Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems

D. Tremblay*, S. Étienne†, D. Pelletier‡

École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec, H3C 3A7, Canada

This paper presents the Method of Manufactured Solutions (MMS) for fluid-structure interactions code verification. The MMS provides benchmark solutions for direct evaluation of the solution error. The best benchmarks are exact analytical solutions with sufficiently complex solution structure to ensure that all terms of the differential equations are exercised in the simulation. The MMS provides a straight forward and general procedure for generating such solutions. When used with systematic grid refinement studies, which are remarkably sensitive, the MMS provides strong code verification with a theorem-like quality. The MMS is first presented on simple 1-D examples. Manufactured solutions for fluid-structure interaction (FSI) problems are then presented with sample results from grid convergence studies.

I. Introduction

In the subject of *Quantification of Uncertainty* in numerical simulations, the three most important items are: *Verification of Codes*, *Verification of Calculations* and *Validation*.[1] For logical and practical reasons these activities are performed in this order. Verification of a code involves error *evaluation* from a known solution to establish that the numerical code works correctly. Verification of a calculation involves error *estimation* to make sure that the code delivers the expected accuracy on a specific application problem. Both verifications are purely numerical exercises with no concern whatever for the accuracy of the physical laws used in the code. This the concern of *Validation*, i.e. the agreement of the mathematical model with the physical system of interest. In other words, Verification is concerned with solving the equations *right* while Validation focuses on solving the *right* equations.

Journal policity statements usually refer only to Verification of Calculations;[2] the assumption is made that the code is correct. Determining the correctness of a code can only be done by systematic grid convergence tests on a problem with a *benchmark* solution. The best standard of comparison is an exact analytical solution expressed in terms of simple mathematical function such as **sin**, **exp**, **tanh**, etc. Infinite series are not desirable as they tend to be more trouble to evaluate accurately than the numerical solution itself. The benchmark solution should not only be exact, it should also exhibit a complex enough structure to ensure that all terms in the governing equations are exercised by the test.

It has often been stated in scientific journals that general accuracy verification of codes for difficult problems (i.e. Navier-Stokes equations) is not possible because exact solutions exist only for simple problems that do not fully exercise the code. In fact, a very general procedure exists for generating analytical solution for verification purposes. It was originally presented in.[3] With a few exceptions,[4,5] acceptance of the method has been slow. Misunderstanding of the method is a common occurrence. Based on our experience and many discussions with collegues, this misunderstanding is due to the deceptive simplicity of the concept. This paper is an attempt to clarify the concepts through simple examples and to provide recent references.

II. The method of manufactured solutions

The Method of Manufactured Solution (MMS) provides a general procedure for generating analytical solutions for code accuracy verification. The procedure is very simple (some will say deceptively so!). We

*Currently Graduate Student at McGill University, Department of Chemical Engineering

†Research Fellow, Département de Génie Mécanique

‡Canada Research Chair, Département de Génie Mécanique, Associate Fellow AIAA

first pick a continuum solution. In general this solution will not satisfy the governing equations because of the arbitrary nature of our choice. An appropriate source term is defined to cancel out any imbalance in the PDE caused by our choice of the continuum solution. Interestingly enough, this choice can often be made independently of the code or of the equations considered. That is, we can pick a solution and use it to verify an incompressible Navier-Stokes code, a Darcy flow model, a heat equation, a materials code, etc.

The solution should be non-trivial in the sense that it exercises all derivatives in the PDE. The solution also defines the boundary conditions in all forms be they Dirichlet, Neumann or Robin. We first illustrate this on sample examples.

A. A first example of the MMS

To emphasize the generality of the concept, we pick the solution *before we specify the governing equations*:

$$U(t,x) = A + \sin(x + Ct) \tag{1}$$

We apply this 1-D transient solution to the nonlinear Burger's equation which we write as a nonlinear operator of u

$$\mathcal{L}_1(u) = u_t + uu_x - \alpha u_{xx} = 0 \tag{2}$$

We determine the source term Q_1 that produces the solution U by applying the operator $\mathcal{L}_1$ to U

$$Q_1 = \mathcal{L}_1(U) = U_t + UU_x - \alpha U_{xx} \tag{3}$$

which yields

$$\begin{aligned} Q_1 &= C\cos(x + Ct) && (4)\\ &\quad +[A + \sin(x + Ct)]\cos(x + Ct) && (5)\\ &\quad +\alpha \sin(x + Ct) && (6) \end{aligned}$$

If we now solve the modified equation

$$\mathcal{L}_1(u) = u_t + uu_x - \alpha u_{xx} = Q_1(t,x) \tag{7}$$

with compatible initial and boundary conditions, the solution will be $U(t,x)$ given by equation (1). The initial condition is obviously $u(0,x) = U(0,x)$ while the boundary conditions are determined from equation (1). Note that the computational domain has not yet been specified. We could consider the usual domain $0 \le x \le 1$ or something like $-10 \le x \le 100$. In either case the same solution (1) applies. However, boundary conditions differ because they are evaluated at different x locations. Moreover, the same solution $U(t,x)$ can be produced by solving (7) with more than one set of boundary conditions. For example, on the interval $[0,1]$ the following inflow and boundary conditions will yield $U(t,x)$ as a solution of (7)

