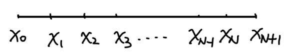
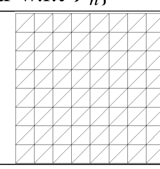
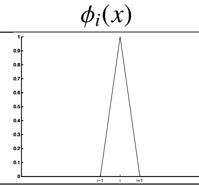
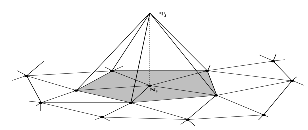
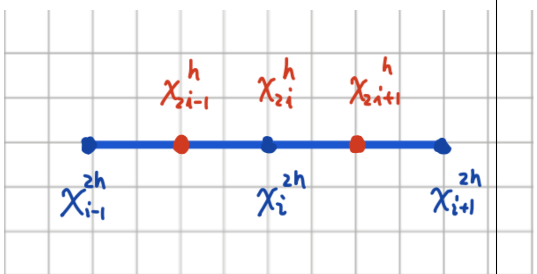
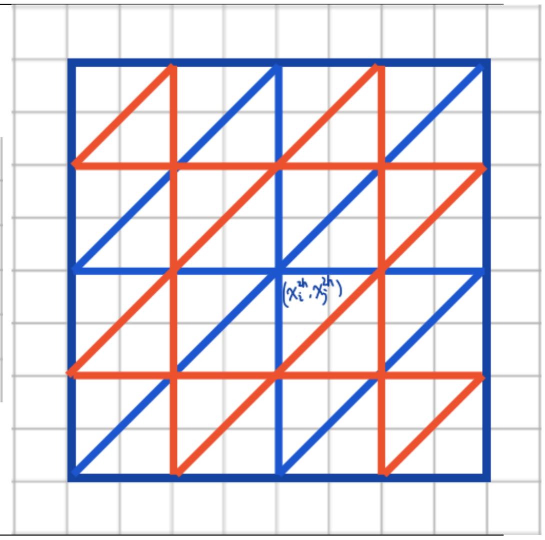
8.2 1D and 2D Finite Element and Multigrid

1D and 2D Comparison for Finite Element and Multigrid

<table>
<tr><td>$1D : \Omega = (0,1)$</td><td>$2D : \Omega = (0,1) \times (0,1)$</td></tr>
<tr><td>$\begin{cases} -u'' = f & x \in \Omega \\ u(0) = u(1) = 0 \end{cases}$</td><td>$\begin{cases} -\triangle u = f & x \in \Omega \\ u = 0 & x \in \partial\Omega \end{cases}$</td></tr>
<tr><td colspan="2">$u = \arg\min_{v \in V} J(v)$</td></tr>
<tr><td>$J(v) = \frac{1}{2}\int_0^1 |v'|^2 dx - \int_0^1 f v dx$</td><td>$J(v) = \frac{1}{2}\int_\Omega |\nabla v|^2 dx - \int_\Omega f v dx$</td></tr>
<tr><td colspan="2">$V = \{v : \Omega \to \mathbb{R}^1$ is continuous and piecewise smooth, $v|_{\partial\Omega} = 0\}$</td></tr>
<tr><td colspan="2">FE space: $V_h = \{v_h \in V, v_h$ is piecewise linear w.r.t $\mathcal{T}_h\}$</td></tr>
<tr><td>
</td><td></td></tr>
<tr><td>$\phi_i(x)$</td><td>$\phi_{ij}(x,y)$</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Find $u_h \in V_h$ s.t. $J(u_h) = \min_{v_h \in V_h} J(v_h)$</td></tr>
<tr><td>$u_h = \sum_{i=1}^n \mu_i \phi_i(x)$</td><td>$u_h = \sum_{i,j=1}^n \mu_{ij}\phi_{ij}(x,y)$</td></tr>
<tr><td>Find $\mu \in \mathbb{R}^n$ s.t. $I(\mu) = \min_{\nu \in \mathbb{R}^n} I(\nu)$</td><td>Find $\mu \in \mathbb{R}^{n\times n}$ s.t. $I(\mu) = \min_{\nu \in \mathbb{R}^{n\times n}} I(\nu)$</td></tr>
<tr><td colspan="2">$I(\nu) = \frac{1}{2}(A * \nu, \nu)_{l^2} - (b, \nu)_{l^2}$</td></tr>
<tr><td>$A = \frac{1}{h}(-1, 2, -1)$</td><td>$A = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$</td></tr>
<tr><td colspan="2">$\mu = \arg\min I(\nu) \Longleftrightarrow \nabla J(\mu) = A * \mu - b = 0$</td></tr>
<tr><td colspan="2">GD Method: $\mu^{m+1} = \mu^m - \eta(A * \mu^m - b)$</td></tr>
<tr><td>$\eta = \frac{h}{4}$</td><td>$\eta = \frac{1}{8}$</td></tr>
</table>

Basic multigrid components

	
$\phi_i^{2h} = \frac{1}{2}\phi^h_{2i-1} + \phi^h_{2i} + \frac{1}{2}\phi^h_{2i+1}$	$\phi^{2h}_{i,j} = \phi^h_{2i,2j} + \frac{1}{2}\left(\phi^h_{2i-1,2j-1} + \phi^h_{2i+1,2j+1}\right) + \frac{1}{2}\left(\phi^h_{2i-1,2j} + \phi^h_{2i,2j-1} + \phi^h_{2i+1,2j} + \phi^{2h}_{2i,2j+1}\right)$
$\Phi^{2h} = R *_2 \Phi^h$	
$R = (\frac{1}{2}, 1, \frac{1}{2})$	$R = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$

8.2.1 Multigrid algorithm for $A * \mu = f$

Algorithm 11 A multigrid algorithm $\mu = \mathrm{MG1}(f; \mu^0; J, \nu_1, \cdots, \nu_J)$

Set up

$$f^1 = f, \quad \mu^1 = \mu^0.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

for $i = 1 : \nu_\ell$ **do**

$$\mu^\ell \leftarrow \mu^\ell + S^\ell * (f^\ell - A_\ell * \mu^\ell). \tag{8.33}$$

end for

Form restricted residual and set initial guess:

$$\mu^{\ell+1} \leftarrow \Pi_\ell^{\ell+1} \mu^\ell, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * \mu^\ell) + A_{\ell+1} * \mu^{\ell+1},$$

end for

Prolongation and restriction from coarse to fine level

for $\ell = J - 1 : 1$ **do**

$$\mu^\ell \leftarrow \mu^\ell + R *_2^\top (\mu^{\ell+1} - \Pi_\ell^{\ell+1} \mu^\ell).$$

end for

$$\mu \leftarrow \mu^1.$$

Remark 9. The above multigrid method for the linear problem $A * \mu = b$ is independent of the choice of the interpolation operation $\Pi_\ell^{\ell+1} : \mathbb{R}^{n_\ell \times n_\ell} \mapsto \mathbb{R}^{n_{\ell+1} \times n_{\ell+1}}$ and in particular, we could take $\Pi_\ell^{\ell+1} := 0$. But such an operation is critical for nonlinear problems.

8.2.2 MgNet

Algorithm 12 μ^J = MgNet1$(f;\mu^0;J,\nu_1,\cdots,\nu_J)$

Set up

$$f^1 = \theta * f, \quad \mu^1 = \mu^0.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

for $i = 1 : \nu_\ell$ **do**

$$\mu^\ell \leftarrow \mu^\ell + \sigma \circ S^\ell * \sigma \circ (f^\ell - A_\ell * \mu^\ell). \tag{8.34}$$

end for

Form restricted residual and set initial guess:

$$\mu^{\ell+1} \leftarrow \Pi_\ell^{\ell+1}\mu^\ell, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * \mu^\ell) + A_{\ell+1} * \mu^{\ell+1},$$

end for

8.3 FEM and 2D in Convolution

Let us first briefly describe finite difference methods and finite element methods for the numerical solution of the following boundary value problem

$$-\Delta u = f, \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega, \quad \Omega = (0,1)^2. \tag{8.35}$$

For the x direction and the y direction, we consider the partition:

$$0 = x_0 < x_1 < \cdots < x_{n+1} = 1, \quad x_i = \frac{j}{n+1}, \quad (i = 0, \cdots, n+1); \tag{8.36}$$

$$0 = y_0 < y_1 < \cdots < y_{n+1} = 1, \quad y_j = \frac{j}{n+1}, \quad (j = 0, \cdots, n+1). \tag{8.37}$$

Such a uniform partition in the x and y directions leads us to a special example in two dimensions, a uniform square mesh $\mathbb{R}^2_h = \{(ih, jh); i, j \in \mathbb{Z}\}$ (Figure 8.3). Let $\Omega_h = \Omega \cap \mathbb{R}^2_h$, the set of interior mesh points and $\partial\Omega_h = \partial\Omega \cap \mathbb{R}^2_h$, the set of boundary mesh points.

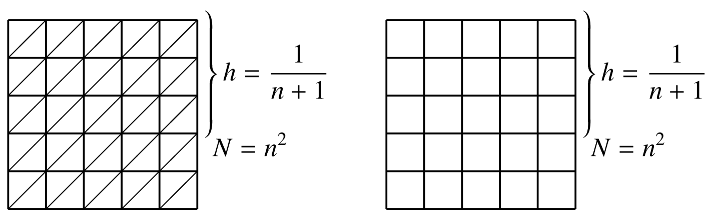


Fig. 8.9. Two-dimensional uniform grid for finite element and finite difference

8.4 Finite element methods

We consider two finite elements: continuous linear element and bilinear element. These two finite element methods find $u_h \in V_h$ such that

$$(\nabla u_h, \nabla v_h) = (f, v_h), \ \forall v_h \in V_h. \tag{8.38}$$

Basis functions ϕ_{ij} satisfy

$$\phi_{ij}(x_k, y_l) = \delta_{(i,j),(k,l)}. \tag{8.39}$$

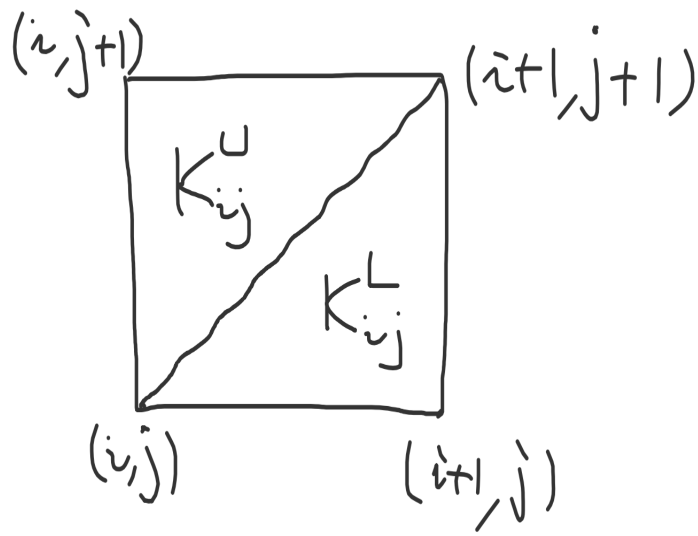


Fig. 8.10. The picture of $E_{i,j}$

8.4.1 Linear finite element

Continuous linear finite element discretization of (8.35) on the left triangulation in Fig 8.3. The discrete space for linear finite element is

$$\mathcal{V}_h = \{v_h : v_h|_K \in P_1(K) \text{ and } v_h \text{ is globally continuous}\}.$$

