Jan S Hesthaven and Tim Warburton

Nodal Discontinuous Galerkin Methods:

Algorithms, Analysis, and Applications

– LIST OF CORRECTIONS AND CLARIFICATIONS

September 6, 2010

Springer

List of corrections and clarifications

Notation: First number is page number, second number line number with a "+" counting from the top and a "-" counting from the bottom. Line counts include equations.

Thanks to C. Bahls, A. Panizza, X. Zhu, C. Rohde, S. Field, J. Thorenson, J. Li, A.Engsig-Karup, A. Jameson, A. Kloeckner, M. Rosing for pointing out these misprints.

12, 5 : "noncoersive" should be "noncoercive"

45, -12 Should be dr and not dx in integral, i.e.,

$$\int_{-1}^{1} uv\, dr$$

50, Eq.(3.3) : r_i should be ξ_i.

50, -5 : Last expression should be

$$\mathcal{V}_{ij} = \tilde{P}_{j-1}(r_i).$$

56, 1 : To avoid confusion, $\ell_i(r)$, should be $\ell(r)$.
58, -1 : Matrix FToV should be

$$\text{FToV} = \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&1 \\ 0&0&0&0&1 \\ 0&0&0&1&0 \end{bmatrix}.$$

59, +3 : This also impacts $(\text{FToV})(\text{FToV})^T$ which becomes

$$(\mathrm{FToV})(\mathrm{FToV})^T = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&0&0&1&1&0&0&0\\ 0&0&0&1&1&0&0&0\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$

This also implies that [59, +5] should read *rows 1 and 8 indicate*.

67, 11 : MeshGen1D should be called with argument $(0.0, 2 * pi, 10)$.

67, -2 : $x \in [-2, 2]$ should be $x \in [-1, 1]$.

68, +2 : $E(-2, 0) = E(2, 0) = 0$ should be $E(-1, t) = E(1, t) = 0$.

68, +15 : Should be

$$= \frac{1}{J^k \mu^k} \cdots$$

70,-7 : Software line

```
[Nv, VX, K, EToV] = MeshGen1D(-2.0, 2.0, 80);
```

should be

```
[Nv, VX, K, EToV] = MeshGen1D(-1.0, 1.0, 80);
```

71, -20 : Wrong exact solution. Should be

$$H^{(m)}(x,t) = n^{(m)} \left[A^{(m)} \exp(i\omega n^{(m)} x) + B^{(m)} \exp(-i\omega n^{(m)} x) \right] \exp(i\omega t)$$

71, -18 : Wrong exact solution. Should be

$$B^{(2)} = \exp(i2n^{(2)}\omega) A^{(2)}$$

71, -16 : Wrong exact solution. Should be

$$A^{(2)} = \exp(-i\omega(n^{(1)} + n^{(2)}))$$

79, +13 : The expression for $|\tilde{v}_n|$ is correct but it should be clarified by write it as. "We recover

$$|\tilde{v}_n| \leq \left(\frac{1}{n(n+1)} \right)^p \left| \int_{\mathsf{I}} (\mathcal{L} v)^p \tilde{P}_n(r)\, dr \right|,$$

by integration by parts $2p$ times. Recalling (4.2) and combining this with ..."

85, +17 : $a(u_x, v)_\Omega$ should be $a(u_x, \phi)_\Omega$

87, +3 : $\|\varepsilon_N\|^2_{\Omega,h}$ should be $\|\varepsilon_h\|^2_{\Omega,h}$

89, +16 : Should be

$$\left[-\frac{i\omega h}{2}\mathcal{M} + a\mathcal{S}\ldots\right.$$

109, last equation : Should be $\nu \to 0$ in lim.

118, Sec 5.2 : Clarification: In this section, f_h^k has the meaning of the interpolation of the projection of f, i.e.,

$$f_h^k(x,t) = \sum_{i=1}^{N_p} (\mathcal{P}_N f)(x_i,t)\ell_i^k(x).$$

This is equivalent to assuming that all integrals are exact.

119, -8 : "from" should be "form".

122, +5 : Eq.(5.8) should be

$$\frac{1}{2}\frac{d}{dt}\|u_h\|^2_{\Omega,h} \le 0.$$

123, -8 : Last expression should be

$$f_h^k(x,t) = \sum_{i=1}^{N_p} f_h^k(x_i^k,t)\ell_i^k(x),$$

124, +3 : $u_h^k(x_i,t)$ should be $u_h^k(x_i^k,t)$.

