1} [(1 - x_j)^2 + 100(x_{j+1} - x_j^2)^2].$$

The contour plot of Rosenbrock's function for $n = 2$ is shown in Figure 14. The minimum point is (1,1), where $f(\mathbf{x}) = 0$. The gradient components of this function are given by

$$\left.\begin{aligned} g_{j+1} &= 200(x_{j+1} + x_j^2) \\ g_j &= -2[x_j g_{j+1} + (1 - x_j)] \end{aligned}\right\}, \qquad j = 1,3,5,\ldots,n - 1,$$

and the Hessian is the 2 × 2 block diagonal matrix with entries

$$\left.\begin{aligned} H_{j+1,j+1} &= 200 \\ H_{j+1,j} &= -400x_j \\ H_{j,j} &= -2(x_j H_{j+1,j} + g_{j+1} - 1) \end{aligned}\right\}, \qquad j = 1,3,5,\ldots,n - 1.$$

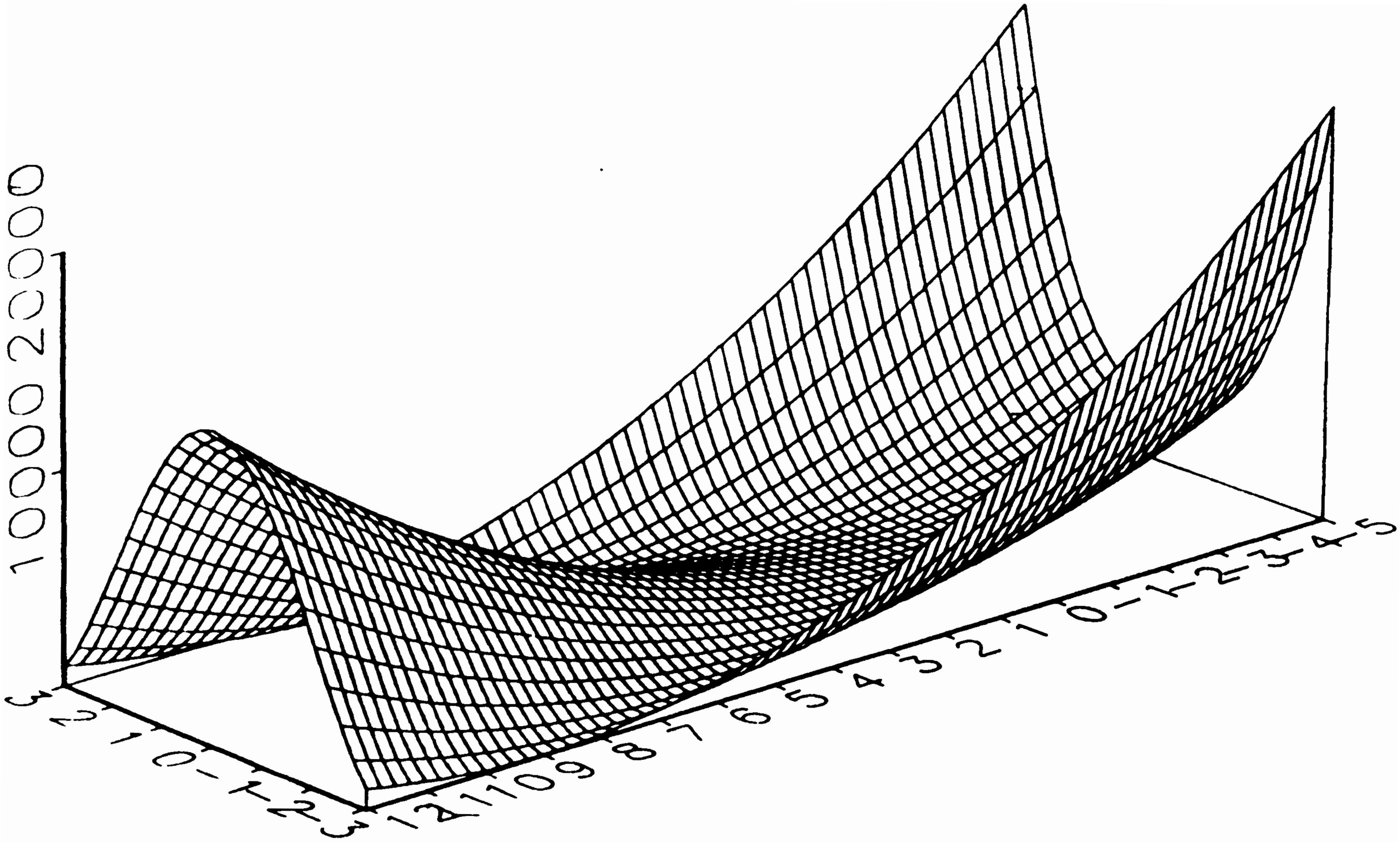
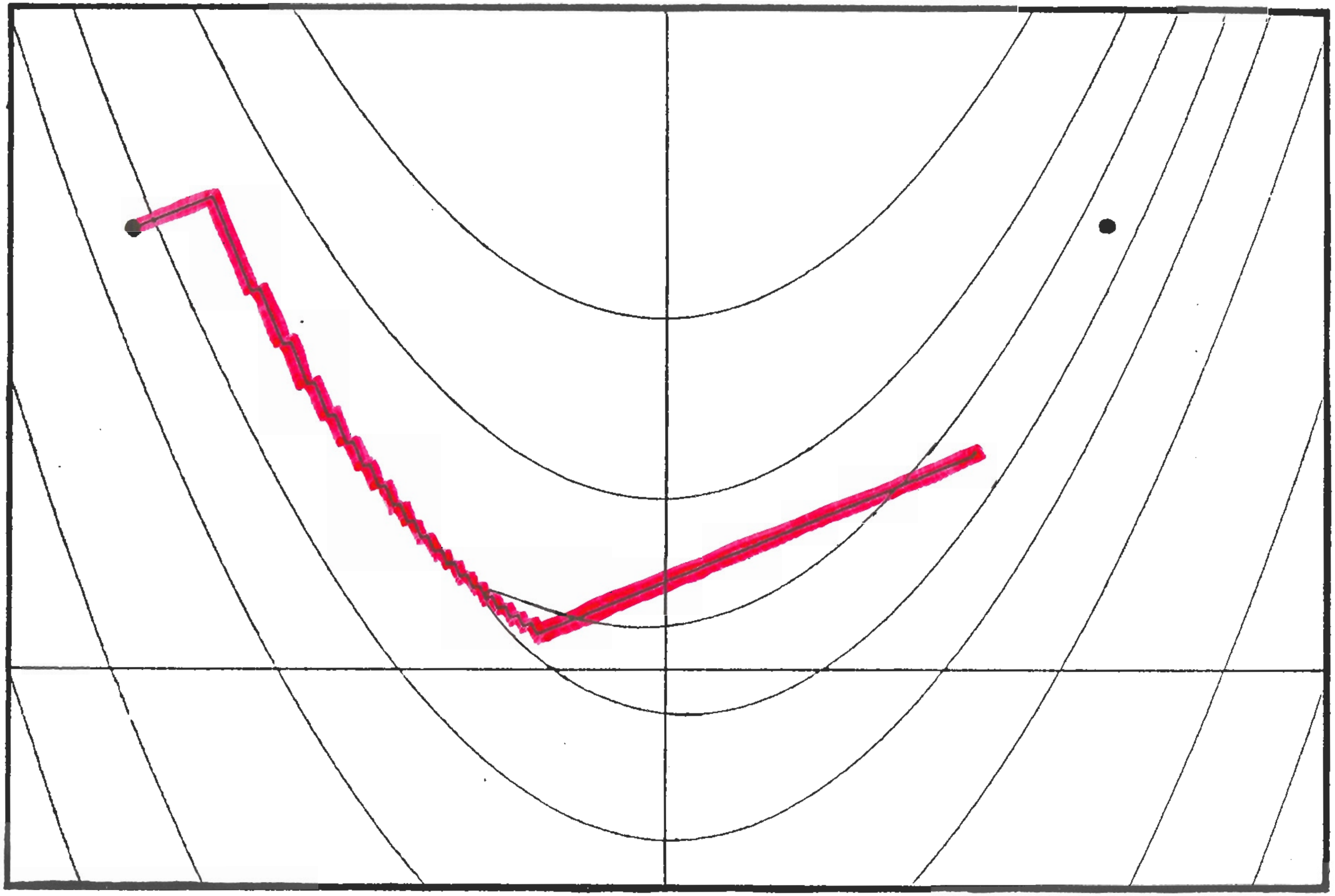
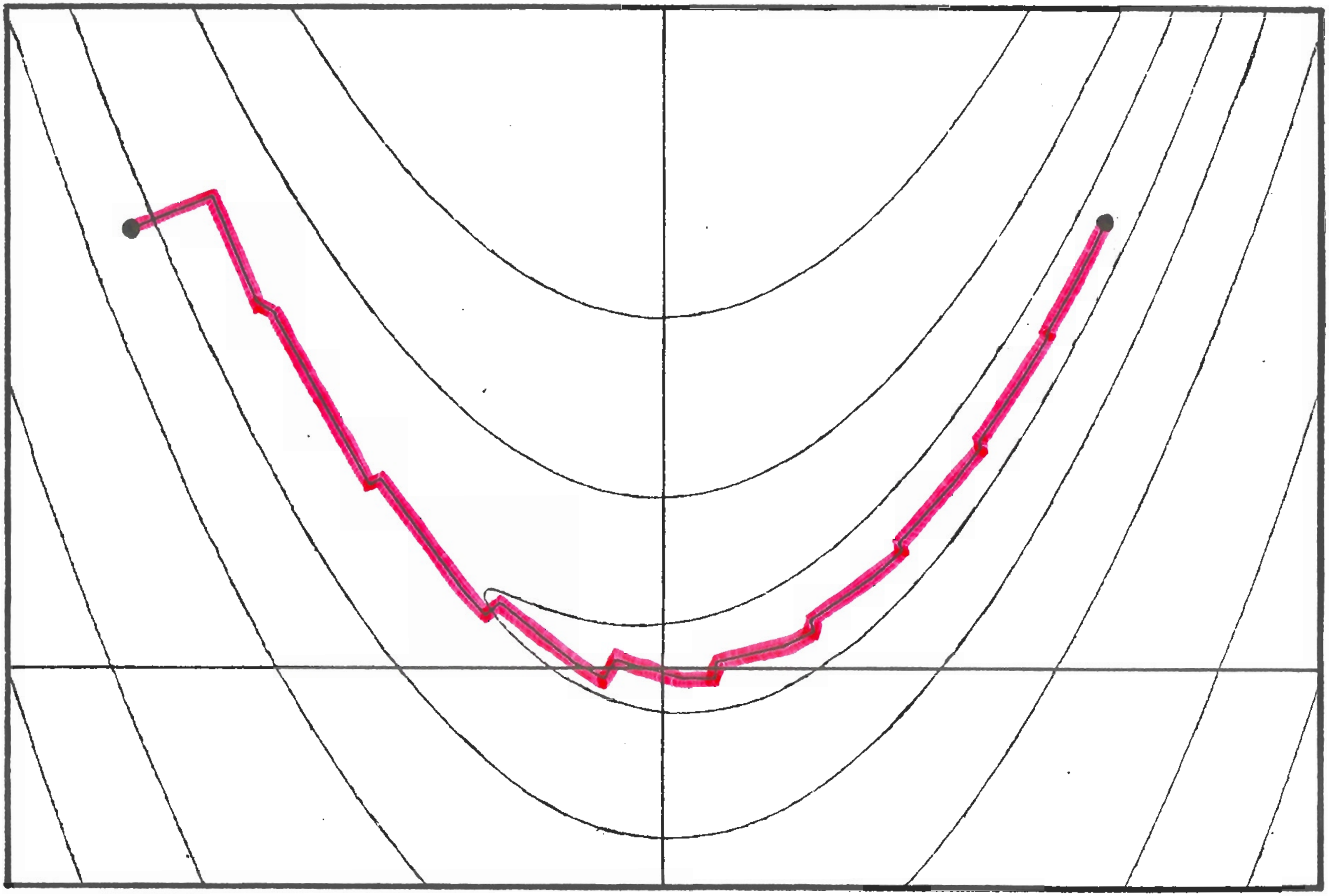