Denote $E_{i,j} = [x_i, x_{i+1}] \times [y_i, y_{i+1}] = K^U_{i,j} \cup K^L_{i,j}$. For linear element case,

(8.40)

$$\begin{aligned}
(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) &= \sum_{i,j=1}^{n} \int_{E_{i,j}} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h dxdy = \sum_{i,j=1}^{n} \int_{K^U_{i,j}} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h dxdy + \sum_{i,j=1}^{n} \int_{K^L_{i,j}} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h dxdy \\
&= \sum_{i,j=1}^{n} \int_{K^U_{i,j}} \left(\frac{u_{i,j+1} - u_{i+1,j+1}}{h} \frac{v_{i,j+1} - v_{i+1,j+1}}{h} + \frac{u_{i,j+1} - u_{i,j}}{h} \frac{v_{i,j+1} - v_{i,j}}{h}\right) dxdy \\
&+ \sum_{i,j=1}^{n} \int_{K^L_{i,j}} \left(\frac{u_{i+1,j} - u_{i,j}}{h} \frac{v_{i+1,j} - v_{i,j}}{h} + \frac{u_{i+1,j} - u_{i+1,j+1}}{h} \frac{v_{i+1,j} - v_{i+1,j+1}}{h}\right) dxdy \\
&= \sum_{i,j=1}^{n} \Big[(u_{i+1,j} - u_{i,j})(v_{i+1,j} - v_{i,j}) + (u_{i,j+1} - u_{i,j})(v_{i,j+1} - v_{i,j})\Big] \\
&= (A * u, v)_{l^2}
\end{aligned}$$

where $A = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$ and $A * u$ is given by (8.41).

It is easy to verify that the formulation for the linear element method is

$$4u_{i,j} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = f_{i,j}, \quad u_{i,j} = 0 \text{ if } i \text{ or } j \in \{0, n+1\}, \tag{8.41}$$

where

$$f_{i,j} = \int_\Omega f(x,y)\phi_{i,j}(x,y)\mathrm{d}x\mathrm{d}y \approx h^2 f(x_i, y_j). \tag{8.42}$$

Proposition 1. *The mapping $A*$ has following properties*

1. *A is symmetric, namely*

$$(A * u, v)_{l^2} = (u, A * v)_{l^2}.$$

2. $(A * v, v)_F > 0, \quad$ *if* $\quad v \neq 0$.
3. $A * u = f$ *if and only if*

$$u \in \arg\min_{v \in \mathcal{V}_h} J(v) = \frac{1}{2}(A * v, v) - (f, v). \tag{8.43}$$

4. *The eigenvalues λ_{kl} and eigenvectors u^{kl} of A are given by*

$$\lambda_{kl} = 4(\sin^2 \frac{k\pi}{2(n+1)} + \sin^2 \frac{l\pi}{2(n+1)}),$$

$$u_{ij}^{kl} = \sin \frac{ki\pi}{n+1} \sin \frac{lj\pi}{n+1}, \ 1 \le i \le n, \ 1 \le j \le n,$$

and $\rho(A) < 8$. Furthermore,

$$\lambda_{n,n} = 8\cos^2 \frac{\pi}{2(n+1)} \approx 8(1 - (\frac{\pi}{2(n+1)})^2) \approx 8 - \frac{2\pi^2}{(n+1)^2}$$

8.4.2 Bilinear element

Continuous bilinear finite element discretization of (8.35) on the right mesh in Fig. 8.3. The discrete space for linear finite element is

$$\mathcal{V}_h = \{v_h : v_h|_K \in \{1, \ x, \ y, \ xy\} \text{ and } v_h \text{ is globally continuous}\}.$$

For bilinear element case, we have

(8.44)

$$
\begin{aligned}
(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) &= \sum_{i,j=1}^{n} \int_{E_{i,j}} \nabla \mathbf{u}_h, \nabla \mathbf{v}_h dxdy \\
&= \sum_{i,j=1}^{n} \int_{E_{i,j}} \left(\frac{(u_{i+1,j} - u_{i,j})(y_{j+1} - y)}{h^2} + \frac{(u_{i,j+1} - u_{i+1,j+1})(y - y_j)}{h^2} \right) \\
&\qquad \left(\frac{(v_{i+1,j} - v_{i,j})(y_{j+1} - y)}{h^2} + \frac{(v_{i,j+1} - v_{i+1,j+1})(y - y_j)}{h^2} \right) \\
&\qquad + \left(\frac{(u_{i,j+1} - u_{i,j})(x_{i+1} - x)}{h^2} + \frac{(u_{i+1,j} - u_{i+1,j+1})(x - x_i)}{h^2} \right) \\
&\qquad \left(\frac{(v_{i,j+1} - v_{i,j})(x_{i+1} - x)}{h^2} + \frac{(v_{i+1,j} - v_{i+1,j+1})(x - x_i)}{h^2} \right) dxdy \\
&= (A * u, v)_{l^2}.
\end{aligned}
$$

where $A = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$ and $A * u$ is given by (8.45).

And we have

$$8u_{ij} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} + u_{i+1,j+1} + u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1}) = f_{i,j}, \tag{8.45}$$

and $u_{i,j} = 0$ if i or $j \in \{0, n+1\}$.

8.5 Piecewise (bi-)linear functions on multilevel grids

An image can be viewed as a function on a grid. Images with different resolutions can then be viewed as functions on grids of different sizes. The use of such multiple-grids is a main technique used in the standard multigrid method for solving discretized partial differential equations, and it can also be interpreted as a main ingredient used in convolutional neural networks (CNN) for image calssification.

An image can be viewed as a function on a grid [9] on a rectangle domain $\Omega \in \mathcal{R}^2$. Without loss of generality, we assume that the grid, $\mathcal{T}$, is of size

$$m = 2^s + 1 \quad n = 2^t + 1$$

for some integers $s, t \geq 1$. Starting from $\mathcal{T}_1 = \mathcal{T}$, we consider a sequence of coarse grids with $J = \min(s, t)$ (as depicted in Fig. 8.5 with $J = 4$):

$$\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_J \tag{8.46}$$

such that $\mathcal{T}_\ell$ consist of $m_\ell \times n_\ell$ grid points, with

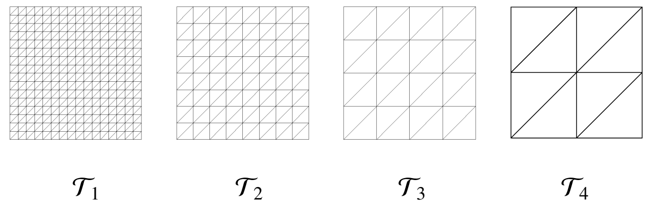


Fig. 8.11. multilevel grids for piecewise linear functions

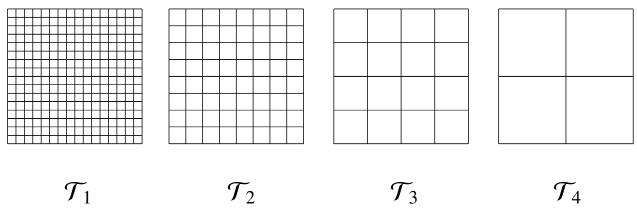


Fig. 8.12. multilevel grids for piecewise bilinear functions

$$m_\ell = 2^{s-\ell+1} + 1, \quad n_\ell = 2^{t-\ell+1} + 1. \tag{8.47}$$

The grid points of these grids can be given by

$$x_i^\ell = ih_\ell, y_j^\ell = jh_\ell, i = 0, \dots, m_\ell - 1, j = 0, \dots, n_\ell - 1.$$

Here $h_\ell = 2^{-s+\ell-1}a$ for some $a > 0$. The above geometric coordinates (x_i^ℓ, y_j^ℓ) are usually not used in image precess literatures, but they are relevant in the context of multigrid method for numerical solution of PDEs. We now consider piecewise bilinear (or linear) functions on the sequence of grids (10.1) and we obtain a nested sequence of linear vector spaces

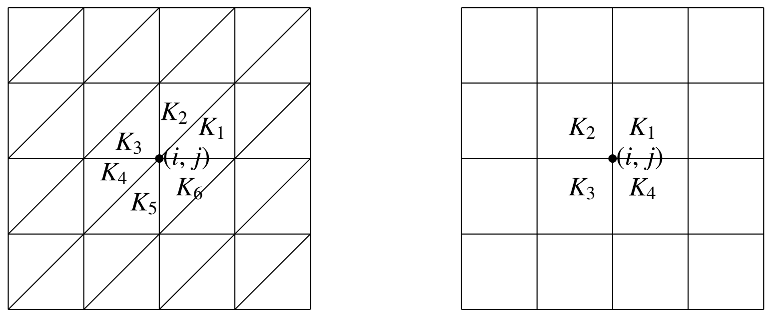
$$\mathcal{V}_1 \supset \mathcal{V}_2 \supset \dots \supset \mathcal{V}_J. \tag{8.48}$$

8.5.1 Nodal bases and dual bases on multilevel spaces

Here each $\mathcal{V}_\ell$ consists of all piecewise linear (or bilinear) functions with respect to the grid (10.1) and (10.2). Each $\mathcal{V}_\ell$ has a set of basis functions: $\phi_{i,j}^\ell \in \mathcal{V}_\ell$ satisfying:

$$\phi_{i,j}^\ell(x_p^\ell, y_q^\ell) = \delta_{(i,j),(p,q)} = \begin{cases} 1 & \text{if} \quad (p,q) = (i,j), \\ 0 & \text{if} \quad (p,q) \neq (i,j). \end{cases} \tag{8.49}$$

For the piecewise linear finite element space, the nodal basis function $\phi_{i,j}^\ell$ associate with each (x_i^ℓ, y_j^ℓ) (satisfying (8.49)) is given by



$$
\phi_{i,j}^{\ell}(x,y) = \begin{cases} \frac{x_{i+1}^{\ell}-x}{h}, & (x,y)\in K_1,\\ \frac{y_{j+1}^{\ell}-y}{h}, & (x,y)\in K_2,\\ \frac{x-x_{i-1}^{\ell}-(y-y_j^{\ell})}{h}, & (x,y)\in K_3,\\ \frac{x-x_{i-1}^{\ell}}{h}, & (x,y)\in K_4,\\ \frac{y-y_{j-1}^{\ell}}{h}, & (x,y)\in K_5,\\ \frac{x_{i+1}^{\ell}-x+y-y_j^{\ell}}{h}, & (x,y)\in K_6\\ 0, & \text{elsewhere.} \end{cases} \tag{8.50}
$$

shown in Fig. 8.13.

