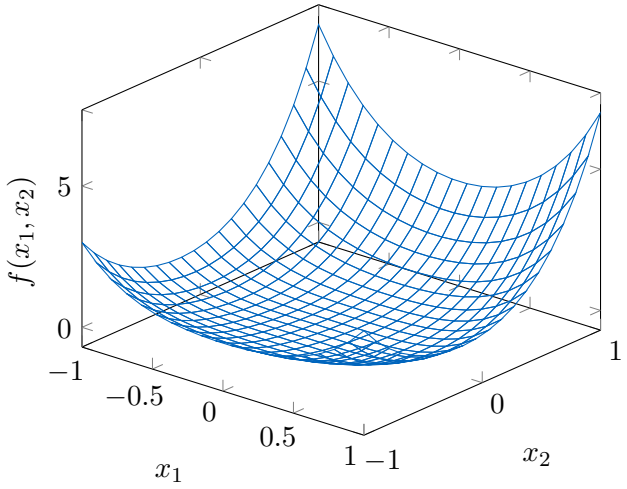
Q 01. Implement gradient and conjugate-gradient methods[1] for unconstrained minimisation in a coding language of your choice. Use the code to minimise the $f(\mathbf{x})$ below from $\mathbf{x}^{(0)} = (0.25, 0.25)$

$$f(\mathbf{x}) = x_2^2 + x_2||\mathbf{x}||^2 + ||\mathbf{x}||^4, \quad \text{with } \mathbf{x} \in \mathcal{R}^2$$



Q 02. The Stochastic Neighbour Embedding (SNE, G.E. Hinton and S.T. Roweis. *Stochastic Neighbor Embedding.* In NIPS, 15, 833–840, 2002) is a classic machine learning algorithm for non-linear dimensionality reduction. Find and read the original paper presenting the SNE, implement its plain vanilla algorithm (the one in Section 2[2]) in a programming language of your choice. To optimise the SNE, use your code for gradient and conjugate-gradient minimisation (Q 01, above).

Evaluate your implementations of the SNE using the beloved `Glass identification` set of data[3].

[1] See next page for an overview.

[2] Do not bother about writing code to optimise the perplexity parameter σ_i of the SNE. Rather, set $\alpha_i = \alpha$ to be equal to some integer (about 5% of the number of observations, like).

[3] Experiment with different initial solutions.

Descent directions (overview) For some $f : \mathcal{R}^n \to \mathcal{R}$ called an *objective function*, the problem

$$\text{minimise } f(\mathbf{x}) \text{ in } \mathcal{R}^n \tag{1}$$

is called an *unconstrained optimisation problem*. The point $\hat{\mathbf{x}}$, the solution of problem (1), is called a *global minimiser* of f, whereas point $\hat{\mathbf{x}}$ is called a *local minimiser* of f if, for some $R > 0$, we have

$$f(\hat{\mathbf{x}}) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in B(\hat{\mathbf{x}}, R),$$

with $B(\cdot, \cdot)$ a ball of radius R centred in $\hat{\mathbf{x}}$.

Let f be continuous and continuously differentiable in $\mathcal{R}^n$, $f \in C^1(\mathcal{R}^n)$. Let $\nabla f(\mathbf{x})$ be the *gradient* of f at point $\mathbf{x}$,

$$\nabla f(\mathbf{x}) = \Big[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n}\Big]^T.$$

If $f \in C^2(\mathcal{R}^n)$, let $\nabla^2 f(\mathbf{x})$ or $\mathbf{H}(\mathbf{x})$ be the *hessian* of f at point $\mathbf{x}$, with

$$\mathbf{H}_{i,j}(\mathbf{x}) = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}, \quad i, j = 1, 2, \ldots, n.$$

If $\hat{\mathbf{x}} \in \mathcal{R}^n$ is a local minimiser of f under the condition that $f \in C^1(B(\hat{\mathbf{x}}, R))$ for a suitable $R > 0$, then $\nabla f(\mathbf{x}) = \mathbf{0}$. Moreover, if $f \in C^2(B(\hat{\mathbf{x}}, R))$, the $\mathbf{H}(\hat{\mathbf{x}})$ is positive semidefinite. If $\hat{\mathbf{x}} \in B(\hat{\mathbf{x}}, R)$ and $\mathbf{H}(\hat{\mathbf{x}})$ is positive definite, then $\hat{\mathbf{x}}$ is a local minimiser of f in $B(\hat{\mathbf{x}}, R)$.

A point $\hat{\mathbf{x}}$ such that $\nabla f(\hat{\mathbf{x}}) = \mathbf{0}$ is called a *stationary point*. This condition is necessary for optimality to hold. The condition is also sufficient if f is convex in $\mathcal{R}^n$: That is, if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{R}^n$ and for any $\alpha \in [0, 1]$, $f[\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}] \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y})$.

Descent methods are iterative procedures for solving problem (1). They are can be formulated as

- '*Given an initial solution* $\mathbf{x}^{(0)} \in \mathcal{R}^n$*, compute for* $k = 0, 1, 2, \ldots$ *the quantity*

  $$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

  until convergence, where $\mathbf{d}^{(k)}$ *is a suitably selected direction and* α_k *is a positive parameter called stepsize that indicates the step along the direction* $\mathbf{d}^{(k)}$*,*'

As we are minimising f, $\mathbf{d}^{(k)}$ needs be a *descent direction*. This is guaranteed by the conditions

$$\begin{aligned} \mathbf{d}^{(k)^T} \nabla f(\mathbf{x}^{(k)}) < 0, &\quad \text{if } \nabla f(\mathbf{x}^{(k)}) \neq 0; \\ \mathbf{d}^{(k)} = 0, &\quad \text{if } \nabla f(\mathbf{x}^{(k)}) = 0. \end{aligned}$$

- The *gradient method* (steepest descent) sets $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}$.
- The *conjugate-gradient method* sets $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) + \beta_k \mathbf{d}^{(k-1)}$, with $\mathbf{d}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$ and β_k a suitable scalar which we could set to be $\beta_k^{\text{FR}} = \dfrac{||\nabla f(\mathbf{x}^{(k)})||^2}{||\nabla f(\mathbf{x}^{(k-1)})||^2}$ (Fletcher-Reeves).

The strategies to find good stepsizes α_k are tedious. A constant value $\alpha \in (0, 1)$ is often acceptable.