Dirichlet-Dirichlet

$$\begin{aligned} u(t,0) &= A + \sin(Ct)\\ u(t,1) &= A + \sin(1 + Ct) \end{aligned}$$

Dirichlet-Neumann

$$\begin{aligned} u(t,0) &= A + \sin(Ct)\\ u_x(t,1) &= \cos(1 + Ct) \end{aligned}$$

Robin-Neumann

$$\begin{aligned} au + bu_x|_{(t,0)} &= d\\ u_x(t,1) &= \cos(1 + Ct) \end{aligned}$$

With a and b given, simply select $d = a[A + \sin(Ct)] + b\cos(Ct)$.

B. A second example

To further clarify the concepts, we now apply the above manufactured solution to a different differential equation, that *might* be a candidate for a 1-D turbulence mixing length formulation. The Burger like differential operator is now

$$\begin{aligned} \mathcal{L}_2(u) &= u_t + uu_x - \alpha u_{xx} && (8)\\ &\quad -2\lambda[x(u_x)^2 + x^2 u_{xx}] = 0 && (9)\end{aligned}$$

We determine Q_2 that generates U by applying $\mathcal{L}_2$ to U.

$$\begin{aligned} Q_2 = \mathcal{L}_2(U) &= U_t + UU_x - \alpha U_{xx} && (10)\\ &\quad -2\lambda[x(U_x)^2 + x^2 U_{xx}] && (11)\end{aligned}$$

which yields:

$$\begin{aligned} Q_2 &= C\cos(x+Ct) && (12)\\ &\quad +[A+\sin(x+Ct)]\cos(x+Ct) && (13)\\ &\quad +\alpha\sin(x+Ct) && (14)\\ &\quad -\lambda[x\cos^2(x+Ct) + x^2\sin(x+Ct)] && (15)\\ & && (16)\end{aligned}$$

If we now solve the following modified equation

$$\begin{aligned} \mathcal{L}_2(u) &= u_t + uu_x - \alpha u_{xx} && (17)\\ &\quad -2\lambda[x(u_x)^2 + x^2 u_{xx}] = Q_2 && (18)\\ & && (19)\end{aligned}$$

with compatible initial and boundary conditions, the solution for this *turbulent* flow model will be $U(t,x)$ given by equation (1), as it was for the previous laminar Burger equation.

C. Application to verification

Such non-trivial analytic solutions can be used to Verify a Code by performing systematic grid convergence studies. This is based on the behavior of the error E as the mesh size h is reduced:

$$E = f_h - f_{ex} = h^p + H.O.T. \tag{20}$$

where f_h is the discrete solution, f_{ex} the exact solution, h a measure of the discretization, and p the convergence rate of the numerical scheme. This behavior applies to every consistent methodology (FDM,FVM, FEM etc.). This idea is to monitor the behavior of E as the grid is refined. Grid doubling is not necessary, just refinement. However, thorough iterative convergence is required. Theoretically, values of $C = E/h^p$ should become constant as the grid is refined.

The procedure will detect all ordered errors (interior discretization, boundary condition discretization etc.). It will not evaluate the adequacy of non ordered approximation such as the distance to an outflow boundary or $\frac{\partial p}{\partial n} = 0$ at a wall. The errors of such approximations do no vanish as $h \to 0$, hence they are *non-ordered approximation*. However, if the code uses a second order approximation of $\frac{\partial p}{\partial n} = 0$ at a wall and the MMS procedure shows that it is indeed second order accurate, then the code is verified on this point. However, the method will not detect coding mistakes that slow down the iterative solver while leaving the answer unaffected. See Roache[1] for further discussion.

When this grid convergence test is completed satisfactorily, we have verified: any equation transformation used (body fitted grids), the order of the discretization, the coding of the discretization and the matrix solution procedure. The technique is very powerful. Users sometimes say that the method is too sensitive, revealing minor inconsistencies in the special treatment of a single grid point that may corrupt the convergence rate of the method everywhere.[1] The algebraic complexity may be something of a challenge. However, symbolic manipulation can easily deal with it.

Code verification guarantees that the finite element implementation is correct. However, when applying a verified code to a practical case, one must also perform grid refinement studies to ensure that the code is used correctly. Error estimates may be computed using classical Richardson Extrapolation[1] or any unstructured mesh error estimator.[6] Results from such grid refinement studies are best reported using error bands of the Grid Convergence Index (GCI) for structured meshes[1] or upper and lower bounds of the GCI for unstructured grids.[5]

III. MMS for FSI problems

We present a generic methodology to get manufactured solutions for FSI problems.