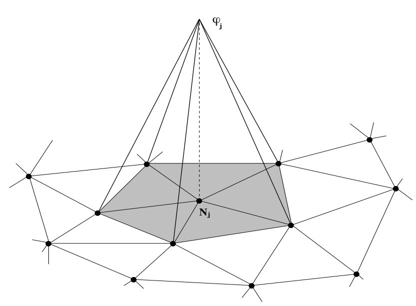


Fig. 8.13. Nodal basis for linear element.

For bilinear element, it is easy to see that the nodal basis function $\phi^\ell_{i,j}$ associated with each (x^ℓ_i, y^ℓ_j) (satisfying (8.49)) is given by

$$\phi^\ell_{i,j}(x,y) = \begin{cases} \frac{(x^\ell_{i+1}-x)(y^\ell_{j+1}-y)}{h^2}, & (x,y) \in K_1, \\ \frac{(x-x^\ell_{i-1})(y^\ell_{j+1}-y)}{h^2}, & (x,y) \in K_2, \\ \frac{(x-x^\ell_{i-1})(y-y^\ell_{j-1})}{h^2}, & (x,y) \in K_3, \\ \frac{(x^\ell_{i+1}-x)(y-y^\ell_{j-1})}{h^2}, & (x,y) \in K_4, \\ 0 & \text{elsewhere}. \end{cases} \tag{8.51}$$

Associated with the above nodal basis functions $\phi^\ell_{i,j}(x,y) \subset \mathcal{V}_\ell$, we define the corresponding dual basis functions $\psi^\ell_{i,j}(x,y) \subset \mathcal{V}_\ell$ satisfying

$$(\psi^\ell_{i,j}(x,y), \phi^\ell_{p,q}(x,y))_{L^2(\Omega)} = \delta_{(i,j),(p,q)}. \tag{8.52}$$

The existence of dual basis functions is obvious, but the exact expression of the dual basis functions are in general difficult to obtain. In fact, (8.52) is the only property that is needed in the application of dual basis.

We write $\mathbf{u}_h(x,y) = \sum\limits_{i,j=1}^{n} u_{i,j}\phi_{i,j}(x,y), \mathbf{v}_h(x,y) = \sum\limits_{i,j=1}^{n} v_{i,j}\phi_{i,j}(x,y)$.

Lemma 23. *For bilinear functions, we have*

$$\begin{aligned} \phi^{\ell+1}_{i,j}(x,y) = {} & \phi^\ell_{2i,2j}(x,y) + \frac{1}{2}\left(\phi^\ell_{2i-1,2j}(x,y) + \phi^\ell_{2i,2j-1}(x,y) + \phi^\ell_{2i+1,2j}(x,y) + \phi^\ell_{2i,2j+1}(x,y)\right) \\ & + \frac{1}{4}\left(\phi^\ell_{2i-1,2j-1}(x,y) + \phi^\ell_{2i+1,2j-1}(x,y) + \phi^\ell_{2i+1,2j+1}(x,y) + \phi^\ell_{2i-1,2j+1}(x,y)\right). \end{aligned} \tag{8.53}$$

For linear functions, we have

$$\begin{aligned} \phi^{\ell+1}_{i,j}(x,y) = {} & \phi^\ell_{2i,2j}(x,y) + \frac{1}{2}\left(\phi^\ell_{2i-1,2j-1}(x,y) + \phi^\ell_{2i+1,2j+1}(x,y)\right) \\ & + \frac{1}{2}\left(\phi^\ell_{2i-1,2j}(x,y) + \phi^\ell_{2i,2j-1}(x,y) + \phi^\ell_{2i+1,2j}(x,y) + \phi^\ell_{2i,2j+1}(x,y)\right). \end{aligned} \tag{8.54}$$

Thus, for each $\mathbf{v}^\ell \in \mathcal{V}_\ell, \mathbf{f}^\ell \in \mathcal{V}'_\ell = \mathcal{V}_\ell$, we have

$$\mathbf{v}^\ell(x,y) = \sum_{i=1}^{m_\ell}\sum_{j=1}^{n_\ell} v^\ell_{i,j}\phi^\ell_{i,j}(x,y), \quad \mathbf{f}^\ell(x,y) = \sum_{i=1}^{m_\ell}\sum_{j=1}^{n_\ell} f^\ell_{i,j}\psi^\ell_{i,j}(x,y), \tag{8.55}$$

where

$$v^\ell_{i,j} = \mathbf{v}^\ell(x^\ell_i, y^\ell_j), \quad f^\ell_{i,j} = (\mathbf{f}^\ell, \phi^\ell_{i,j})_{L^2(\Omega)}. \tag{8.56}$$

Let us introduce the following tensors:

$$v^\ell = (v^\ell_{i,j}), \quad f^\ell = (f^\ell_{i,j}), \quad \phi^\ell = (\phi^\ell_{i,j}), \quad \psi^\ell = (\psi^\ell_{i,j}). \tag{8.57}$$

The following identities obviously hold:

$$\mathbf{v}^\ell = (v^\ell, \phi^\ell)_{l^2}, \quad \mathbf{f}^\ell = (f^\ell, \psi^\ell)_{l^2}, \quad (\mathbf{f}^\ell, \mathbf{v}^\ell)_{L^2(\Omega)} = (f^\ell, v^\ell)_{l^2}.$$

Denote $\phi^\ell = (\phi^\ell_{i,j}) \in \mathbb{R}^{m_\ell \times n_\ell}$, by the definitions of convolution (10.17) and stride (10.10), (8.53) means that

$$\phi^{\ell+1} = R *_2 \phi^\ell,$$

where

$$R = \begin{cases} \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix} & \text{for bilinear functions;} \\ \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} & \text{for linear functions.} \end{cases} \tag{8.58}$$

8.6 Deconvolution

For any linear mapping $C : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m'\times n'}$, its transpose is the unique linear mapping $C^\top : \mathbb{R}^{m'\times n'} \mapsto \mathbb{R}^{m\times n}$ satisfying

$$(C^\top u, v)_{l^2} = (u, Cv)_{l^2} \quad \forall\, u \in \mathbb{R}^{m'\times n'}, v \in \mathbb{R}^{m\times n}$$

Associated with any kernel K, a deconvolution is defined as the transpose of convolution with stride 2 with respect to the l^2-inner product as:

$$(u, K *_2^\top v)_{l^2} = (K *_2 u, v)_{l^2}, \tag{8.59}$$

with

$$u \in \mathbb{R}^{m\times n} \quad \text{and} \quad v \in \mathbb{R}^{\frac{m+1}{2}\times\frac{m+1}{2}}. \tag{8.60}$$

Lemma 24. *For any $K \in \mathbb{R}^{(2k+1)\times(2k+1)}$,*

$$K *_2^\top = \tilde{K} * \mathcal{S}^\top, \tag{8.61}$$

where $\tilde{K}$ is defined as

$$\tilde{K}_{p,q} = K_{-p,-q}, \quad p, q = -k : k. \tag{8.62}$$

Intuitively, if we take $K_{0,0}$ as the center for the convolutional kernel K, then $\tilde{K}$ is the central symmetry of K. In 2D case, it can also be understood as the rotation of π with respect to the center $K_{0,0}$.

Recalling the definition of deconvolution in (8.59), we have

$$(u, K *_2^\top v)_{\ell^2} = (K *_2 u, v)_{\ell^2} = (\mathcal{S}\mathcal{C}_K u, v)_{\ell^2} = (u, \mathcal{C}_K^\top \mathcal{S}^\top v)_{\ell^2}, \tag{8.63}$$

with definition

$$\mathcal{S}^\top : \mathbb{R}^{\frac{m+1}{2}\times\frac{n+1}{2}} \mapsto \mathbb{R}^{m\times n}, \tag{8.64}$$

and

$$[\mathcal{S}^\top(f)]_{i,j} = \begin{cases} 0 & \text{if i or j is even,} \\ f_{i/2,j/2}, & \text{else.} \end{cases} \tag{8.65}$$

Thus to say, we have the simple version of the deconvolution for $K*$ as

$$K *_2^\top v = \mathcal{C}_K^\top \circ \mathcal{S}^\top(v) = \mathcal{C}_{\tilde{K}} \circ \mathcal{S}^\top(v) = \tilde{K} * \mathcal{S}^\top(v), \tag{8.66}$$

thus to say

$$K*_2^\top = \tilde{K} * \mathcal{S}^\top. \tag{8.67}$$

In short, we have the next decomposition

- convolution with stride = stride ∘ convolution,
- deconvolution with stride = transposed convolution ∘ transposed stride = convolution with the central symmetry of original kernel ∘ transposed stride.

Theorem 19. *Let us consider*

$$K = (K_{p,q}), \quad p, q = -1, 0, 1. \tag{8.68}$$

Then we have

$$K *_2^\top v = \tilde{K} * \mathcal{S}^\top(v).$$

As in (8.65) *and the Lemma* 24*, we have the final version is*

$$[K *_2^\top v]_{2i,2j} = K_{0,0} v_{i,j}, \tag{8.69}$$

with

$$[K *_2^\top v]_{2i-1,2j} = K_{0,1} v_{i-1,j} + K_{0,-1} v_{i,j}, \quad [K *_2^\top v]_{2i,2j-1} = K_{1,0} v_{i,j} + K_{-1,0} v_{i,j-1}, \tag{8.70}$$

and

$$[K *_2^\top v]_{2i-1,2j-1} = K_{1,1} v_{i,j} + K_{-1,1} v_{i-1,j} + K_{1,-1} v_{i,j-1} + K_{-1,-1} v_{i-1,j-1}. \tag{8.71}$$

Remark 10. Deconvolution can obviously be also defined for general stride s, but we believe it is sufficient to use $s = 2$ in most applications.

8.7 Linear feature mappings

We consider the following linear mapping

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{8.72}$$

where

$$\mathbf{A} : \mathcal{V} \mapsto \mathcal{V}'. \tag{8.73}$$

We consider the restriction of the mapping $\mathbf{A}_1 \equiv \mathbf{A}$ on the coarser multilevel spaces:

$$\mathbf{A}_\ell : \mathcal{V}_\ell \mapsto \mathcal{V}'_\ell \tag{8.74}$$

and the corresponding equation read as:

$$\mathbf{A}_\ell \mathbf{u}^\ell = \mathbf{f}^\ell. \tag{8.75}$$

In image process, we can view $\mathbf{f}$ as the input images and $\mathbf{u}$ as the extracted features of the original image $\mathbf{f}$. We then view $\mathbf{f}^\ell$ as the projection of images on a coarser resolution and $\mathbf{u}^\ell$ as the extracted features of the coarsened image $\mathbf{f}^\ell$.

One main question is how to obtain coarser images and features defined by (8.75) from the original equation (8.72). We now consider a special technique.