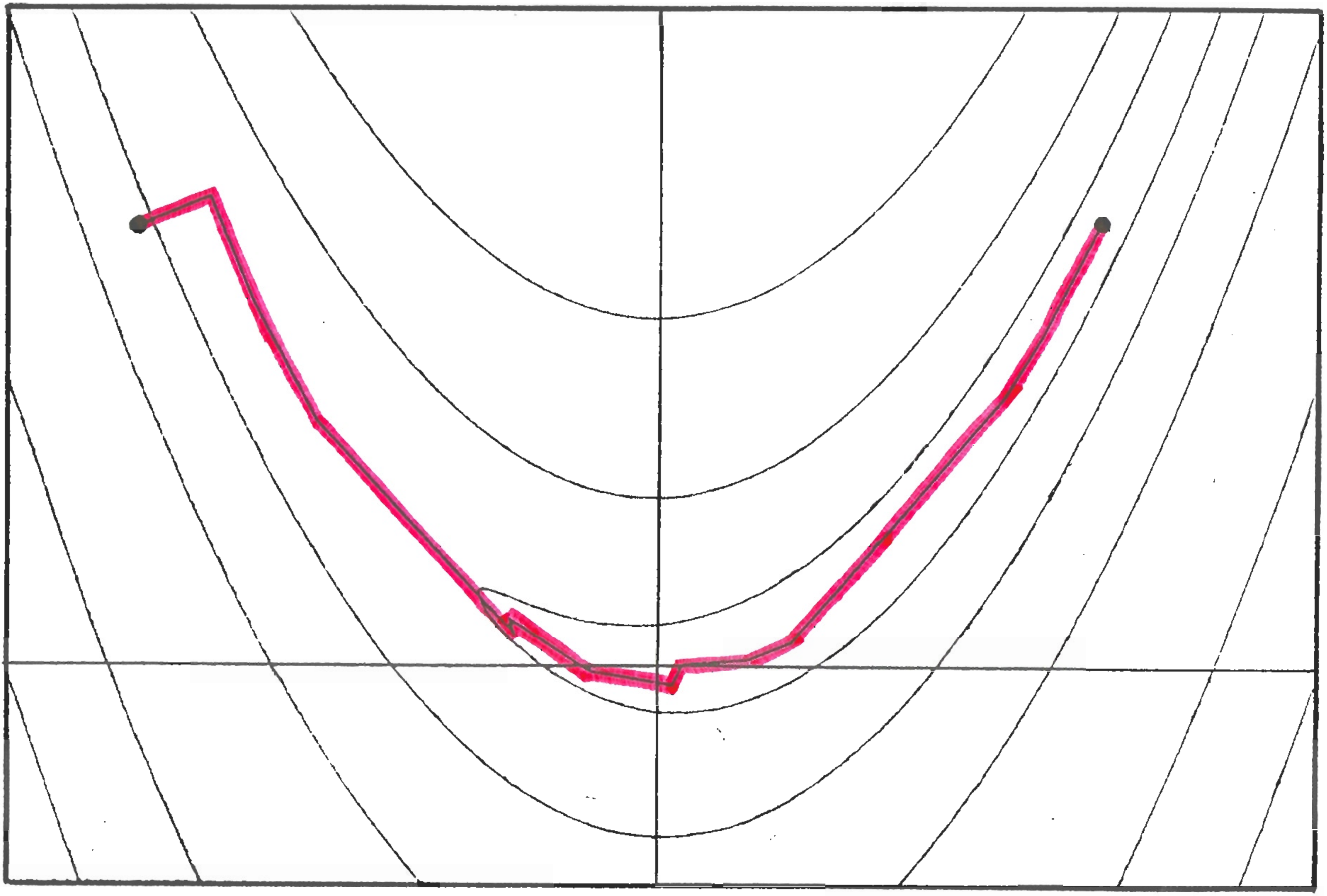
Fig. 1. Surface plot of the Rosenbrock function, $n = 2$.



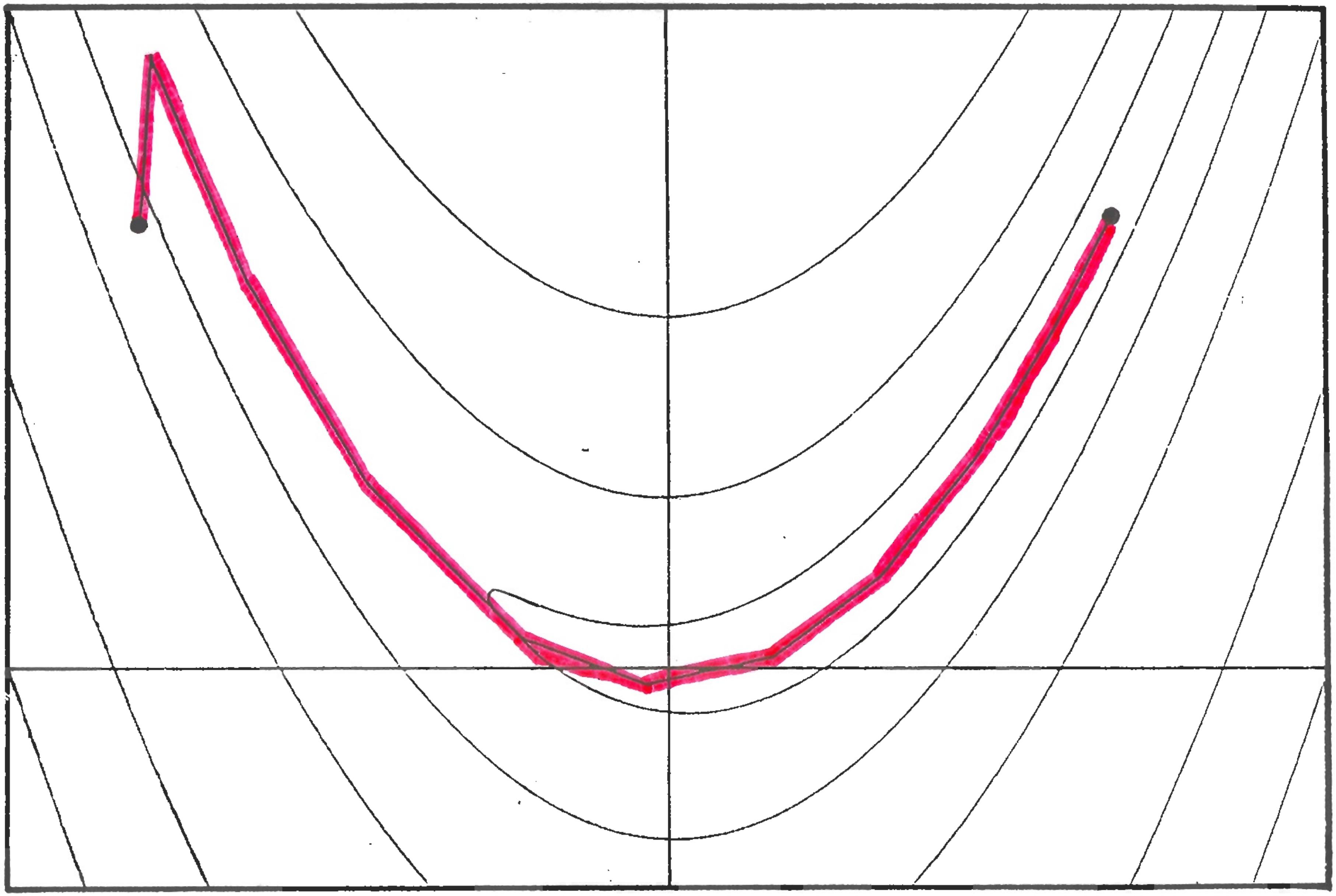
steepest-descent algorithm



conjugate-gradient algorithm



BFGS quasi-Newton algorithm



modified Newton algorithm

Table 1 Comparison Among Five Minimization Methods[a]