A. Equations involved in the verification process

The FSI problem is described by flow, solid and interface equations. It is completed with suitable boundary conditions on all boundaries and fluid-solid interface. Notations and geometry are depicted on figure 1.

The flow is described by the steady Navier-Stokes equations (21, 22) of an incompressible and Newtonian fluid.

$$\text{Continuity:} \qquad \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \quad \text{on} \quad \Omega_1^f \tag{21}$$

$$\text{Momentum:} \qquad \rho_f \mathbf{u}_f \cdot \boldsymbol{\nabla} \mathbf{u}_f = \boldsymbol{\nabla} \overline{\overline{\boldsymbol{\sigma}}}_f + \rho_f \mathbf{f}_f \quad \text{with} \quad \overline{\overline{\boldsymbol{\sigma}}}_f = \mu_f \left[\boldsymbol{\nabla} \mathbf{u}_f + (\boldsymbol{\nabla} \mathbf{u}_f)^T \right] - p_f \overline{\overline{\mathbf{I}}} \quad \text{on} \quad \Omega_1^f \tag{22}$$

These equations are supplemented supplemented with the following boundary conditions:

$$\overline{\overline{\boldsymbol{\sigma}}}_f \cdot \mathbf{n}_f = \overline{\mathbf{t}}_f \quad \text{on} \quad \Gamma_3^f \quad \text{(Neumann-type)} \tag{23}$$

$$\mathbf{u}_f = \overline{\mathbf{u}}_f \quad \text{on} \quad \Gamma_1^f \text{ and } \Gamma_2^f \quad \text{(Dirichlet-type)} \tag{24}$$

$$\mathbf{u}_f = 0 \quad \text{on} \quad \Gamma_{FS} \tag{25}$$

We consider hyperelastic solids undergoing large displacements. The solid is assumed isotropic and described by a St.Venant-Kirchhoff material. In the total Lagrangian approach, differential equations equilibrium are expressed on the initial undeformed configuration:[7]

$$\boldsymbol{\nabla} \cdot \overline{\overline{\boldsymbol{\sigma}}}_l + \mathbf{f}_s = 0 \quad \text{on} \quad \Omega_0^s \tag{26}$$

supplemented with the following boundary conditions on undeformed configuration (underscript 0):

$$\boldsymbol{\chi}_s = \overline{\boldsymbol{\chi}}_s \quad \text{on} \quad \Gamma_{0,1}^s,\ \Gamma_{0,2}^s \text{ and } \Gamma_{0,3}^s \quad \text{(Dirichlet-type)} \tag{27}$$

Finally, interface equilibrium is enforced through the application of normal stress continuity between the fluid and the solid at the interface location. Details on how we match the eulerian flow and total lagrangian structural interface equilibrium are found in Etienne *et al.*:[7]

$$\overline{\overline{\boldsymbol{\sigma}}}_c \cdot \mathbf{n}_s + \overline{\overline{\boldsymbol{\sigma}}}_f \cdot \mathbf{n}_f = 0 \quad \text{on} \quad \Gamma_{FS} \quad \text{(Neumann-type)} \tag{28}$$

where $\overline{\overline{\boldsymbol{\sigma}}}_c$ and $\overline{\overline{\boldsymbol{\sigma}}}_f$ are the Cauchy solid and fluid stress tensors respectively.

B. Description of the procedure for generating MMS with an example

We will generate a manufactured solution (pressure, viscosity and velocity field distributions) which satisfies the continuity equation on the deformed fluid domain, continuity of tractions and displacements at the fluid-solid interface. In general, these fields will not satisfy the momentum equations for the fluid and the stress equilibrium for the solid. Source terms, $\mathbf{f}_f$ and $\mathbf{f}_s$ are added to the momentum (22) and solid stress equations (26) to ensure equilibrium.[1] However, the continuity equation (21) does not have this feature. Thus, we are constrained to choose the velocity distribution to satisfy the continuity. This is a challenging problem for complex geometry domains.

The procedure starts with the definition of a deformed fluid domain. We consider an axisymmetric coordinate system. One of the boundaries corresponds with the x axis. Moreover, to avoid obtaining too