We define $\mathbf{u}^\ell \in \mathcal{V}_\ell$ by

$$(\mathbf{A}\mathbf{u}^\ell, \mathbf{v}^\ell) = (\mathbf{f}, \mathbf{v}^\ell), \quad \forall \mathbf{v}^\ell \in \mathcal{V}_\ell \tag{8.76}$$

Lemma 25. *The restricted $\mathbf{u}^\ell \in \mathcal{V}_\ell$ defined by (8.76) satisfies (8.75) if $\mathbf{A}_\ell : \mathcal{V}_\ell \mapsto \mathcal{V}'_\ell$ and $\mathbf{f}_\ell \in \mathcal{V}'_\ell$ are defined by*

$$(\mathbf{A}_\ell \mathbf{u}^\ell, \mathbf{v}^\ell) = (\mathbf{A}\mathbf{u}^\ell, \mathbf{v}^\ell), \quad \forall \mathbf{v}^\ell \in \mathcal{V}_\ell \tag{8.77}$$

$$(\mathbf{f}^\ell, \mathbf{v}^\ell) = (\mathbf{f}, \mathbf{v}^\ell), \quad \forall \mathbf{v}^\ell \in \mathcal{V}_\ell \tag{8.78}$$

8.7.1 Restriction and prolongation under the convolution notation

Now we derive the restriction and prolongation as follows. We show the details for the case of bilinear functions here. The case of linear function can be shown similarly. Let $f_{i,j}^{\ell+1} = (\mathbf{f}^\ell, \phi_{i,j}^{\ell+1})_{L^2(\Omega)}$, then we have

$$\begin{aligned} f^{\ell+1} &= \int_\Omega \mathbf{f}^\ell \phi^{\ell+1} = \sum_{i=1}^{m_\ell} \sum_{j=1}^{n_\ell} \int_\Omega f_{i,j}^\ell \psi_{i,j}^\ell \left[(R*_2)\phi^\ell\right] = \sum_{i=1}^{m_\ell} \sum_{j=1}^{n_\ell} f_{i,j}^\ell (R*_2) \int_\Omega \psi_{i,j}^\ell \phi^\ell \\ &= \sum_{i=1}^{m_\ell} \sum_{j=1}^{n_\ell} (R*_2) f_{i,j}^\ell e_i e_j^T = R *_2 f^\ell. \end{aligned} \tag{8.79}$$

Hence the restriction

$$R_\ell^{\ell+1} : \mathbb{R}^{m_\ell \times n_\ell} \mapsto \mathbb{R}^{m_{\ell+1} \times n_{\ell+1}}$$

is obtain by $R_\ell^{\ell+1} f^\ell = R *_2 f^\ell$ with $R \in \mathbb{R}^{3\times 3}$ given by (8.58), namely

$$
\begin{aligned}
f_{i,j}^{\ell+1} &= f_{2i,2j}^\ell + \frac{1}{2}(f_{2i-1,2j}^\ell + f_{2i,2j-1}^\ell + f_{2i+1,2j}^\ell + f_{2i,2j+1}^\ell) \\
&\quad + \frac{1}{4}\left(f_{2i-1,2j-1}^\ell + f_{2i+1,2j-1}^\ell + f_{2i+1,2j+1}^\ell + f_{2i-1,2j+1}^\ell\right).
\end{aligned}
\tag{8.80}
$$

Next let $\mathbf{u}^{\ell+1} = \sum_{i=1}^{m_{\ell+1}} \sum_{j=1}^{n_{\ell+1}} u_{i,j}^{\ell+1} \phi_{i,j}^{\ell+1} = (\mathbf{u}^{\ell+1}, \phi^{\ell+1})_{l^2}$, then we have

$$
\begin{aligned}
\mathbf{u}^{\ell+1} &= (u^{\ell+1}, \phi^{\ell+1})_{l^2} = (u^{\ell+1}, R *_2 \phi^\ell)_{l^2} = (R *_2^\top u^{\ell+1}, \phi^\ell)_{l^2} \\
&= \sum_{i=1}^{m_\ell} \sum_{j=1}^{n_\ell} \left(R *_2^\top u^{\ell+1}\right)_{i,j} \phi_{i,j}^\ell.
\end{aligned}
\tag{8.81}
$$

Namely

$$\mathbf{u}^{\ell+1}(x_i^\ell, y_j^\ell) = \left(R *_2^\top u^{\ell+1}\right)_{i,j}.$$

And we obtain the prolongation

$$P_{\ell+1}^\ell = R *_2^\top : \mathbb{R}^{m_{\ell+1} \times n_{\ell+1}} \mapsto \mathbb{R}^{m_\ell \times n_\ell}$$

is defined by

$$u_{2i,2j}^\ell = u_{i,j}^{\ell+1},$$

$$u_{2i-1,2j}^\ell = \frac{1}{2}(u_{i,j}^\ell + u_{i-1,j}^\ell), \quad u_{2i,2j-1}^\ell = \frac{1}{2}(u_{i,j}^{\ell+1} + u_{i,j-1}^{\ell+1})$$

and

$$u_{2i-1,2j-1}^\ell = \frac{1}{4}(u_{i,j}^{\ell+1} + u_{i-1,j}^{\ell+1} + u_{i-1,j-1}^{\ell+1} + u_{i,j-1}^{\ell+1}).$$

In summery, we have the restriction and prolongation as follows:

Lemma 26. *The restriction*

$$R_\ell^{\ell+1} : \mathbb{R}^{m_\ell \times n_\ell} \mapsto \mathbb{R}^{m_{\ell+1} \times n_{\ell+1}} \ \ \textit{is} \ \ R_\ell^{\ell+1} = R *_2$$

and the prolongation

$$P_{\ell+1}^\ell : \mathbb{R}^{m_{\ell+1} \times n_{\ell+1}} \mapsto \mathbb{R}^{m_\ell \times n_\ell} \ \ \textit{is} \ \ P_{\ell+1}^\ell = R *_2^\top$$

where

$$
R = \begin{cases}
\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix} & \textit{for bilinear functions}; \\
\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} & \textit{for linear functions}.
\end{cases}
\tag{8.82}
$$

Let $(\phi^\ell_{i,j})$ is a basis of $\mathcal{V}_\ell$, for any $\mathbf{u}^\ell \in \mathcal{V}_\ell$, then $\mathbf{u}^\ell = \sum_{i=1}^{m_\ell}\sum_{j=1}^{n_\ell} u^\ell_{i,j}\phi^\ell_{i,j}$, and we denote $u^\ell = (u^\ell_{i,j}) \in \mathbb{R}^{m_\ell \times n_\ell}$ the matrix representation of $\mathbf{u}^\ell$ under the basis $(\phi^\ell_{i,j})$. Let $(\psi^\ell_{s,t})$ is a basis of $\mathcal{V}'_\ell$ which is dual to $(\phi^\ell_{i,j})$. Denote $A_\ell = (a^\ell_{stij})$ the tensor representation of $\mathbf{A}_\ell : \mathcal{V}_\ell \mapsto \mathcal{V}'_\ell$ and defined as

$$\mathbf{A}_\ell \phi^\ell_{i,j} = \sum_{s=1}^{m_\ell}\sum_{t=1}^{n_\ell} a^\ell_{stij}\psi^\ell_{s,t}.$$

Hence $a^\ell_{stij} = (\mathbf{A}_\ell \phi^\ell_{i,j}, \phi^\ell_{s,t})$. From $(\mathbf{A}_{\ell+1}\phi^{\ell+1}_{i,j}, \phi^{\ell+1}_{s,t}) = (\mathbf{A}_\ell \phi^{\ell+1}_{i,j}, \phi^{\ell+1}_{s,t})$, we have

$$a^{\ell+1}_{stij} = \sum_{r=1}^{m_\ell}\sum_{q=1}^{n_\ell}\left(\sum_{k=1}^{m_\ell}\sum_{m=1}^{n_\ell} P^{\ell,\ell+1}_{kmij} a^\ell_{rqkm}\right) P^{\ell,\ell+1}_{rqst}$$

Where $P^{\ell,\ell+1} = (P^{\ell,\ell+1}_{rqst})$ is the tensor representation of the prolongation $P^\ell_{\ell+1}$.

Consider the finite element method on two different grids $\mathcal{T}_\ell$, $\mathcal{T}_{\ell+1}$, $h_{\ell+1} = 2h_\ell, \mathcal{V}_{\ell+1} \subset \mathcal{V}_\ell$. With the restriction $R^{\ell+1}_\ell$ and prolongation $P^\ell_{\ell+1}$ obtained in Lemma 26, we have the following relationship to define coarse operation

$$\begin{aligned} A_{\ell+1} &= R^{\ell+1}_\ell A_\ell P^\ell_{\ell+1}. \\ &= R *_2 A_\ell * (R *_2^\top), \quad (\ell = 1 : J-1), \end{aligned} \tag{8.83}$$

with $A_1 = A$.

Theorem 20. *If R is consistent with A_ℓ which means that R should be linear or bilinear as A_ℓ, then we have the $A_{\ell+1}$ operation in coarse grid defined in (8.83) is the same with A_ℓ.*

Proof. For any $u_{\ell+1}$ and $v_{\ell+1}$ in $\mathcal{V}_{\ell+1}$, it remains to prove that

$$(A_\ell P^\ell_{\ell+1} u_{\ell+1}, P^\ell_{\ell+1} v_{\ell+1}) = (A_{\ell+1} u_{\ell+1}, v_{\ell+1})$$

where A_ℓ and $A_{\ell+1}$ are the tensor representation of $\mathbf{A}_\ell$ and $\mathbf{A}_{\ell+1}$.

We can also view them as convolutions. By the definition of operators $R^{\ell+1}_\ell$ and $P^\ell_{\ell+1}$, a direct computation gives the above result. □

Proof. By the definition above, we have that

$$A_{\ell+1}(v) = \mathcal{S}\left((R * A_\ell * R) * \mathcal{S}^\top(v)\right), \tag{8.84}$$

because of the properties of convolution we know that

$$(R * A_\ell * R)* = K*, \tag{8.85}$$

for some

$$K \in \mathbb{R}^{7\times 7}.$$

Then we have the next computation for $A_{\ell+1}(v)$

$$
\begin{aligned}
[A_{\ell+1}(v)]_{i,j} &= [\mathcal{S}\Big((R * A_\ell * R) * \mathcal{S}^\top(v)\Big)]_{i,j}, \\
&= [K * \mathcal{S}^\top(v)]_{2i,2j}, \\
&= \sum_{p,q=-3}^{3} [\mathcal{S}^\top(v)]_{2i+p,2j+q} K_{p,q}, \\
&= \sum_{p,q=-1}^{1} [\mathcal{S}^\top(v)]_{2(i+p),2(j+q)} K_{2p,2q}, \\
&= \sum_{p,q=-1}^{1} v_{i+p,j+q} \hat{K}_{p,q},
\end{aligned} \tag{8.86}
$$

Thus to say, we have

$$
A_{\ell+1}(v) = \hat{K} * v, \tag{8.87}
$$

with

$$
\hat{K}_{p,q} = K_{2p,2q}, \quad p, q = -1, 0, 1,
$$

with K is defined in (8.85).