	Classic Newton	Nonlinear Conjugate Gradient	Quasi-Newton (Full Memory)	Quasi-Newton (Limited Memory)	Truncated Newton
Tasks per iteration					
Form **g**	αn	αn	αn	αn	αn
Form H	$(\alpha/2)n(n+1)$				
Form M					$m(\alpha n)$
Factor M					$O(l) - (l^2)$
Calculate **p**	$\frac{1}{6}n^3$	$7n$	$O(n^2)$	$23n$ $[+ O(l) - O(l^2)]$	$IT(\alpha n + 7n + O(l))$ $< \alpha n$
Perform line search		$\sim (2\text{–}4)\alpha n$	αn	αn	
Storage	$O(n^2)$	$3n-7n$	$O(n^2)$	$14n$ $[+ O(\max\{n,m\})]$	$O(\max\{n, m\})$
Advantages	Locally quadratically convergent	Easy to implement Modest storage and computation cost	Locally superlinearly convergent	Adapts well to available storage	Exploits problem structure to accelerate convergence (by preconditioning) Can display local quadratic convergence
Disadvantages	Requires $O(n^2)$ storage and $O(n^3)$ work	Convergence often slow Requires fairly accurate line search	Requires $O(n^2)$ storage		Requires construction and factorization of preconditioner Performance may be slow for highly nonlinear functions when directions of negative curvature are detected repeatedly

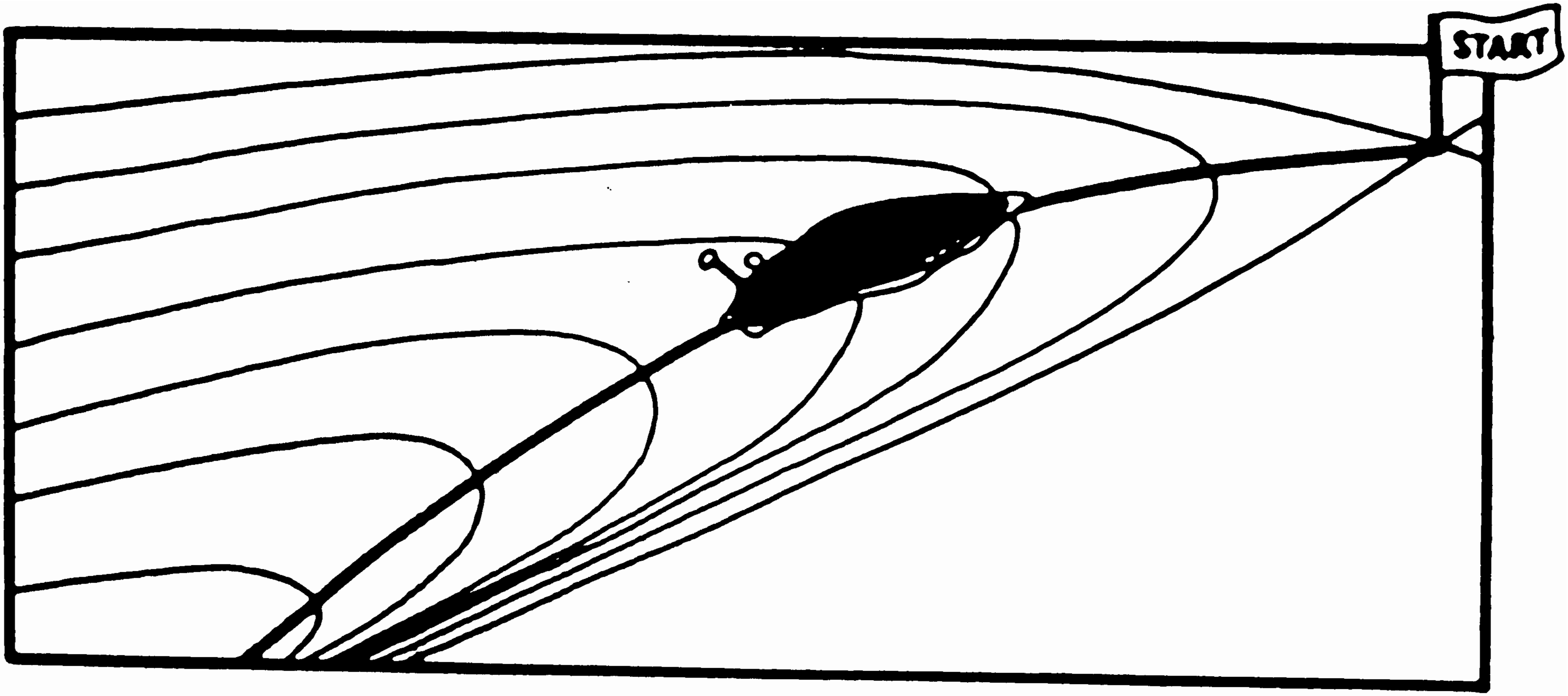


Fig. 5. The minimum-energy reaction path corresponds to an imaginary relaxation path of steepest descent, along which the molecular system glides with almost infinitely slow motion from a saddle point to an adjacent minimum.