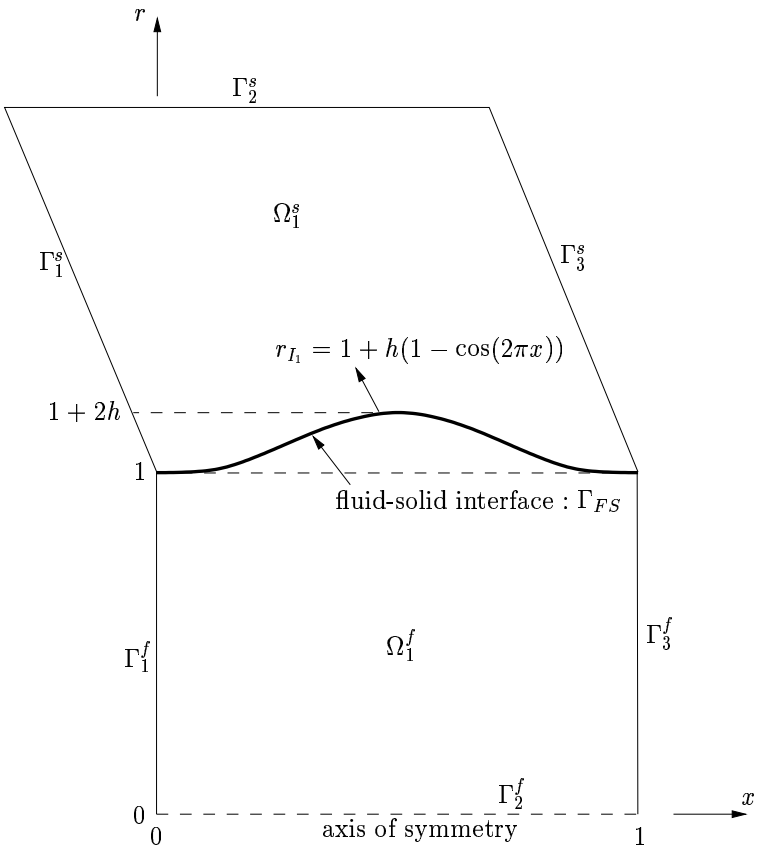


Figure 1. Fluid domain in the deformed configuration.

simple a solution, like the Poiseuille type flow, we have chosen a non trivial fluid domain geometry as illustrated on figure 1.

We now proceed with an example to describe the methodology of manufactured solutions for Fluid-Structure Interactions. The shape of the interface in the deformed configuration is described by the following equation:

$$r_{I_1}(x) = 1 + h[1 - \cos(2\pi x)] \tag{29}$$

We first need to find a divergence free velocity field on this domain. All the details of the procedure we have developed are given in appendix. We apply it and pick the simplest functional , $K[f(x)] = 1$ and the power $k = 1$, both appearing in equation (51). These two parameters allow to build a rich and wide variety of velocity field shapes, then giving us all the freedom we need. The process goes on by simple deduction of following functionals appearing in equation (52):

$$M[f(x)] = \int K[f(x)]f'(x)dx = 1 + h[1 - \cos(2\pi x)] \tag{30}$$

$$L[f(x)] = \int f(x)K[f(x)]f'(x)dx = \frac{\{1 + h[1 - \cos(2\pi x)]\}^2}{2} \tag{31}$$

which means, $M[r] = r$ and $L[r] = r^2/2$. Finally, application of equations (58) and (59) leads to the following velocity field:

$$\begin{cases} u(x,r) = 3r\{1 + h[1 - \cos(2\pi x)]\} - \{1 + h[1 - \cos(2\pi x)]\}^2 - 2r^2 \\ v(x,r) = 2\pi r\{1 + h[1 - \cos(2\pi x)] - r\}\sin(2\pi x) \end{cases} \tag{32}$$

which is illustrated on figure 2 using a value of $h = 0.03$. The pressure and viscosity distributions will be developed later. They appear only in the momentum equation (22). The source term, $\mathbf{f}_f$, provides the

freedom to choose any expression for $\mu(x,r)$ and $p(x,r)$ in the fluid domain. This flexibility will prove to be essential later to satisfy equation (28). The flow boundary conditions are obtained from this manufactured solution.

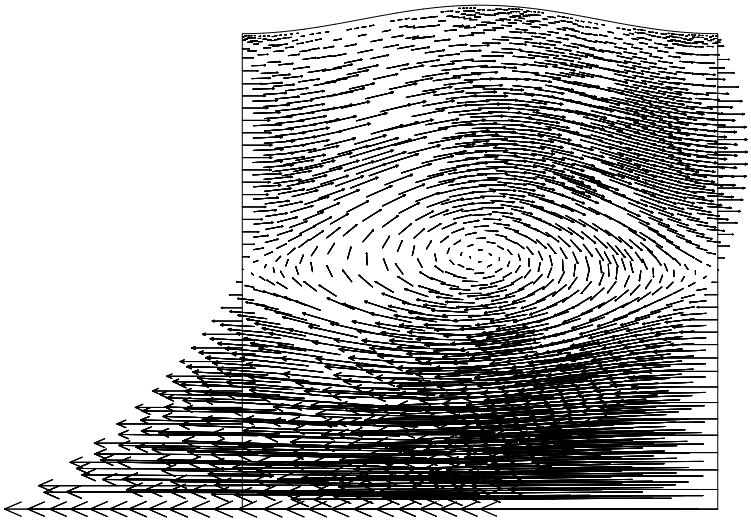


Figure 2. Divergence free flow.