Then by the direct computation of (8.85) as

$$
(R * A_\ell * R) * = K *
$$

and take the even index we have that

$$
A_{\ell+1} = \hat{K} = A_\ell, \tag{8.88}
$$

if R is consistent with A_ℓ which means that R should be linear or bi-linear as A_ℓ. □

8.8 Multigrid for finite element methods

By the definition of convolution (10.17), we can rewrite (8.41) and (8.45) as follows:

$$
A* : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}, \quad A*u = f, \quad \Leftrightarrow \quad u = \arg\min J(v) = \arg\min \left(\frac{1}{2}(A * v, v) - (f, v)_{\ell^2}\right) \tag{8.89}
$$

where $u = (u_{ij})$, $f = (f_{ij})$,

$$A = \begin{cases} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \text{for linear finite element,} \\ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} & \text{for bilinear finite element.} \end{cases} \tag{8.90}$$

It is easy to see that

$$\nabla J(v) = A * v - f := -r, \quad r = f - A * v.$$

Applying the gradient descent method to (8.43), we obtain the following iterative method:

$$u^{k+1} = u^k + \eta r^k, \quad r^k = f - A * u^k. \tag{8.91}$$

Here we can clearly see that gradient descent method is equivalent to damped Jacobi method. Usually, we call this as smoother.

Lemma 27. *The gradient descent method (8.91) for linear finite element converges if $\eta = \frac{1}{8}$, and the one for bilinear finite element converges if $\eta = \frac{1}{16}$. Furthermore, the high frequence in $u - u^k$ are damped very rapidly.*

Proof. According to (8.91) ,

$$u^{k+1} - u = (I - \eta A) * (u^k - u).$$

The gradient descent method (8.91) converges if $\rho((I - \eta A)*) < 1$, namely $\eta\rho(A*) < 2$. For linear finite element, if $\eta = \frac{1}{8}$, $\rho(A*) < 8$, thus the gradient descent method (8.91) converges. For bilinear finite element, if $\eta = \frac{1}{16}$, $\rho(A*) < 16$, thus the gradient descent method (8.91) converges.

For linear finite element, let (λ_i, v_i) satisfy $A * v_i = \lambda_i v_i$ and $0 < \lambda_1 \leq \lambda_2 \leq \cdots \lambda_N$ with $N = n^2$. Expand the error $u^k - u$ in terms of eigenvectors v_i, namely,

$$u^k - u = \sum_{i=1}^{N} a_i^k v_i.$$

Then

$$u^k - u = \sum_{i=1}^{N} a_i^k (1 - \eta\lambda_i) v_i = \sum_{i=1}^{N} a_i^0 (1 - \eta\lambda_i)^k v_i.$$

According to Proposition 1, it is easy to see that

$$1 - \frac{1}{8}\lambda_N \approx \frac{\pi^2}{4(n+1)^2} \ll 1.$$

For $\eta = \frac{1}{8}$, the coefficient $a_N^0(1 - \frac{1}{8}\lambda_N)^k$ of v_N approximates to zero much faster. This means that high frequency in the error will damp rapidly. □

For an initial guess u^0, the left picture in Fig 8.14 plots the error $u - u^0$ and the the right one plots the error $u - u^1$. Fig 8.14 shows that the high frequency in the error of the initial guess u^0 is damped after one step of smoothing and results in a smoother error $u - u^1$. Next, we make the following particular choice:

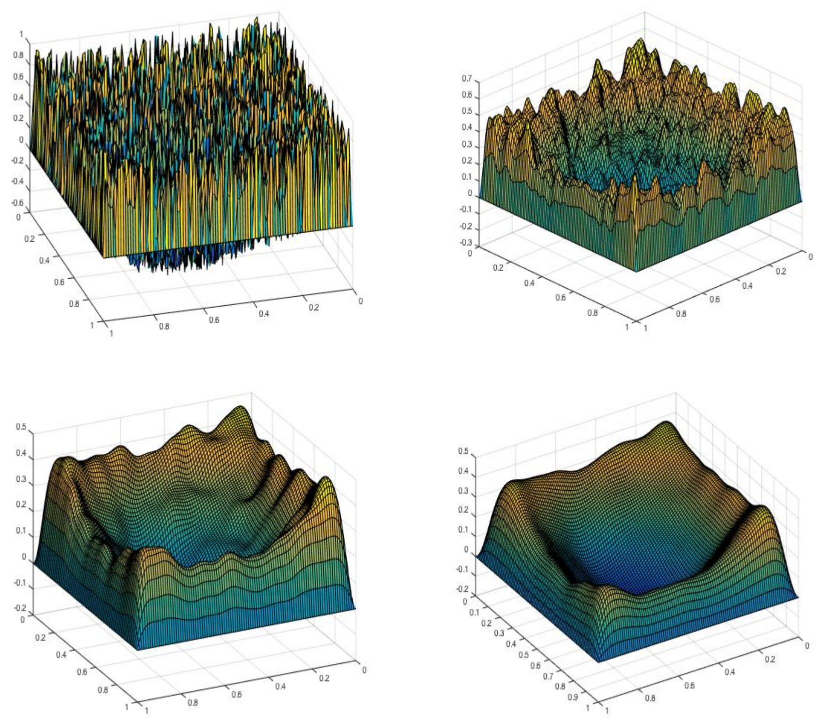


Fig. 8.14. The errors of an random initial guess u^0, u^{10}, u^{50} and u^{100}.

$$\eta = \frac{1}{8}. \tag{8.92}$$

The gradient descent method can be written in terms of $S_0 : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$ satisfying

$$u^1 = (S_0 f) = \frac{1}{8}f, \tag{8.93}$$

for equation (8.41) with initial guess zero. If we apply this method twice, then

$$u^2 = S_1(f) = S_0 f + S_0(f - A * (S_0 f)),$$

with element-wise form

$$u^2_{i,j} = \frac{3}{16} f_{i,j} + \frac{1}{64}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}). \tag{8.94}$$

Then by the definition of convolution (10.17), we have

$$u^1 = S_0 * f \quad u^2 = S_1 * f. \tag{8.95}$$

with

$$S_0 = \frac{1}{8}, \tag{8.96}$$

and

$$S_1 = \frac{1}{64}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 12 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{8.97}$$

Next, we make the following particular choice for the bilinear case:

$$\eta = \frac{1}{16}. \tag{8.98}$$

The gradient descent method can be written in terms of $S_0 : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$ satisfying

$$u^1 = (S_0 f)_{i,j} = \frac{1}{16} f_{i,j}, \tag{8.99}$$

for equation (8.45) with initial guess zero. If we apply this method twice, then

$$u^2 = S_1(f) = S_0 f + S_0(f - A * (S_0 f)),$$

with element-wise form

$$u^2_{i,j} = \frac{3}{32} f_{i,j} + \frac{1}{256}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} + f_{i+1,j+1} + f_{i-1,j-1} + f_{i-1,j+1} + f_{i+1,j-1}). \tag{8.100}$$

Then by the definition of convolution (10.17), we have

$$u^1 = S_0 * f \quad u^2 = S_1 * f. \tag{8.101}$$

with

$$S_0 = \frac{1}{16}, \tag{8.102}$$

and

$$S_1 = \frac{1}{256}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 24 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{8.103}$$

We note the gradient descent method (8.91) can be written as

$$u^k = u^{k-1} + S_0 * (f - A * u^{k-1}). \tag{8.104}$$

$$u^{2k} = u^{2(k-1)} + S_1 * (f - A * u^{2(k-1)}). \tag{8.105}$$

We can sometimes use S_1 and the above identity to define a smoother directly, namely

$$u^k = u^{k-1} + S_1 * (f - A * u^{k-1}). \tag{8.106}$$

Similarly, with the smoother obtained by (8.101), we can define $S^\ell : \mathbb{R}^{m_\ell \times n_\ell} \mapsto \mathbb{R}^{m_\ell \times n_\ell}$.

First solve the problem on the fine grid $\mathcal{T}_\ell$, denote the solution by u_ℓ, the error by $e = u - u_\ell$ and the residual

$$r^\ell = f - A_\ell * u_\ell.$$

It is obvious that $A_\ell * e_\ell = r^\ell$. We need to solve the residual equation

$$A_\ell * e_\ell = r^\ell. \tag{8.107}$$

The idea of multigrid method is to solve this residual equation on the coarse grid space $\mathcal{V}_{\ell+1}$ and repeat the process until coarsest grid.

We note that the restriction of (8.107) to the coarse level $\ell + 1$ is

$$A_{\ell+1} * e_{\ell+1} = r^{\ell+1}$$

with $r^{\ell+1} = R_\ell^{\ell+1} r^\ell = R * r^\ell$.

Denote the finite element approximation to e_ℓ on the coarse grid $\mathcal{T}_{\ell+1}$ by $e_{\ell+1}$. Interpolate the error $e_{\ell+1}$ back to the fine space $\mathcal{V}_\ell$ and add the resulting residual to $u_{\ell+1}$, that is

$$u_\ell \leftarrow u_\ell + P_{\ell+1}^\ell e_{\ell+1}$$

Now using the smoother S^ℓ, prolongation $P_{\ell+1}^\ell$, restriction $R_\ell^{\ell+1}$ and mapping A^ℓ as given in (8.83), we can formulate the following algorithm as a major component of a multigrid algorithm.

Algorithm 13 $(u^1, u^2, \cdots, u^J) = \mathrm{MG0}(f; u^0; J, \nu_1, \cdots, \nu_J)$

Set up

$$f^1 = f, \quad u^1 = u^0.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

for $i = 1 : \nu_\ell$ **do**

$$u^\ell \leftarrow u^\ell + S^\ell * (f^\ell - A_\ell * u^\ell). \tag{8.108}$$

end for

Form restricted residual and set initial guess:

$$u^{\ell+1,0} \leftarrow 0, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * u^\ell), A_{\ell+1} = R *_2 A_\ell * (R*_2^\top).$$

end for

Here S^ℓ can be chosen as S_0 or S_1 as definition in (8.102) and (8.103).

Using the above algorithm, there are different multigrid algorithms such as: \-cycle, V-cycle and W-cycle. Let us now only give one special form of multigrid algorithm for solving (8.35) as follows.

Algorithm 14 $u = \mathrm{MG1}(f; u^0; J, \nu_1, \cdots, \nu_J)$

$$u \leftarrow u^0.$$

$$(u^1, u^2, \cdots, u^J) = \mathrm{MG0}(f; u; J, \nu_1, \cdots, \nu_J).$$

Prolongation and restriction from coarse to fine level

for $\ell = J - 1 : 1$ **do**

$$u^\ell \leftarrow u^\ell + R *_2^\top u^{\ell+1}.$$

end for

$$u \leftarrow u^1.$$

If we add the post-smoothing with a symmetric form which means we use $[S^\ell *]^\top$ as the smoother, then we can get the V-cycle version multigrid algorithm.