130, +10 : Eq.(5.16), lower limit should be $\eta_c < \eta \le 1$.

134, -3 : The statement is only true for $\hat{\mathbf{n}} = 1$. The general statement should be

$$\hat{\mathbf{n}} \cdot (au)^* = \frac{2a^+a^-}{a^+ + a^-}\left(\{\{u\}\} + \frac{1}{2}[\![u]\!]\right)$$

151, -9 : Should be

```
ul = uh; ul(3:Np,:) =0; ul = V*ul;
```

162,-1 to 163, +2 : Statements

```
drho  (mapI) = ...
drhou(mapI) = ...
dEner(mapI) = ...
```

should be

```
drhof  (mapI) = ...
drhouf(mapI) = ...
dEnerf(mapI) = ...
```

163, +6-8 : Statements

```
drho  (map0) = ...
drhou(map0) = ...
dEner(map0) = ...
```

should be

```
drhof  (map0) = ...
drhouf(map0) = ...
dEnerf(map0) = ...
```

165, +7-8 : The purpose of this piece of code is to describe initial conditions at the cell-centers as is required for discontinuous initial conditions. The correct statement should be

```
cx = ones(Np,1)*(sum(MassMatrix*x,1))/2;
```

If the initial conditions are smooth, `cx`, can be replaced by `x` in the codes in lines 9-11.

175, +20ff : Should be "Thus, $w(r)$ is an N-th-order polynomial approximation to a function which measures the difference between the equidistant points and the Legendre-Gauss-Lobatto points; ..."

179, -2 : Sixth number should be "0.9808" - not "0.9800".

182, +5 : Middle expression should be

$$\ldots \mathcal{V}^T \boldsymbol{l}(\boldsymbol{r}) = \boldsymbol{\psi}(\boldsymbol{r}) \ldots$$

185, -2 : Should be

$$\mathcal{S}_r = \mathcal{M}\mathcal{D}_r, \;\; \mathcal{S}_s = \mathcal{M}\mathcal{D}_s.$$

194, +9 : "each of length $3KN_{f_p}$.

195, +1 : Remove one "find".

206, -4 : Should be "consider a diatomic gas with ..."

208, 2 : Stabilization term (last term) should be

$$\ldots + \frac{\lambda}{2}(\mathbf{q}_h^- - \mathbf{q}_h^+).$$

209, -8 : Should be

$$\ldots \frac{x - x_0 - t}{2\pi},$$

245, +9 : "discretize" instead of "discretizing"

245, +19 : "types" instead of "type"

246, +9 : Should be "As in Section 5.3, we have ..."

259, -3 : Exact solution should be

$$u(x,t) = \cos(-\pi^3 t + \pi x).$$

268, +4 : Should be

$$\ldots = hq_h^k.$$

268, +8 : Equation (7.4)

$$\ldots + \tau \frac{u_h^{k+1} - 2u_h^k + u_h^{k-1}}{h} = \ldots.$$

268, -5 : Equation (7.5) – same as above in (7.4).

271, +3 : In caption – n should be N.

282, -7/-8 : Should be $\partial\Omega^D$ and $\partial\Omega^N$ to reflect the boundaries.

282, -2 : p^O should be u^D.

287, -3 : Should be

$$\ldots = (f, \phi)_\Omega, \ldots$$

287, 13 : Statement 48 in file should be `A\(M*(-f(:)) + ubc);`

290, -3 Should be ".. with $\boldsymbol{\pi} = \nabla u$ we "

294, -8ff : Until p295, line 4, u_h should be exchanged with $\mathcal{P}_N u$ to avoid the confusion that u_h is used for a numerical solution elsewhere. In these few lines, u_h is the projection of the solution and these results are basic approximation results.

294, -7 : Last term should be

$$\|h^{-1/2}(u - u_h)\|_{\Gamma_b}^2$$

295, +14 : Remove one $|u|_{\Omega,\sigma,h}$.

298, -10 : `rhs` should be `f`, i.e., `>> f = f(P)`.

312, -12 : "compare" instead of "compared"

353, -19 : Remove "which" to have "spectrum and appears to be isolated."

373, +6 : "complex situations."