Lagrange Multipliers

Lagrange multipliers can be used to find the stationary points of functions, subject to a set of constraints. Suppose we wish to find the stationary points of a function $f(x, y) = 4x^2 + 3x + 2y^2 + 6y$ subject to the constraint $y = 4x + 2$. In the Lagrange method the constraint is written in the form $g(x, y) = 0$:

$$g(x, y) = y - 4x - 2 = 0 \tag{1.40}$$

To find stationary points $f(x, y)$ subject to $g(x, y) = 0$ we first determine the total derivative df, which is set equal to zero:

$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy = (8x + 3)\,dx + (4y + 6)\,dy = 0 \tag{1.41}$$

Without the constraint the stationary points would be determined by setting the two partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ equal to zero, as x and y are independent. With the constraint, x and y are no longer independent but are related via the derivative of the constraint function g:

$$dg = \frac{\partial g}{\partial x}dx + \frac{\partial g}{\partial y}dy = -4dx + dy = 0 \tag{1.42}$$

The derivative of the constraint function, dg, is multiplied by a parameter λ (the *Lagrange multiplier*) and added to the total derivative df:

$$\left(\frac{\partial f}{\partial x} + \lambda\frac{\partial g}{\partial x}\right)dx + \left(\frac{\partial f}{\partial y} + \lambda\frac{\partial g}{\partial y}\right)dy = 0 \tag{1.43}$$

The value of the Lagrange multiplier is obtained by setting each of the terms in parentheses to zero. Thus for our example we have:

$$8x + 3 - 4\lambda = 0 \tag{1.44}$$

$$4y + 6 + \lambda = 0 \tag{1.45}$$

From these two equations we can obtain a further equation linking x and y:

$$\lambda = 2x + 3/4 = -6 - 4y \quad \text{or} \quad x = -27/8 - 2y \tag{1.46}$$

Combining this with the constraint equation enables us to identify the stationary point, which is at (−59/72, −23/18).

This simple example could, of course, have been solved by simply substituting the constraint equation into the original function, to give a function of just one of the variables. However, in many cases this is not possible. The Lagrange multiplier method provides a powerful approach which is widely applicable to problems involving constraints such as in constraint dynamics (Section 7.5) and in quantum mechanics.

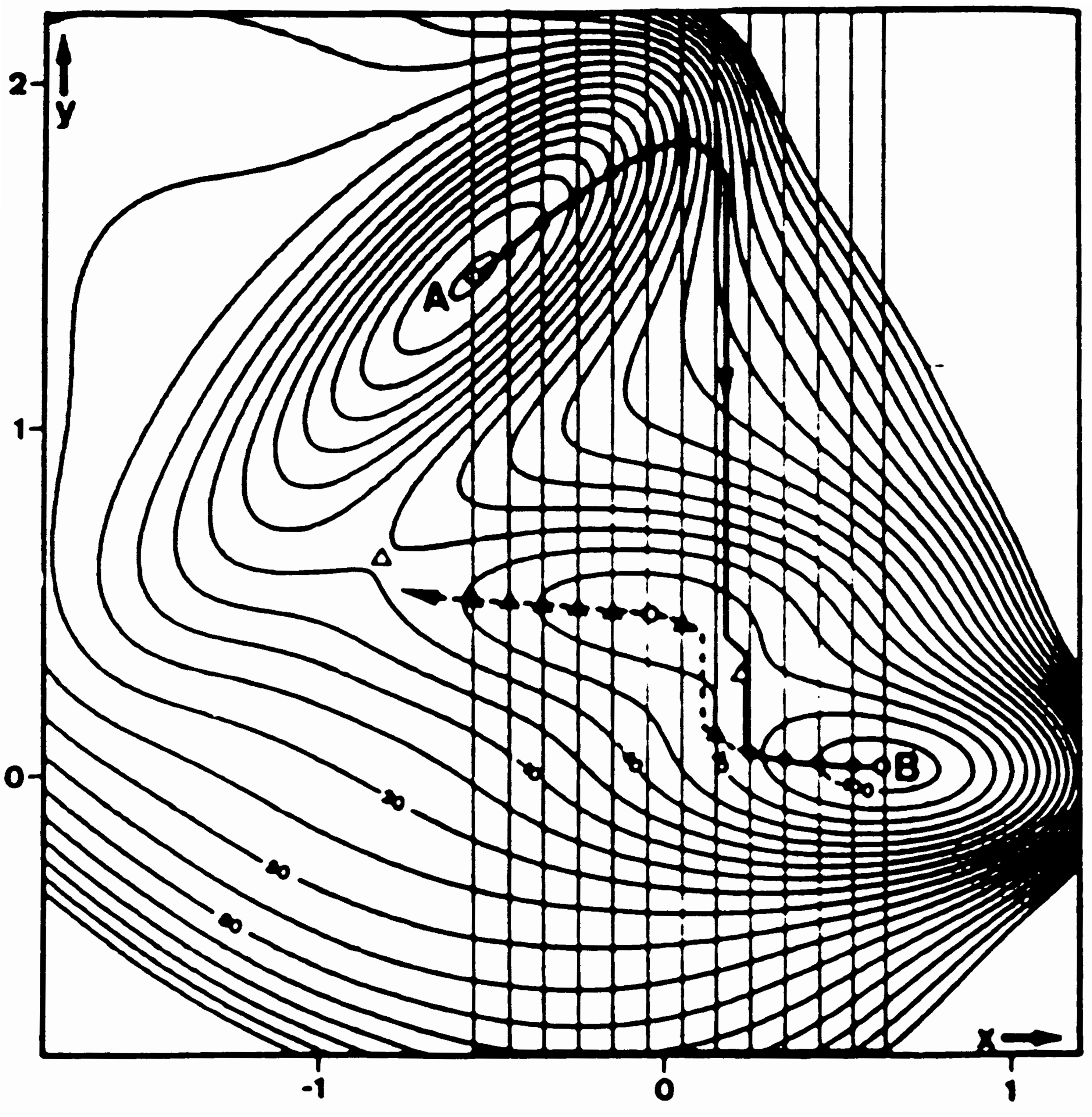


Fig. 6. Application of the reaction coordinate method to the two-parametric model potential.