Algorithm 15 $u = \mathrm{MG2}(f; u^0; J, \nu_1, \cdots, \nu_J; \nu_1', \cdots, \nu_J')$

$$(u^1, u^2, \cdots, u^J) = \mathrm{MG0}(f; u^0; J, \nu_1, \cdots, \nu_J).$$

Prolongation and restriction from coarse to fine level

for $\ell = J - 1 : 1$ **do**

$$u^\ell \leftarrow u^\ell + R *_2^\top u^{\ell+1}.$$

for $i = 1 : \nu'_\ell$ **do**

$$u^\ell \leftarrow u^\ell + \tilde{S}^\ell * (f^\ell - A_\ell * u^\ell)$$

end for

end for

$$u = u^1.$$

Here $\tilde{S}^\ell$ means the central symmetry of kernel for smoother S^ℓ as in the definition of (8.62) in Lemma 24.

We note that

$$\mathrm{MG1}(f; u^0; J, \nu_1, \cdots, \nu_J) = \mathrm{MG2}(f; u^0; J, \nu_1, \cdots, \nu_J; 0, \cdots, 0).$$

Either MG1 or MG2 only represents one cycle in a multigrid process. There many different ways to use this basis multigrid cycle. For a given iterate u, we need to define a metric to measure the accuracy of u. One way to define it is:

$$\mathrm{error}(u) = \|f - A * u\| / \|f - A * u^0\|.$$

Sometimes, when we debug a code, we can first try to find the exact solution. u_{exact}, and then define

$$\mathrm{error}(u) = \|u_{\mathrm{exact}} - u\|_A.$$

Below is one example fo algorithm for application of the basic multigrid cycle, say MG1:

Algorithm 16 $u = \mathrm{multigrid1}(f; u^0; J, \nu_1, \cdots, \nu_J; \mathrm{tol});$

$$u \leftarrow u^0.$$

while $\mathrm{error}(u) \geq tol$ **do**

$$u \leftarrow u + \mathrm{MG1}(f - Au; 0; \nu_1, \cdots, \nu_J).$$

end while

A slightly more general multigrid method.

---

1. Initialization of inputs

$$g_1 \leftarrow g, \quad u_1 \leftarrow \text{random}.$$

2. Smoothing and restriction
   - For $\ell = 1 : J$

- For $i = 1 : \nu_\ell$

$$u_\ell \leftarrow u_\ell + S_\ell * (g_\ell - A_\ell * u_\ell). \tag{8.109}$$

- Form restricted residual and set initial guess:

$$u_{\ell+1,0} \leftarrow \Pi_\ell^{\ell+1} u_\ell, \quad g_{\ell+1} \leftarrow R_\ell *_2 (g_\ell - A_\ell * u_\ell) + A_{\ell+1} * u_{\ell+1}^0,$$

3. Prolongation with post-smoothing
   - For $\ell = J - 1 : 1$

$$u_\ell \leftarrow u_\ell + R_\ell *_2^\top (u_{\ell+1} - u_{\ell+1}^0).$$

   - For $i = 1 : \nu'_\ell$

$$u_\ell \leftarrow u_\ell + S'_\ell * (g_\ell - A_\ell * u_\ell)$$

4. Output

$$u_1$$

Theorem 21. *If A^ℓ, $R_\ell^{\ell+1}$ and $B^{\ell,i} = S^\ell$ are all linear operations as described in multigrid method in §8.3 and all $\sigma = \mathrm{id}$ in Algorithm 21. Then Algorithm 13 is equivalent to Algorithm 21 with any choice of $\Pi_\ell^{\ell+1}$.*

Proof. Here we replace $u^{\ell,i}$ and f^ℓ by $\tilde{u}^{\ell,i}$ and $\tilde{f}^\ell$ in MgNet. What we want to prove are

$$\tilde{f}^\ell = f^\ell + A_\ell \tilde{u}^{\ell,0} \quad \text{and} \quad u^{\ell,i} = \tilde{u}^{\ell,i} - \tilde{u}^{\ell,0}, \tag{8.110}$$

with $u^{\ell,i}$, f^ℓ in Algorithm 13 and $\tilde{u}^{\ell,i}$, $\tilde{f}^\ell$ in Algorithm 21 for any choice of $\Pi_\ell^{\ell+1}$. We prove this result by induction.

- It is easy to check that $\ell = 1$ is right by taking $\theta = \mathrm{id}$.
- Once the above equation (8.110) is right for ℓ, let us prove the corresponded result for $\ell + 1$.
  - For $\tilde{f}^{\ell+1}$, as the definition in Algorithm 21, we have

$$\begin{aligned}
\tilde{f}^{\ell+1} &= R_\ell^{\ell+1}(\tilde{f}^\ell - A^\ell \tilde{u}^{\ell,\nu_\ell}) + A^{\ell+1}\tilde{u}^{\ell+1,0}, \\
&= R_\ell^{\ell+1}(f^\ell + A^\ell \tilde{u}^{\ell,0} - A^\ell \tilde{u}^{\ell,\nu_\ell}) + A^{\ell+1}\tilde{u}^{\ell+1,0} \\
&= R_\ell^{\ell+1}(f^\ell - A^\ell(\tilde{u}^{\ell,\nu_\ell} - u^{\ell,0})) + A^{\ell+1}\tilde{u}^{\ell+1,0} \\
&= R_\ell^{\ell+1}(f^\ell - A^\ell u^{\ell,\nu_\ell}) + A^{\ell+1}\tilde{u}^{\ell+1,0}, \\
&= f^{\ell+1} + A^{\ell+1}\tilde{u}^{\ell+1,0}.
\end{aligned}$$

  - For $u^{\ell+1,i}$, first we have

$$u^{\ell+1,0} = 0 = \tilde{u}^{\ell+1,0} - \tilde{u}^{\ell+1,0},$$

  then we prove

$$u^{\ell+1,i} = \tilde{u}^{\ell+1,i} - \tilde{u}^{\ell+1,0} \tag{8.111}$$

by induction for i.
We assume (8.111) holds for $0, 1, \cdots, i-1$. Let us miner $\tilde{u}^{\ell+1,0}$ in both sides of the smoothing process (9.3) in Algorithm 21. Then we have

$$\begin{aligned}
\tilde{u}^{\ell+1,i} - \tilde{u}^{\ell+1,0} &= \tilde{u}^{\ell+1,i-1} - \tilde{u}^{\ell+1,0} + B^{\ell+1,i}(\tilde{f}^{\ell+1} - A^{\ell+1}\tilde{u}^{\ell+1,i-1}),\\
&= \tilde{u}^{\ell+1,i-1} - \tilde{u}^{\ell+1,0} + B^{\ell+1,i}(f^{\ell+1} + A^{\ell+1}\tilde{u}^{\ell+1,0} - A^{\ell+1}\tilde{u}^{\ell+1,i-1}),\\
&= u^{\ell+1,i-1} + B^{\ell+1,i}(f^{\ell+1} - A^{\ell+1}u^{\ell+1,i-1}).
\end{aligned}$$

This is exact the smoothing process in Algorithm 13 as we take $B^{\ell+1,i} = S^{\ell+1}$.

□

8.9 ReLU multigrid method for nonnegative solution

Considering $f = (1, 1, ..., 1)^\top$,

Algorithm 18 $(u^1, u^2, \cdots, u^J) = \mathrm{MG0}(f; u^0; J, \nu_1, \cdots, \nu_J)$

Set up

$$f^1 = f, \quad u^1 = u^0.$$

Smoothing and restriction from fine to coarse level (nested)
for $\ell = 1 : J$ **do**
  for $i = 1 : \nu_\ell$ **do**

$$u^\ell \leftarrow u^\ell + S^\ell * \mathrm{Relu}((f^\ell - A_\ell * u^\ell)). \tag{8.112}$$

  end for
  Form restricted residual and set initial guess:

$$u^{\ell+1,0} \leftarrow 0, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * u^\ell), A_{\ell+1} = R *_2 A_\ell * (R *_2^\top).$$

end for

Algorithm 5 is not convergent.

Algorithm 19 $(u^1, u^2, \cdots, u^J) = \mathrm{MG0}(f; u^0; J, \nu_1, \cdots, \nu_J)$

Set up

$$f^1 = f, \quad u^1 = u^0.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

for $i = 1 : \nu_\ell$ **do**

$$u^\ell \leftarrow u^\ell + \mathrm{Relu}(S^\ell * (f^\ell - A_\ell * u^\ell)). \tag{8.113}$$

end for

Form restricted residual and set initial guess:

$$u^{\ell+1,0} \leftarrow 0, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * u^\ell), A_{\ell+1} = R *_2 A_\ell * (R *_2^\top).$$

end for

Algorithm 6 is convergent, the iterative steps is list below. $l = 2$

J	MG	Algorithm 6
2	15	27
3	16	35
4	17	38

8.9.1 Π is interpolation

Not Convergent

$$u^{\ell,i} \leftarrow u^{\ell,i-1} + \mathrm{Relu} \circ B_{\ell,i}(f^\ell - A^\ell(u^{\ell,i-1})). \tag{8.114}$$

$$u^{\ell,i} \leftarrow u^{\ell,i-1} + B_{\ell,i} \circ \mathrm{Relu}(f^\ell - A^\ell(u^{\ell,i-1})). \tag{8.115}$$

Almost the same as without Relu

$$u^{\ell,i} \leftarrow u^{\ell,i-1} + B_{\ell,i}(f^\ell - \mathrm{Relu} \circ A^\ell(u^{\ell,i-1})). \tag{8.116}$$

$$u^{\ell,i} \leftarrow u^{\ell,i-1} + B_{\ell,i}(f^\ell - A^\ell \circ \mathrm{Relu}(u^{\ell,i-1})). \tag{8.117}$$

8.10 Multigrid methods for nonlinear problem

In classification, the key problem can be reduced to find the representation(feature) for high dimension image for classifying. Her we propose to solve the next unbalanced nonlinear system

$$F(u) = f, \tag{8.118}$$

for finding the suitable feature representation $u \in \mathbb{R}^c$ for image $f \in \mathbb{R}^n$. The rationality of system can be traced back to the low-dimension assumption that natural image need to be concentrated on a low-dimension manifold with respect to the pixel space.

Model problem

$$\begin{cases} -\nabla \cdot (a(u)\nabla u) = f, & x \in \Omega \\ \qquad u = 0 & \text{on} \partial\Omega \end{cases} \tag{8.119}$$

Define

$$(A(u), v) = ((a(u)\nabla u, \nabla v)). \tag{8.120}$$

Then we have

$$A(u) = f. \tag{8.121}$$

Now the discretization problem reads: Find $u_h \in V_h$ such that

$$(A_h(u_h), v_h) = ((a(u_h)\nabla u_h, \nabla v_h)) \quad \forall v_h \in V_h. \tag{8.122}$$

Next we think about the nonlinear multigrid methods. **Question:** Given $u_h^0 \in V_h, u_h^0 \approx u_h$, we need to find a correction $e_{2h} \in V_{2h}$, s.t.