376, -14 : Software statement, line 16. Should be

```
theta1 = atan2(VY(v1)-yo, VX(v1)-xo);
theta2 = atan2(VY(v2)-yo, VX(v2)-xo);
```

399, -1 : "information"

413, -7 : Should be

$$\ldots \boldsymbol{t}^{3,2} = \left(-\sqrt{\frac{1}{12}}, -\frac{1}{6}, \sqrt{\frac{8}{9}} \right),$$

434, -4ff : The 6 equations are written with a wrong sign on the boundary flux terms. This is due to a definition of $[q] = q^+ - q^-$ which is inconsistent with the bottom of p.433.
There should be a '-' in front of all boundary terms as a consequence of this.

435-436 : This same inconsistency is reflected in the code which should be

```
% form field differences at faces
dHx(:)  = Hx(vmapM)-Hx(vmapP);  dEx(:)  = Ex(vmapM)-Ex(vmapP);
dHy(:)  = Hy(vmapM)-Hy(vmapP);  dEy(:)  = Ey(vmapM)-Ey(vmapP);
dHz(:)  = Hz(vmapM)-Hz(vmapP);  dEz(:)  = Ez(vmapM)-Ez(vmapP);
```

and

```
fluxHx =  ny.*dEz - nz.*dEy - alpha*(dHx - ndotdH.*nx);
fluxHy =  nz.*dEx - nx.*dEz - alpha*(dHy - ndotdH.*ny);
fluxHz =  nx.*dEy - ny.*dEx - alpha*(dHz - ndotdH.*nz);

fluxEx =  -ny.*dHz + nz.*dHy - alpha*(dEx - ndotdE.*nx);
fluxEy =  -nz.*dHx + nx.*dHz - alpha*(dEy - ndotdE.*ny);
fluxEz =  -nx.*dHy + ny.*dHx - alpha*(dEz - ndotdE.*nz);
```

447, -9 : Software statement

```
x(1) = (alpha-beta)/(alpha+beta+2);
```

should be

```
x(1) = -(alpha-beta)/(alpha+beta+2);
```

420, -2 : Should be

$$\mathcal{S}_r = \mathcal{M}\mathcal{D}_r, \;\; \mathcal{S}_s = \mathcal{M}\mathcal{D}_s, \;\; \mathcal{S}_t = \mathcal{M}\mathcal{D}_t.$$

449, -1 : $P_k^{(2i+2j+2,0)}(b)$ should be $P_k^{(2i+2j+2,0)}(c)$

450, 2 : Second expression should be

$$\ldots, b = 2\frac{1+s}{1-t} - 1, \ldots$$

459 : CorrectBCTable.m fails if an element has two faces on the boundary. An improved version but with a different calling sequence is

```
function BCType = CorrectBCTable_v2(EToV,VX,VY,BCType,fd,BCcode)

% function BCType = CorrectBCTable(EToV,BCType,fd,BCcode);
% Purpose: Setup BCType for boundary conditions in 2D
%
%    EToV   : Element-To-Vertice table
%    VX, VY : (x,y)-coordinates of mesh vertices
%    BCType : Table with types of faces for BC's
%    fd     : handle to distance function
%    BCcode : Integer for specific boundary type
%
% By Allan P. Engsig-Karup

Globals2D;

VNUM = [1 2;2 3;3 1]; % face orientations

pxc = 0.5*(VX(EToV)+VX(EToV(:,[2 3 1])));
pyc = 0.5*(VY(EToV)+VY(EToV(:,[2 3 1])));
dc  = abs(fd([pxc(:) pyc(:)])); % distances to boundaries from face centers
tol = 1e-4; % tolerance
idx = find(dc<tol);
BCType(idx) = BCcode;
return
```

464, +11 : Should be "interior values, $u^- = u$(vmapM)."

464, +13 : Should be "exterior values, $u^+ = u$(vmapP)."

464, +14 : vmapB is a one-vector but the size is problem dependent.

467, +12 : Software line

```
[TRI,xout,yout,uout,interp] PlotField2D(2*N, x, y, vort);
```

should be

```
[TRI,xout,yout,uout,interp] = PlotField2D(2*N, x, y, vort);
```

467, -3 : Software line

```
PlotContour2D(TRI, xout, yout, vortout, linspace(-6, 6, 12));
```

should be

```
PlotContour2D(TRI, xout, yout, vort, linspace(-6, 6, 12));
```

481, ref 134 : "applications"