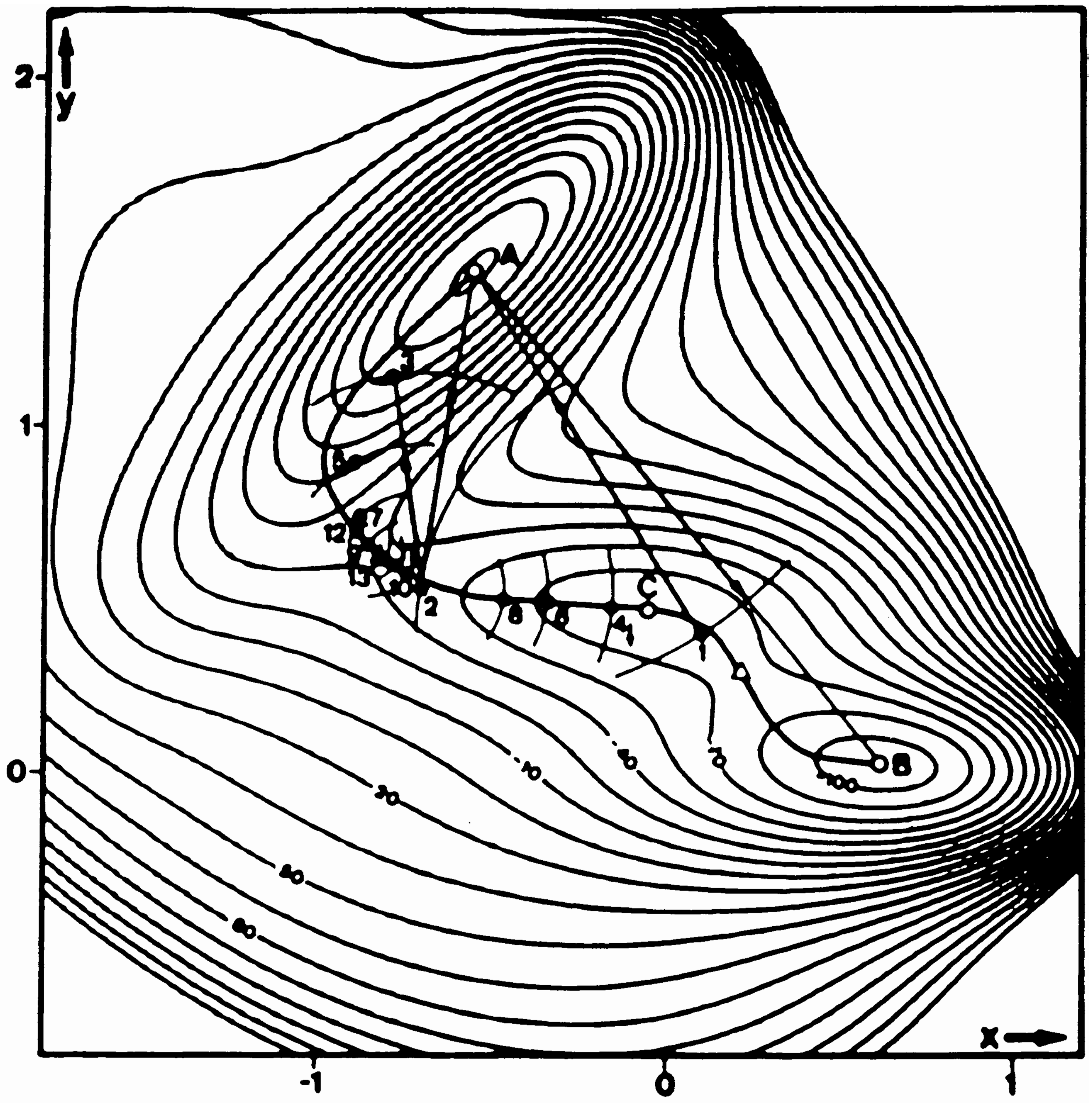


Fig. 14. Localization of a saddle point on the two-parametric model potential surface by means of ascending valley points.

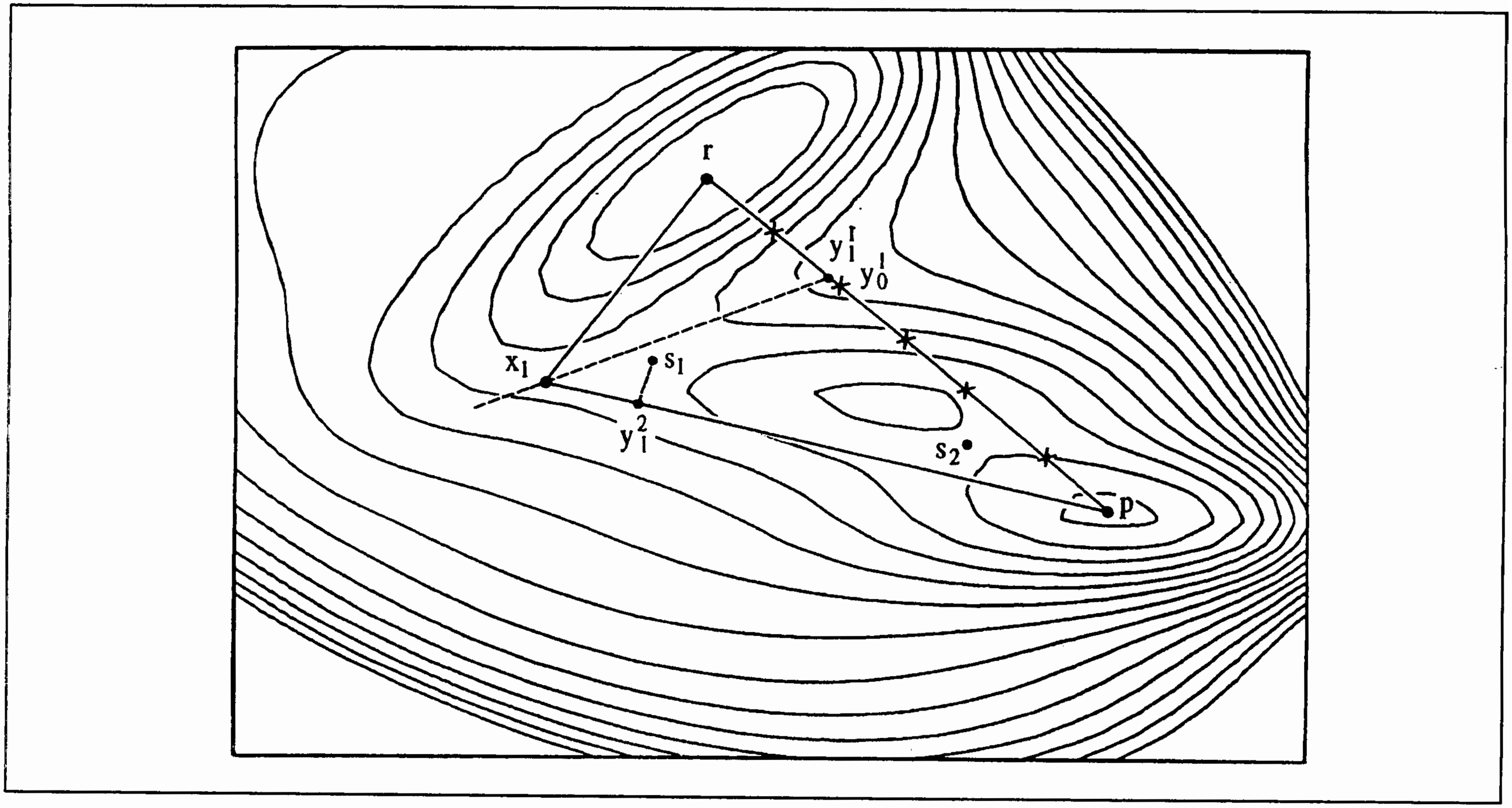


Fig. 5.30: The conjugate peak refinement method. r and p are the initial minima (the reactants and the products). A coarse step search along the line connecting r and p suggests that there is a maximum near the point y_0^1. Minimisation along the line connecting r to p gives the point y_1^1. A line minimisation is then performed along the conjugate vector to give the point x_1. In the second iteration the procedure is repeated for the lines $r - x_1$ and $x_1 - p$. A maximum is found at y_1^2, which after minimisation along the conjugate vector gives the saddle point s_1. Subsequent iterations of the algorithm enable the second saddle point s_2 to be identified. (Figure adapted in part from Fischer S and M Karplus 1992. Conjugate Peak Refinement: An Algorithm for Finding Reaction Paths and Accurate Transition States in Systems with Many Degrees of Freedom. Chemical Physical Letters **194**:252–261.)