Secondly, as the interface location of the solid and fluid domain must coincide, we pick a displacement structural field which matches the displacements at the interface location (29). As this field won't satisfy equilibrium equation (26), we adjust the source term $\mathbf{f}_s$. As there is no other constraint on the structural displacement field, this is much easier than generating the fluid velocity field. However, having considered incompressible material, a similar procedure as that developed in appendix would have been required for the manufactured structural displacement field. In our case, we've simply picked the following displacement field, keeping $h = 0.03$ for compatibility.:

$$\xi = \frac{1}{4}(1-r) \tag{33}$$

$$\eta = (2-r)h[1-\cos(2\pi x)] \tag{34}$$

So far we have a divergence free velocity field and a displacement field which satisfy equations (21), (22) and (26). The last step consists in guarantying continuity of the fluid and solid forces at the fluid-solid interface. This will be possible by adjusting the pressure and viscosity distribution as functions of space. Since $\mathbf{n}_1^f = -\mathbf{n}_1^s$, we will drop indices of normals in equation (28) to reach:

$$\overline{\overline{\boldsymbol{\sigma}}}_c \cdot \mathbf{n} = \overline{\overline{\boldsymbol{\sigma}}}_f \cdot \mathbf{n} \tag{35}$$

Developing equation (35) in terms of x and r components gives

$$\sigma_{c_{xx}} n_x + \sigma_{c_{xr}} n_r = \sigma_{f_{xx}} n_x + \sigma_{f_{xr}} n_r \tag{36}$$

$$\sigma_{c_{rx}} n_x + \sigma_{c_{rr}} n_r = \sigma_{f_{rx}} n_x + \sigma_{f_{rr}} n_r \tag{37}$$

Then, setting $S_x = \sigma_{c_{xx}} n_x + \sigma_{c_{xr}} n_r$, $S_r = \sigma_{c_{rx}} n_x + \sigma_{c_{rr}} n_r$, equations (36, 37) become

$$S_x = \left(2\mu_f \frac{\partial u_f}{\partial x} - p_f\right) n_x + \mu_f \left(\frac{\partial u_f}{\partial r} + \frac{\partial v_f}{\partial x}\right) n_r \tag{38}$$

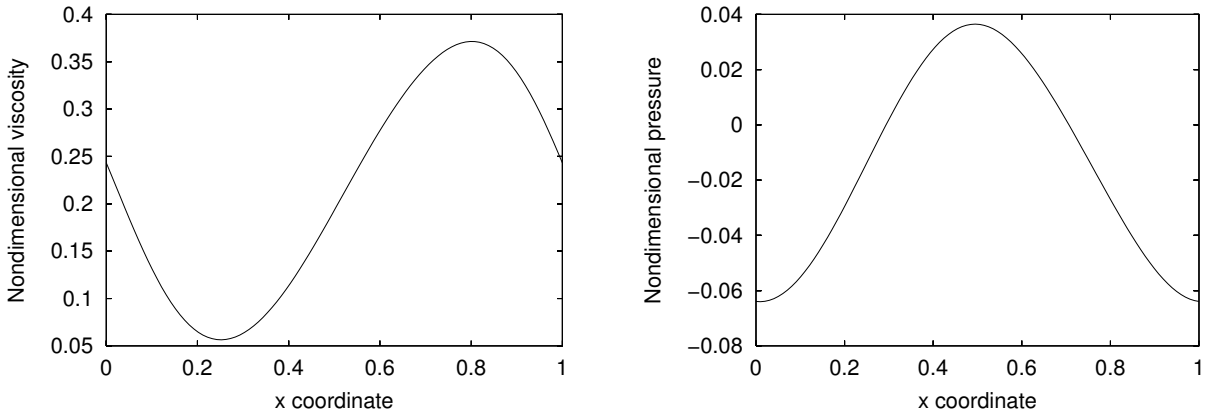
$$S_r = \mu_f \left(\frac{\partial u_f}{\partial r} + \frac{\partial v_f}{\partial x}\right) n_x + \left(2\mu_f \frac{\partial v_f}{\partial r} - p_f\right) n_r \tag{39}$$

Solving for pressure and viscosity leads to the following pressure and viscosity interface distributions,

$$\mu_f = \frac{S_x n_r - S_r n_x}{-\mathbf{B} n_x^2 - \mathbf{C} n_x n_r + \mathbf{A} n_x n_r + \mathbf{B} n_r^2} \tag{40}$$

$$p_f = \frac{-\mathbf{A}S_r n_x - \mathbf{B}S_r n_r + \mathbf{B}S_x n_x + \mathbf{C}S_x n_r}{-\mathbf{B}n_x^2 - \mathbf{C}n_x n_r + \mathbf{A}n_x n_r + \mathbf{B}n_r^2} \tag{41}$$

where $\mathbf{A} = 2\partial u/\partial x$, $\mathbf{B} = (\partial u/\partial r + \partial v/\partial x)$ and $\mathbf{C} = 2\partial v/\partial r$.