1. $u_h^1 = u_h^0 + e_{2h} \approx u_h$;
2. $e_{2h} = 0$ if $u_h^0 = u_h$;
3. The solution of e_{2h} is obtained by solving the coarse grid equation

$$A_{2h}(w_{2h}) = g_{2h}.$$

Solution: $Q_{2h}A_h(u_h^0 + e_{2h}) = f_{2h}$.

(i) $Q_{2h}[A_h(u_h^0 + e_{2h}) - A_h(u_h^0)] = Q_{2h}[f_{2h} - A_h(u_h^0)]$.

Now we use the approximation

$$Q_{2h}A_h(u_h^0 + e_{2h}) \approx A_{2h}(u_{2h}^0 + e_{2h}). \tag{8.123}$$

Since we need to find approximation of $Q_{2h}A_h(u_h^0)$, s.t. $e_{2h} = 0$ if $h_h^0 = u_h$;

Then we must have

$$Q_{2h}A_h(u_h^0) \approx A_{2h}(u_{2h}^0).$$

To solve the overdetermined nonlinear system, we can try the multilevel ideas with smoothing is fine level, and truncated it into coarse level by recursion. One strategy to involve the multi-scale idea is to "soothing" in the fine level, and restrict it as a good approximation in the coarse level, this idea can be found in many literatures especially for multigrid methods in optimization [**? ?**]. So, there is a more general nonlinear multigrid scheme - fully approximation scheme (FAS) [**? ?**], which can be considered as the generalization of linear multigrid scheme 13. Here we show a FAS scheme with Slash cycle as

Algorithm 20 $u =$ Bslash-FAS$(u^{1,0}, f, J, m_1, \cdots, m_J)$

Initialization

$$f^1 = f.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

Nonlinear relaxation on level ℓ:

for $i = 1 : m_j$ **do**

$$u^{\ell,i} = u^{\ell,i-1} + [\nabla F^\ell(u^{\ell,i-1})]^{-1}(f^\ell - F^\ell(u^{\ell,i-1})).$$

end for

Form the initial guess and right side term for level $\ell + 1$:

$$u^{\ell+1,0} = \Pi_\ell^{\ell+1} u^{\ell,m_\ell}, \quad f^{\ell+1} = R_\ell^{\ell+1}(f^\ell - F^\ell(u^{\ell,m_\ell})) + F^{\ell+1}(u^{\ell+1,0}).$$

end for

Prolongation and correction from coarse to fine level

for $\ell = J - 1 : 1$ **do**

Form error in coarse level

$$e^{\ell+1} = u^{\ell+1,m_{\ell+1}} - u^{\ell+1,0}.$$

Correction by using error in coarse level

$$u^{\ell,m_\ell} \leftarrow u^{\ell,m_\ell} + P_{\ell+1}^\ell e^{\ell+1}.$$

end for

If the problem in (8.118) is linear, then we have the next theorem to show that this FAS scheme is consist with the classical multigrid methods for linear systems.

Theorem 22. *If $F(u)$ in (8.118) is a linear operation. Then Algorithm 20 is equivalent to Algorithm 11 with any choice of $\Pi_\ell^{\ell+1}$.*

9

MgNet: a Unified Framework for CNN and MG

9.1 MgNet: a new network structure

In this section, we introduce a new neural network structure, named as MgNet, motivated by the multigrid algorithm, Algorithm 13 and its nonlinear version in Algorithm 20, as discussed in the previous section.

Here we recall the most important two structures in multigrid

1. iterative scheme (for linear system)

$$u^{\ell,i} = u^{\ell,i-1} + B^{\ell,i}(f^{\ell} - A^{\ell} * u^{\ell,i-1}). \tag{9.1}$$

2. interpolation and restriction

$$u^{\ell+1,0} = \Pi_{\ell}^{\ell+1} *_2 u^{\ell}, \quad f^{\ell+1} = R_{\ell}^{\ell+1} *_2 (f^{\ell} - A^{\ell}(u^{\ell})) + A^{\ell+1} * u^{\ell+1,0}. \tag{9.2}$$

Considering the fine to coarse process of multigrid with the aforementioned two structures in (9.1) and (9.2). We are now in a position to state the main algorithm, namely MgNet by just put some nonlinear activation function σ in some places.

Algorithm 21 $u^J = \mathrm{MgNet}(f; J, \nu_1, \cdots, \nu_J)$

Initialization: $f^1 = \theta(f)$, $u^{1,0} = 0$
for $\ell = 1 : J$ **do**
  for $i = 1 : \nu_\ell$ **do**

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} * \sigma(f^{\ell} - A^{\ell} * u^{\ell,i-1}). \tag{9.3}$$

  end for
  Note $u^{\ell} = u^{\ell,\nu_\ell}$

$$u^{\ell+1,0} = \Pi_{\ell}^{\ell+1} *_2 u^{\ell} \tag{9.4}$$

$$f^{\ell+1} = R_{\ell}^{\ell+1} *_2 (f^{\ell} - A^{\ell}(u^{\ell})) + A^{\ell+1} * u^{\ell+1,0}. \tag{9.5}$$

end for

The main steps in MgNet can be understood as solving the following data-feature mappings in each grid ℓ:

$$A^\ell * u^\ell = f^\ell, \quad \ell = 1 : J, \tag{9.6}$$

where

$$f^\ell \in \mathbb{R}^{c_\ell \times m_\ell \times n_\ell}, \tag{9.7}$$

and

$$u^\ell \in \mathbb{R}^{h_\ell \times m_\ell \times n_\ell}, \tag{9.8}$$

with constrain

$$u > 0. \tag{9.9}$$

More details about this basic assumption in image classification can be found in [**?**].

Here, the next diagram gives a brief illustration for the above structure.

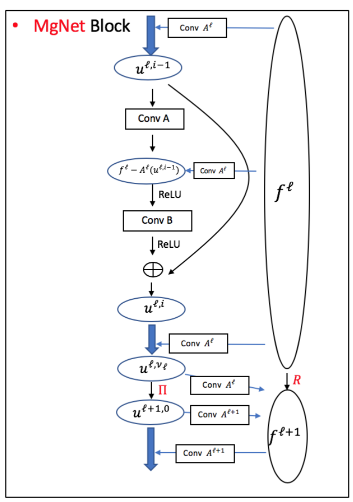


Fig. 9.1. Structure of MgNet

The first property of MgNet is that it recovers the multigrid methods.

Despite of the simplicity look of Algorithm 21, there are rich mathematical structures and variants which we briefly discuss below.

9.1.1 Initialization: feature space channels

Initially for $\ell = 1$, we take $m_1 = m$ and $n_1 = n$ and we may define the linear mapping

$$\theta : \mathbb{R}^{c\times m\times n} \mapsto \mathbb{R}^{c_1\times m_1\times n_1}, \tag{9.10}$$

to obtain $f^1 = \theta(f)$ with c given in (1.2) changed to the channel of the initial data space to c_1. Usually

$$c_1 \geq c. \tag{9.11}$$

One possibility is that we choose $c_1 = c$. In this case, we choose θ =identity. But in general, we may need to choose $c_1 \gg c$. One possible advantage of preprocessing the RGB ($c = 3$) to different color spaces is that we can better choose what kind of features the CNN can detect, and under what conditions those detections will be invariant.

One possibility of understanding and modifying this step is to decompose the data f into a number of more specialized data

$$f = \sum_{k=1}^{c_1} \xi_k f_k^1 = \xi^T f^1. \tag{9.12}$$

We may use some knowledge from image processing or physics to design a procedure to obtain the right decomposition of (9.12), or we can just train it. Conceivably, we may view $f^1 = \theta(f)$ as a special approximation solution of (9.12) with the same sparsity pattern to ξ.

9.1.2 Extracted Units: u^ℓ and channels

The first new feature and the main new ingredient in the proposed neural network is the introduction of feature variables u^ℓ in (9.8), which will be known as the extracted units.

We emphasize that the extracted-units u^ℓ and the data f^ℓ can have different numbers of channels:

$$u^\ell \in \mathbb{R}^{c_{u,\ell}\times m_\ell\times n_\ell}, \quad f^\ell \in \mathbb{R}^{c_{f,\ell}\times m_\ell\times n_\ell} \tag{9.13}$$

One possibility is that the number of channels for both u and f remain unchanged in different grids:

$$c_{f,\ell} = c_f, \quad \ell = 1 : J, \tag{9.14}$$

and

$$c_{u,\ell} = c_u, \quad \ell = 1 : J. \tag{9.15}$$

Both c_f and c_u are two super-parameters that need to be tuned, and we may even take $c_u = c_f$.

9.1.3 Poolings: $\Pi_{\ell+1}^{\ell}$ and $R_{\ell+1}^{\ell}$

The pooling $\Pi_{\ell+1}^{\ell}$ in (9.4) and $R_{\ell+1}^{\ell}$ in (9.5) are in general different. They can be trained in general, but they may be a priori chosen.

There are many different possibilities to choose $\Pi_{\ell+1}^{\ell}$. The simplest choice of $\Pi_{\ell+1}^{\ell}$ is

$$\Pi_{\ell+1}^{\ell} = 0. \tag{9.16}$$

A more sophisticated choice can be obtained by considering an interpolation from fine grid to coarse (that, for example preserves linear function locally). Namely

$$\Pi_{\ell+1}^{\ell} = \bar{\Pi}_{\ell+1}^{\ell} \otimes I_{c_\ell \times c_\ell} \tag{9.17}$$

with $\bar{\Pi}_{\ell+1}^{\ell}$ given in finite element methods.

9.1.4 Data-feature mapping: A^ℓ

The second new feature of MgNet is that this data-feature mapping only depends on the grid $\mathcal{T}_\ell$, and it does not depend on layers within the same grid. This amounts to a significant saving of the number of parameters. In comparison, the existing CNN, such as pre-act ResNet, can be interpreted as a network related to the case that A^ℓ is replaced by $A^{\ell,i}$, namely

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} * \sigma(f^\ell - A^{\ell,i} * u^{\ell,i-1}). \tag{9.18}$$

The underlying convolution kernels can be different on different grids and they can all be trained.

9.1.5 Feature extractors: $\sigma \circ B^{\ell,i} * \sigma$

Here we adopt the feature extractor as:

$$\sigma \circ B^{\ell,i} * \sigma. \tag{9.19}$$

Other than the level dependent extractors, the following different strategies can be used

Constant Extractors : $B^{\ell,i} = B^\ell$ for $i = 1 : \nu_\ell$
Scaled Extractors :$B^{\ell,i} = \alpha_i B^\ell$ for $i = 1 : \nu_\ell$
Variable Extractors : $B^{\ell,i}$

This brief framework gives us the basic principle on designing a CNN models for classification. All models are seen as the special choice of data-feature mapping A^ℓ, feature extractors $B^{\ell,i}$ and the pooling operators $\Pi_{\ell+1}^{\ell}$ with $R_{\ell+1}^{\ell}$.