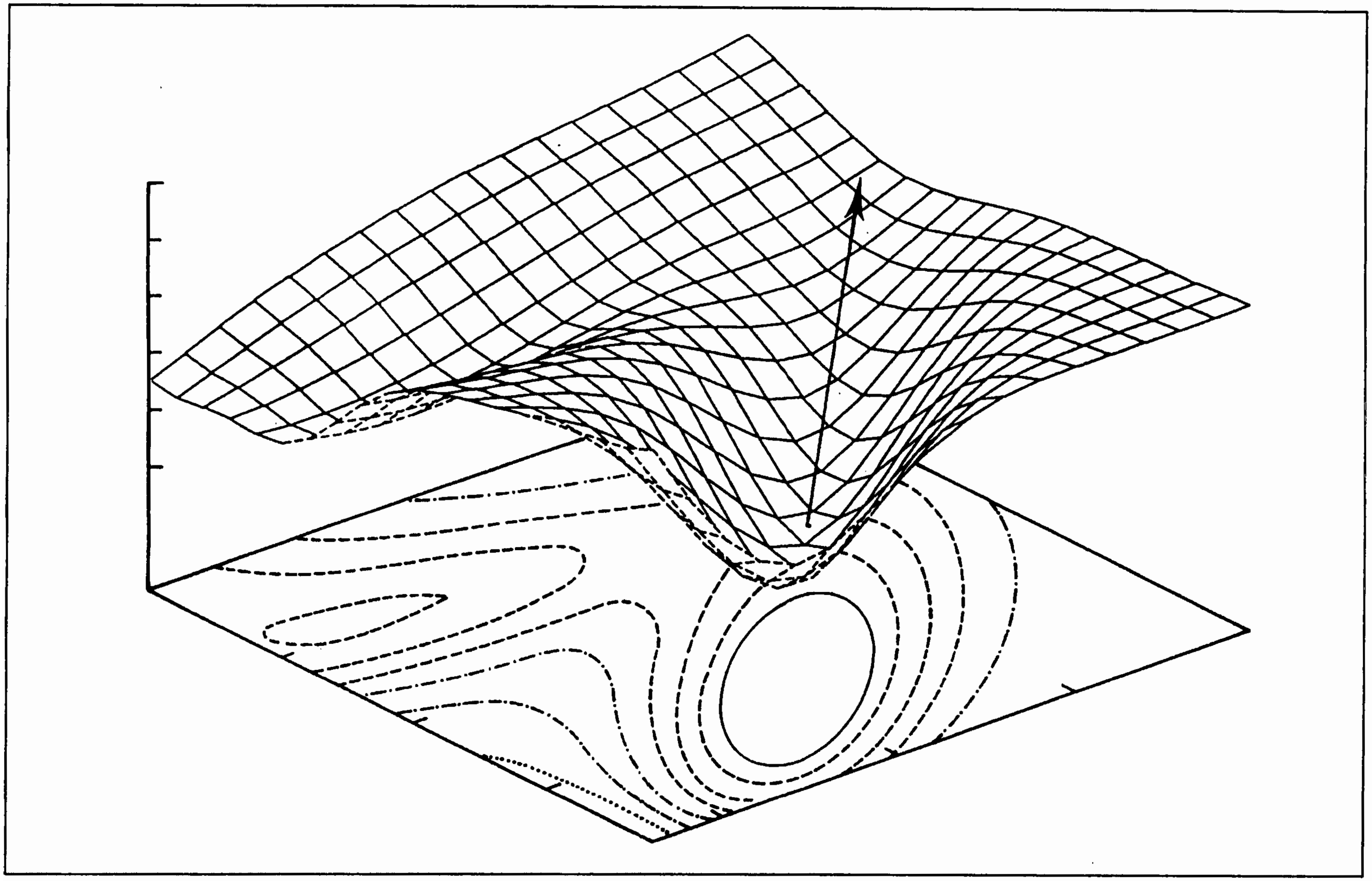


Fig. 5.27: Too large a step size may lead to the wrong saddle point or an inefficient algorithm.