Viscosity distribution Pressure distribution

Figure 3. Viscosity and pressure distributions at the interface.

Thus, pressure and viscosity distributions can easily be obtained using the fluid velocity and displacement fields and the expression of the normal of the deformed interface. These distributions are shown on figure 3 for $E = 2$ and $\nu = 0.1$. We simply apply these distributions in the fluid domain for all r. The procedure outlined above can yield negative values of the viscosity, which is not wished. In fact, this happened with our first choice of the displacement field. This was easily resolved by applying an additional displacement gradient in the x direction to the solid. This has the effect of offsetting the resulting viscosity distribution by a constant to ensure its positive in the whole fluid domain. We are now in position to apply appropriate structural displacement boundary conditions.

At that step, the problem is finalized. Fluid velocity field, pressure and viscosity distributions as well as the solid displacement field have been defined and satisfy to equations (21), (22), (26) and (28) and boundary conditions.

For information, momentum body force terms $\mathbf{f}_f$ are

$$\begin{aligned} f_x^f &= \rho_f \left(u_f \frac{\partial u_f}{\partial x} + v_f \frac{\partial u_f}{\partial r} \right) + \frac{\partial p_f}{\partial x} \\ &- \mu \left(\frac{\partial^2 u_f}{\partial x^2} + \frac{1}{r}\frac{\partial u_f}{\partial r} + \frac{\partial^2 u_f}{\partial r^2} \right) - 2\frac{\partial \mu_f}{\partial x}\frac{\partial u_f}{\partial x} - \frac{\partial \mu_f}{\partial r}\left(\frac{\partial u_f}{\partial r} + \frac{\partial v_f}{\partial x} \right) \end{aligned} \tag{42}$$

$$\begin{aligned} f_r^f &= \rho_f \left(u_f \frac{\partial v_f}{\partial x} + v_f \frac{\partial v_f}{\partial r} \right) + \frac{\partial p_f}{\partial r} \\ &- \mu_f \left(\frac{\partial^2 v_f}{\partial x^2} + \frac{1}{r}\frac{\partial v_f}{\partial r} + \frac{\partial^2 v_f}{\partial r^2} + \frac{v_f}{r^2} \right) - \frac{\partial \mu_f}{\partial x}\left(\frac{\partial u_f}{\partial r} + \frac{\partial v_f}{\partial x} \right) - 2\frac{\partial \mu_f}{\partial r}\frac{\partial v_f}{\partial r} \end{aligned} \tag{43}$$

and structural equilibrium equation (26) source terms $\mathbf{f}_s$ are

$$f_x^s = -\left(\frac{\partial \sigma_{l_{xx}}}{\partial x} + \frac{\sigma_{l_{xr}}}{r} + \frac{\partial \sigma_{l_{xr}}}{\partial r} \right) \tag{44}$$

$$f_r^s = -\left(\frac{\partial \sigma_{l_{rx}}}{\partial x} + \frac{\partial \sigma_{l_{rr}}}{\partial r} + \frac{\sigma_{l_{rr}} - \sigma_{l_{\theta\theta}}}{r} \right) \tag{45}$$

C. Grid convergence analysis

Figure 4 presents grid convergence. An unstructured mesh generation and a mesh adaption procedure have been employed. We compare the true error and its estimate obtained by a Zhu-Zienkiewicz error estimator,[6]

evaluated in term of flow and structural energy norms. The energy norm presented on figure 4 combines these two energy norms. The L2 pressure true and estimate error norms are also depicted.

As can be seen, the true and estimated errors decrease at a slope of 2 for both energy and L2 norm, which reveals second order accuracy in space. Moreover, the estimated error converges to the true error as the mesh is refined. This indicates that the accuracy of the solution and the quantitative reliability of the estimator improve with adaptive remeshing. This is known as asymptotic exactness of the estimator. At that step the resulting FSI code is verified.

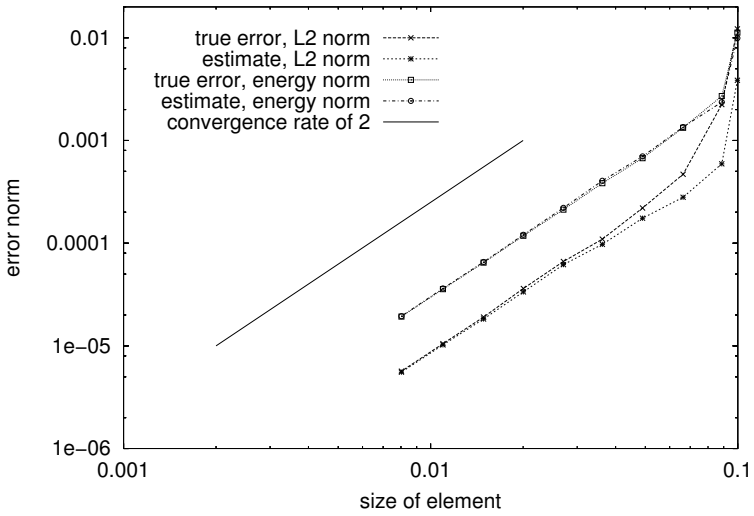


Figure 4. Grid convergence.