9.2 Variants and generalizations of MgNet

The MgNet model algorithm is one very basic and it can be generalized in many different ways. It can also be used as a guidance to modify and extend many existing CNN models.

The following result show how MgNet is related to he pre-act ResNet [6].

Theorem 23. *The MgNet model Algorithm 21, admits the following identities*

$$r^{\ell,i} = f^{\ell,i-1} - A^{\ell} \circ \sigma \circ B^{\ell,i} \circ \sigma(r^{\ell,i-1}), \quad i = 1 : \nu_{\ell}, \tag{9.20}$$

where

$$r^{\ell,i} = f^{\ell} - A^{\ell} * u^{\ell,i}. \tag{9.21}$$

Furthermore, (9.20) *represents pre-act ResNet [6] as shown before.*

Proof. Because of the linearity of A^{ℓ} and invariant within the same grid ℓ, we can apply A^{ℓ} on both sides of (9.3) and minus with f^{ℓ}, thus we have

$$f^{\ell} - A^{\ell} * u^{\ell,i} = f^{\ell} - A^{\ell} * u^{\ell,i-1} - A^{\ell} * \sigma \circ B^{\ell,i} \circ \sigma(f^{\ell} - A^{\ell} * u^{\ell,i-1}).$$

This finish the proof with definition in (9.21). □

The above result is very simple but critically important. In view of Theorem 23, it shows how multigrid and CNN are intimately related. Furthermore, it provides a different version of iResNet, which can be viewed as the dual version of the original pre-act ResNet. This relation is quit similar with the dual relation of u and f in multigrid method [**?**].

Lemma 28. *The ResNet [5] step as in* (10.37) *admits the following relation:*

$$\tilde{r}^{\ell,i+1} = \sigma(\tilde{r}^{\ell,i}) - A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(\tilde{r}^{\ell,i}), \tag{9.22}$$

where

$$\tilde{r}^{\ell,i} = r^{\ell,i-1} - A^{\ell,i} * \sigma \circ B^{\ell,i} * r^{\ell,i-1}. \tag{9.23}$$

Proof. First, we apply $A^{\ell,i+1} \circ \sigma \circ B^{\ell,i+1}$ on the both sides of (10.37) and get

$$A^{\ell,i+1} * \sigma \circ B^{\ell,i+1} * r^{\ell,i} = A^{\ell,i+1} * \sigma \circ B^{\ell,i+1} * \sigma(\tilde{r}^{\ell,i}). \tag{9.24}$$

Minus by $r^{\ell,i}$ on the both sides and recall the definition in (9.23), we have

$$\tilde{r}^{\ell,i+1} = r^{\ell,i} - A^{\ell,i+1} * \sigma \circ B^{\ell,i+1} * \sigma(\tilde{r}^{\ell,i}).$$

By the definition of $r^{\ell,i} = \sigma(\tilde{r}^{\ell,i})$, we finish this proof. □

We call the above form (9.22) as σ-ResNet, similar to the MgNet we replace $A^{\ell,i}$ by A^{ℓ} and get the next Mg-ResNet form as:

$$r^{\ell,i} = \sigma(r^{\ell,i-1}) - A^{\ell} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1}). \tag{9.25}$$

Table 9.1. Comparison for all iterative forms

Primal-Dual	Model	Iterative form
Feature space	Abstract-MgNet	Solving $A^\ell(u^\ell) = f^\ell$
	General-MgNet	$u^{\ell,i} = u^{\ell,i-1} + B^{\ell,i}(f^\ell - A^\ell(u^{\ell,i-1}))$
	MgNet	$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} \circ \sigma(f^\ell - A^\ell(u^{\ell,i-1}))$
Data space	pre-act ResNet	$r^{\ell,i} = r^{\ell,i-1} - A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1})$
	Mg pre-act ResNet	$r^{\ell,i} = r^{\ell,i-1} - A^{\ell} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1})$
	Mg-ResNet	$r^{\ell,i} = \sigma(r^{\ell,i-1}) - A^{\ell} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1})$
	σ-ResNet	$r^{\ell,i} = \sigma(r^{\ell,i-1}) - A^{\ell,i} * \sigma \circ B^{\ell,i} * \sigma(r^{\ell,i-1})$
	ResNet	$r^{\ell,i} = \sigma(f^{\ell,i-1} - A^{\ell,i} * \sigma \circ B^{\ell,i} * r^{\ell,i-1})$

If we take these pooling and prolongation operators as discussed in the previous sections and focus on the iterative forms on a certain grid ℓ, we may compare them all as:

We can have these connections for all iterative scheme in data space:

(9.26)
$$\text{ResNet} \xleftrightarrow{(9.23)} \sigma\text{-ResNet} \xleftrightarrow{A^{\ell,i}\leftrightarrow A^\ell} \text{Mg-ResNet} \xleftrightarrow{\sigma(f^{\ell,i-1})\leftrightarrow f^{\ell,i-1}} \text{Mg pre-act ResNet} \xleftrightarrow{A^\ell\leftrightarrow A^{\ell,i}} \text{pre-act ResNet}.$$

In this sense, these MgNet related models can be understood as models between pre-act ResNet and ResNet. And all these models can be understood as iteration in the data space as a dual relationship with feature space as MgNet.

The rationality of replacing $A^{\ell,i}$ by layer independent A^ℓ may be justified by the following theorem.

Theorem 24. *On each grid $\mathcal{T}_\ell$,*

1. *Any CNN model with*

$$f^{\ell,i} = \chi^{\ell,i} \circ \sigma(f^{\ell,i-1}), \tag{9.27}$$

can be written as

$$f^{\ell,i} = \sigma(f^{\ell,i-1}) - \xi^\ell \circ \sigma \circ \eta^{\ell,i} \circ \sigma(f^{\ell,i-1}). \tag{9.28}$$

2. *Any CNN model with*

$$f^{\ell,i} = \sigma \circ \chi^{\ell,i}(f^{\ell,i-1}). \tag{9.29}$$

can be written as

$$f^{\ell,i} = \sigma\left(f^{\ell,i-1} - \xi^\ell \circ \sigma \circ \eta^{\ell,i}(f^{\ell,i-1})\right). \tag{9.30}$$

Proof. Let use prove the first case as an example, the second case can be proven with the same process.

With similar structure in MgNet, we can take

$$\xi^\ell = \hat{\delta}^\ell := [\hat{\delta}_1, \cdots, \hat{\delta}_{c_\ell}], \tag{9.31}$$

and

$$\eta^{\ell,i} = [\mathrm{id}_{c_\ell}, -\mathrm{id}_{c_\ell}] \circ (\chi^{\ell,i} - \mathrm{id}_{c_\ell}). \tag{9.32}$$

Here

$$\mathrm{id}_{c_\ell} : \mathbb{R}^{n_\ell \times n_\ell \times c_\ell} \mapsto \mathbb{R}^{n_\ell \times n_\ell \times c_\ell}, \tag{9.33}$$

is the identity map and

$$\hat{\delta}_k : \mathbb{R}^{n_\ell \times n_\ell \times 2c_\ell} \mapsto \mathbb{R}^{n_\ell \times n_\ell}, \tag{9.34}$$

with

$$\hat{\delta}_k([X, Y]) = -([X]_k + [Y]_k), \tag{9.35}$$

for any $X, Y \in \mathbb{R}^{n_\ell \times n_\ell \times c_\ell}$ and $[X, Y] \in \mathbb{R}^{n_\ell \times n_\ell \times 2c_\ell}$.

First, we see that $\eta^{\ell,i}$ with the above form is a convolution from $\mathbb{R}^{n_\ell \times n_\ell \times c_\ell}$ to $\mathbb{R}^{n_\ell \times n_\ell \times 2c_\ell}$. Following the identity

$$ReLU(x) + ReLU(-x) = x, \tag{9.36}$$

and the definition of ξ^ℓ i.e.

$$\xi^\ell = \hat{\delta}^\ell, \tag{9.37}$$

as a special case in MgNet. For more details, we can give a exact form of $\hat{\delta}_k$ as in (9.35) with

$$\hat{\delta}_k = [0, \cdots, 0, -\delta, \cdots 0; 0, \cdots, 0, -\delta, \cdots 0], \quad k = 1 : c_\ell, \tag{9.38}$$

where δ is the identity kernel during one channel.

At last, we have

$$\begin{aligned}
\left[\xi^\ell \circ \sigma \circ [\mathrm{id}_{c_\ell}, -\mathrm{id}_{c_\ell}](x)\right]_k &= \left[\xi^\ell \circ \sigma \circ [x, -x]\right]_k, && (9.39)\\
&= \hat{\delta}_k([\sigma(x), \sigma(-x)]), && (9.40)\\
&= -\delta([\sigma(x)]_k) - \delta([\sigma(-x)]_k), && (9.41)\\
&= -(\sigma([x]_k) + \sigma(-[x]_k)), && (9.42)\\
&= -[x]_k && (9.43)
\end{aligned}$$

Thus to say,

$$\xi^\ell \circ \sigma \circ [\mathrm{id}_{c_\ell}, -\mathrm{id}_{c_\ell}] = -\mathrm{id}_{c_\ell}. \tag{9.44}$$

Then the modified dual form of MgNet in (9.22) becomes

$$f^{\ell,i} = \sigma(f^{\ell,i-1}) - \xi^{\ell,i} \circ \sigma \circ \eta^{\ell,i} \circ \sigma(f^{\ell,i-1}), \tag{9.45}$$

$$= \sigma(f^{\ell,i-1}) - \left(\xi^\ell \circ \sigma \circ [\mathrm{id}_{c_\ell}, -\mathrm{id}_{c_\ell}]\right) \circ (\chi^{\ell,i} - \mathrm{id}_{c_\ell}) \circ \sigma(f^{\ell,i-1}) \tag{9.46}$$

$$= \sigma(f^{\ell,i-1}) + (\chi^{\ell,i} - \mathrm{id}_{c_\ell}) \circ \sigma(f^{\ell,i-1}), \tag{9.47}$$

$$= \chi^{\ell,i} \circ \sigma(f^{\ell,i-1}). \tag{9.48}$$

This covers (9.28). □

Remark 11. Theorems 24 shows that general CNN in the forms of either (9.27) or (9.29) can be written recast as (9.28) or (9.30) with the data-feature mapping $A^\ell = \xi^\ell$ that is not only independent of the layers, but is actually given a priori as in (9.31). In view of Theorems 23 and 28, the classic CNN models can be essentially recovered from MgNet by choosing ξ^ℓ a priori as in (9.31). Since the classic CNN models have been extensively tested to be successful, the more general MgNet with more general ξ^ℓ (to be trained) are expected to be more efficient than the classic CNN models.