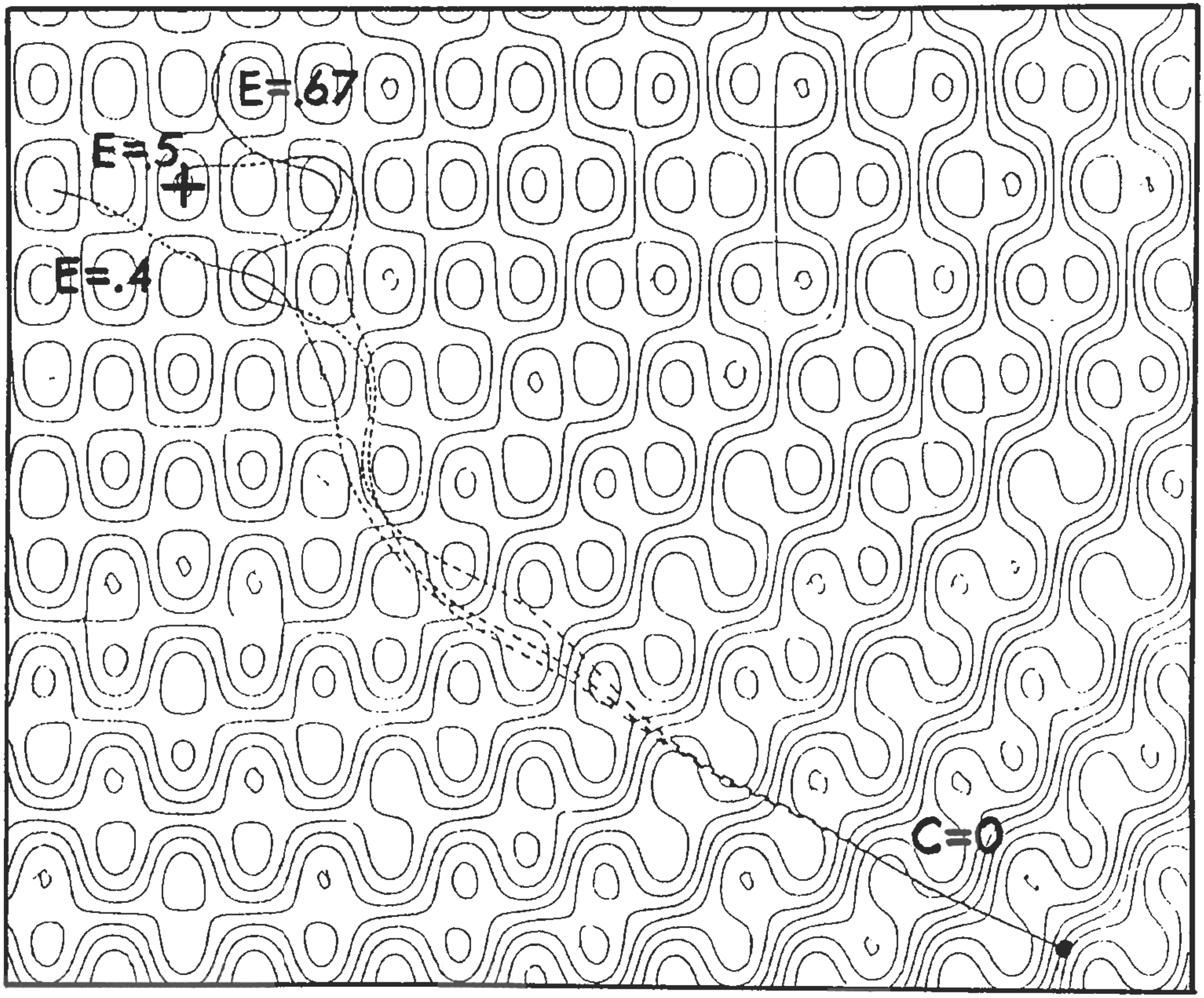
Global Optimization Techniques

- **Deterministic Methods**
  - **Space covering (grid search)**
  - **Systematic search**
  - **Trajectory & generalized descent**
  - **Smoothing / deformation**

- **Stochastic Methods**
  - **Multi–start from many trial points**
  - **Bayesian statistical models**
  - **Clustering techniques**
  - **Monte Carlo & Simulated Annealing**

- **Heuristic Methods from Chemistry**
  - **Distance geometry**
  - **Scheraga's build–up procedure**
  - **Saunders' random kick**
  - **Still's torsional tree–search**

Generalized Descent Global Optimization —



Diffusion Equation Method

For a function *f(x)*, **define** $f^{[N]}(x)$ as the sum of ***N*** even-power derivatives

$$f^{[N]}(x) = \left(1 + \beta \frac{d^2}{dx^2}\right)^N f(x)$$

Let

$$\beta = \frac{t}{N}$$

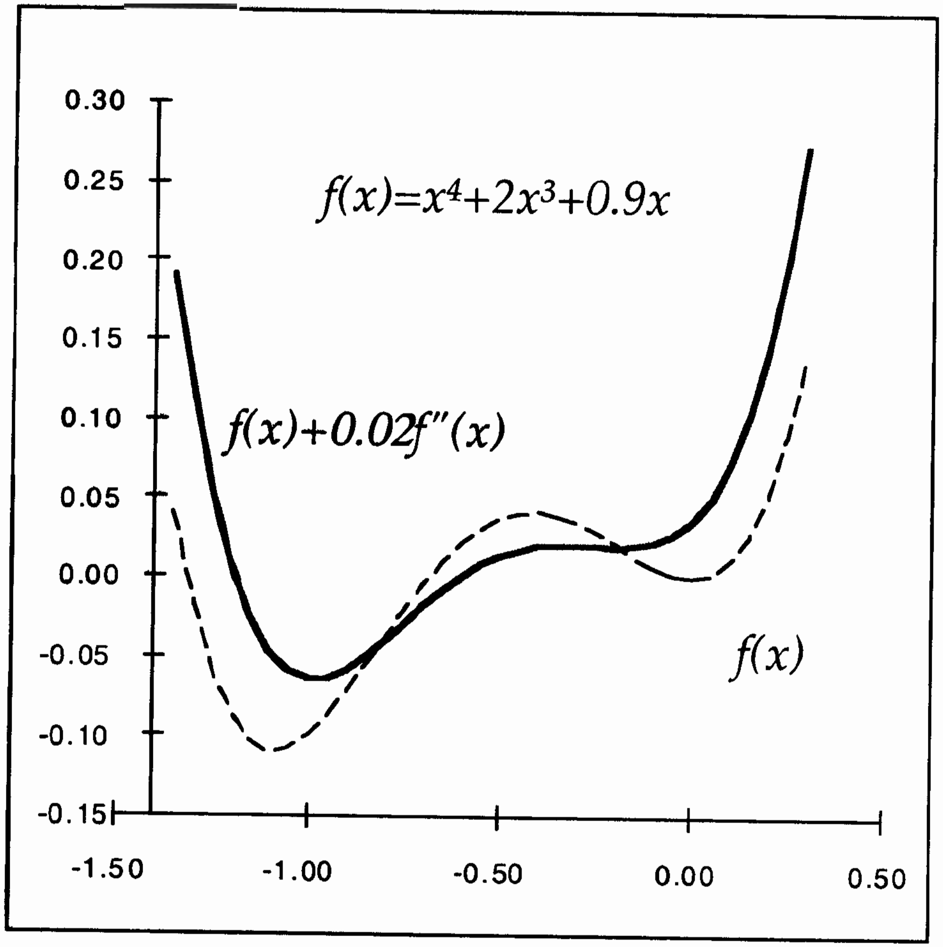
and evaluate the limit of the infinite expansion

$$F(x,t) = \lim_{N\to\infty}\left(1 + \frac{t}{N}\frac{d^2}{dx^2}\right)^N f(x)$$

$$= \exp\left(t\frac{d^2}{dx^2}\right) f(x)$$

where ***F(x,t)*** satisfies the diffusion equation:

$$\frac{\partial^2 F}{\partial x^2} = \frac{\partial F}{\partial t}$$


f(x)=x⁴+2x³+0.9x
f(x)+0.02f''(x)
f(x)