IV. Further uses

Salari and Knupp[8] demonstrate how sensitive the MMS can be and how useful it is in debugging codes. The authors have exercised the MMS in a blind study, in which one author modified a CFD code, developed and verified by the other, deliberately introducing errors. The code author then tested the sabotaged code with the MMS. The code used for this exercise was a full time-dependent, compressible and incompressible, Navier-Stokes solver with plenty of options for the user. In all, 21 cases were studied including one *placebo* (no mistake introduced) and several cases including something other than the solution (wrong time step, post-processing errors etc...). The exercise also highlights some of the limitations of the MMS. All order-of-accuracy mistakes errors (all that could prevent the governing equations from being correctly solved) were successfully detected. This report presents a thorough taxonomy of errors and several detailed examples.

Mixed order methods present special challenges for grid convergence studies. Roy[9] presents the resolution, in an elegant manner, to the long-standing and difficult problem of the treatment of mixed-order methods. Such schemes arise either from the explicit use of first order advection discretization and second-order diffusion, or from the first order observed convergence rate of a nominally second order method due to the use of limiter at shock points. The procedure simply involves 3 grids to evaluate the two leading terms in the error expansion. This can be done with non integer grid refinement for cost-effectiveness. The approach is applicable to verification of code, verification of calculations, the computation of the Grid Convergence Index,[1] and the treatment of the QUICK scheme and similar methods using second order diffusion and third order advection discretizations.

V. Conclusions

This paper has presented a detailed description of the Method of Manufactured Solution for FSI Code Verification and Debugging. The MMS enables one to produce many exact analytical solutions for use as benchmarks in systematic grid refinement tests. Such tests have proven to be remarkably sensitive for code

verification. The method is straightforward and, when applied to all options combinations in a code, can lead to a complete code verification.

The method was detailed on a fluid-structure interaction problem in an axisymmetric coordinate system. Manufactured Solution was presented for two-dimensional incompressible flow strongly coupled with an isotropic structure undergoing large displacements. Results illustrate the power and cost-effectiveness of the approach.

Appendix

Here we present a method to develop a divergence free flow in a two-dimensional non trivial domain for the MMS. We consider an axisymmetric tube whose external wall is defined by an univalent function $f(x)$ as illustrated on figure 5.

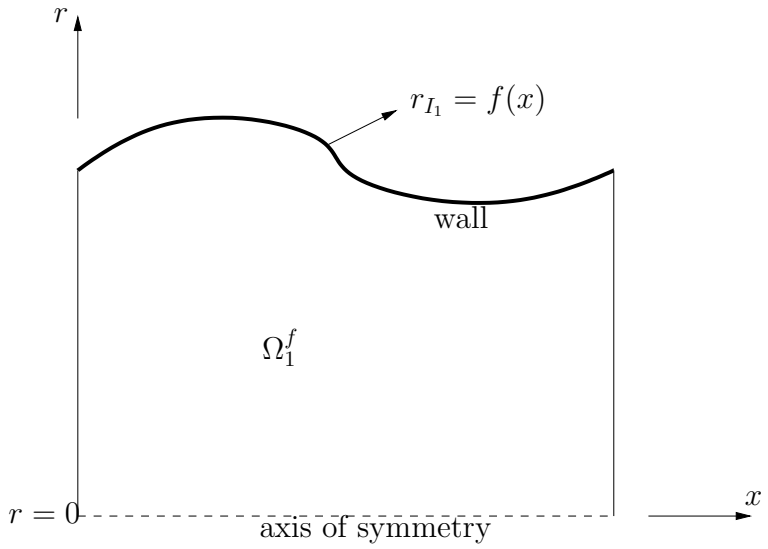


Figure 5. Arbitrary axisymmetric tube

On that domain, the fluid flow must satisfy the following conditions in order to be divergence free:

1. $u = 0$ on $r = r_{I_1}$
2. $v = 0$ on $r = r_{I_1}$
3. $v = 0$ on $r = 0$
4. $\frac{\partial u}{\partial x} + \frac{v}{r} + \frac{\partial v}{\partial r} = 0$ in Ω_1^f

As starting point we have chosen a polynomial velocity profile for the v component which satisfies conditions 2 and 3. Our choice is:

$$v(x, r) = r^k[f(x) - r] \tag{46}$$

for $k \geq 1$. For $k = 1$, the resulting u profile will be non zero on the axis of symmetry while choosing $k > 1$ will generate no-slip condition.

Multiplying $v(x, r)$ by a function of x, $G(x)$ will provide enough flexibility to satisfy the continuity equation. Thus we write:

$$v(x, r) = r^k[f(x) - r]G(x) \tag{47}$$

Substitution in the continuity equation yields:

$$\frac{\partial u}{\partial x} = (k + 2)r^k G(x) - (k + 1)r^{k-1} f(x)G(x) \tag{48}$$

which we integrate with respect to x to obtain $u(x,r)$:

$$u(x,r) = (k+2)r^k \int G(x)dx - (k+1)r^{k-1} \int f(x)G(x)dx + H(r) \quad (49)$$

The choice of the function $G(x)$ is very important. We must be able to evaluate integrals in the expression for u. We write $G(x)$ in the following form:

$$G(x) = K[f(x)]f'(x) \quad (50)$$

The velocity field u can be written:

$$\begin{aligned} u(x,r) &= (k+2)r^k \int K[f(x)]f'(x)dx \\ &- (k+1)r^{k-1} \int f(x)K[f(x)]f'(x)dx + H(r) \quad (51) \\ &= (k+2)r^k M[f(x)] - (k+1)r^{k-1}L[f(x)] + H(r) \quad (52) \end{aligned}$$

where functions $M[f(x)]$ and $L[f(x)]$ are the primitives of $\int K[f(x)]f'(x)dx$ and $\int f(x)K[f(x)]f'(x)dx$ respectively. So far, conditions 2, 3 and 4 are satisfied by the expressions of u and v. To satisfy the last conditions which corresponds to no-slip u condition, we select $H(r)$ such that $u(x, r = r_{I_1}) = 0$. Then,

$$\begin{aligned} u(x, r = r_{I_1}) &= (k+2)r_{I_1}^k M[f(x)] - (k+1)r_{I_1}^{k-1}L[f(x)] + H(r_{I_1}) \quad (53) \\ &= (k+2)f(x)^k M[f(x)] - (k+1)f(x)^{k-1}L[f(x)] + H(r_{I_1}) \quad (54) \\ &= 0 \quad (55) \end{aligned}$$

since $r_{I_1} = f(x)$. We deduce that:

$$H(r_{I_1}) = (k+1)f(x)^{k-1}L[f(x)] - (k+2)f(x)^k M[f(x)] \quad (56)$$

As $f(x) = r_{I_1}$, we have $H(r_{I_1}) = (k+1)r_{I_1}^{k-1}L[r_{I_1}] - (k+2)r_{I_1}^k M[r_{I_1}]$, and we deduce that:

$$H(r) = (k+1)r^{k-1}L[r] - (k+2)r^k M[r] \quad (57)$$

Finally, we obtain the following divergence free velocity field for an arbitrary external wall shape on which no slip apply:

$$\begin{aligned} u(x,r) &= (k+2)r^k \Big[M[f(x)] - M(r)\Big] - (k+1)r^{k-1}\Big[L[f(x)] - L(r)\Big] \quad (58) \\ v(x,r) &= r^k(f(x) - r)K[f(x)]f'(x) \quad (59) \end{aligned}$$

Acknowledgments

This work was sponsored in part by NSERC (Government of Canada), the Canada Research Chair Program (Government of Canada), and by FCAR (Government of Québec).

References

[1]P.J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa publishers, Albuquerque, NM, 1998.

[2]P.J. Roache, K. Ghia, and F. White. Editorial policy statement on the control of numerical accuracy. *ASME Journal of Fluids Engineering*, 108(1):2, March 1986.

[3]S. Steinberg and P.J. Roache. Symbolic manipulation and computational flui dynamics. *Journal of Computational Physics*, 57(2):251–284, January 1985.

[4]W. Oberkampf, F.G. Blottner, and D. P. Aeschliman. Methodology for computational fluid dynamics code verification/ validation. June 1995. AIAA Paper 95-2226, 26th AIAA Fluid Dynamics Conference, San Diego.

[5]D. Pelletier and L. Ignat. On the accuracy of the grid convergence index and the Zhu-Zienkiewicz error estimator. volume ASME FED 213, pages 31–36, June 1995. Quantification of Uncertainy in Computational Fluid Dynamics,.

[6]D. Pelletier. Adaptive computation of complex flows. *International Journal for Numerical Methods in Fluids*, 31:189–202, March 1999. Special issue of IJNMF : 10th Int'l Conf. on Finite Elements in Flow Problems.

[7]S.Etienne, D. Pelletier, and A. Garon. An updated lagrangian monolithic formulation for steady-state fluid-structure interaction problems. January 2005. AIAA Paper 2005-1086, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[8]K. Salari and P. Knupp. Code verification by the method of manufactured solution. Technical Report SAND2000-1444, Sandia National Laboratories, June 2000. ALbuquerque, NM.

[9]C. Roy. Grid convergence error analysis for mixed-order numerical schemes. June 2001. AIAA Paper 2001-2606, AIAA Computational Fluid Dynamics Conference, Anaheim.