Annual Review (in particular structure).
issues with Datastorage

1 Newton-Krylov

Second example works, has absolute value 10^{-5} and relative error 10^{-3} for $\beta = 10^{-3}$. For $\beta = 10$ we are at 10^{-6} absolute and 10^{-4} relative error as for all other problems. My suspicion was that the target is too concentrated, so we get advection dominance. I smoothed out the target to be

$$\widehat{\rho} = (1-t)0.25 + t(1/2.1445)\exp(-((x1+0.2)^2 + (x2+0.2)^2))$$

, where the exponential prefactor is reduced from 3 to 1 and the normalising constant is adjusted appropriately. This results in 10^{-6} and 10^{-4} error for $\beta = 10^{-3}$, which supports my hypothesis from above. It would be interesting to see whether it is Newton-Krylov or Fixed Point that's inaccurate by comparing to fsolve.

2 3D Convolution

The below code shows the 3D Convolution implementation.

```
function M_conv = ComputeConvolutionMatrix(this,f,saveBool)

            if(nargin(f)==1)
                    useDistance = true;
            else
                    useDistance = false;
            end

            N1  = this.N1;  N2  = this.N2;  N3  = this.N3;
            Pts = this.Pts;
            Int = this.Int;  % 1 x N1*N2

            if(useDistance)
                    fPTemp = f(GetDistance(this,Pts.y1_kv,...
                    Pts.y2_kv,Pts.y3_kv));
            else
                    fPTemp = f(Pts.y1_kv,Pts.y2_kv,Pts.y3_kv);
            end
```

```
        fDim = size(fPTemp);

        nElts = prod(fDim(2:end));

        IntT = Int.';   % N1*N2 x 1

        IntT = IntT(:,ones(1,nElts)); % N1*N2 x nElts
        IntT = reshape(IntT,fDim);     % size(f)

        M_conv = zeros([N1*N2*N3,N1*N2*N3,fDim(2:end)]);

        Mmask = repmat({':'},[1,fDim]);

        for i=1:(N1*N2*N3)
        if(useDistance)
           fP = f(GetDistance(this,Pts.y1_kv(i) - Pts.y1_kv,...
             Pts.y2_kv(i) - Pts.y2_kv,...
                     Pts.y3_kv(i) - Pts.y3_kv));
         else
           fP = f(Pts.y1_kv(i) - Pts.y1_kv,...
                        Pts.y2_kv(i) - Pts.y2_kv,...
                        Pts.y3_kv(i) - Pts.y3_kv);
         end
        Mmask{1} = i;
        M_conv(Mmask{:}) = IntT.*fP;
        end
        M_conv(isnan(M_conv)) = 0;

        if((nargin >= 3) && islogical(saveBool) && saveBool)
                this.Conv = M_conv;
        end

end

function d  = GetDistance(this,pts_y1,pts_y2,pts_y3)
        if(nargin == 2)
```

```
            pts_y3 = pts_y1.y3_kv;
            pts_y2 = pts_y1.y2_kv;
            pts_y1 = pts_y1.y1_kv;
        end

        %ptsCart = GetCartPts(this,pts_y1,pts_y2,pts_y3);
        %d        = sqrt(ptsCart.y1_kv.^2 + ptsCart.y2_kv.^2);
        d        = sqrt(pts_y1.^2 + pts_y2.^2 + pts_y3.^2);
    end
```

2.1 Testing Convolution

We compute the convolution

$$n * \chi(x) = \int_\Omega \chi(x-y)n(y)dy.$$

In a first example we have

$$\begin{aligned} n(\vec{y}) &= \cos(y_1), \\ \chi(\vec{y}) &= \sin(y_1+y_2+y_3), \end{aligned}$$

with the exact solution

$$\begin{aligned} n * \chi(\vec{x}) &= 0.5\sin(0.5)\left(-2\sin(0.5)\cos(1-c) + 2\sin(0.5)\cos(3-c) - 4\sin(0.5)\sin(1-c)\right), \\ c &= x_1 + x_2 + x_3, \end{aligned}$$

in $[0,1]^2$. With $N = 10$, we have an error of 2.7412×10^{-11}.

Next we test

$$\begin{aligned} n(\vec{y}) &= y_1 y_2 y_3, \\ \chi(\vec{y}) &= exp(-|\vec{y}|^2), \end{aligned}$$

with the exact solution

$$\begin{aligned} T_1 &= (1/2)(\exp(-x_1^2) + \sqrt{\pi}x_1(erf(1-x_1) + erf(x_1)) - \exp(-(x_1-1)^2)), \\ T_2 &= (1/2)(\exp(-x_2^2) + \sqrt{\pi}x_2(erf(2-x_2) + erf(x_2)) - \exp(-(x_2-2)^2)), \\ T_3 &= (1/2)(\exp(-x_3^2) + \sqrt{\pi}x_3(erf(3-x_3) + erf(x_3)) - \exp(-(x_3-3)^2)), \\ n * \chi(\vec{x}) &= T_1 T_2 T_3, \end{aligned}$$

on $[0,1]\times[0,2]\times[0,3]$. For $N=10$ the error is 8.2056×10^{-5}. For $N=15$ it reduces to 5.8809×10^{-8} and for $N=20$ we get 6.6143×10^{-11}.

Finally we consider

$$\begin{aligned}
n(\vec{y}) &= (\sin(\pi y_1)^2)(\sin(\pi y_2)^2)(\sin(\pi y_3)^2)\\
\chi(\vec{y}) &= y_1 y_2 y_3
\end{aligned}$$

with the exact solution

$$n * \chi(\vec{x}) = (1/4)(2y_1-1)(y_2-1)(3/4)(2y_3-3),$$

on $[0,1]\times[0,2]\times[0,3]$. For $N=10$ we get an error of 0.0015 and for $N=20$ it reduces to 7.7335×10^{-8}. For $N=25$ we get 1.0006×10^{-11}. The errors are measured in the absolute standard Matlab norm.

3 Curl free control

We consider

$$\begin{aligned}
&\min_{\rho,\mathbf{w}} \frac{1}{2}\int_0^T\int_\Omega (\rho-\widehat{\rho})^2\, d\mathbf{x}dt + \frac{\beta}{2}\int_0^T\int_\Omega \mathbf{w}^2 d\mathbf{x}dt + \frac{\eta}{2}\int_0^T\int_\Omega (\nabla\times\mathbf{w})^2\, d\mathbf{x}dt\\
&\text{subject to:}\\
&\frac{\partial\rho}{\partial t} = \nabla^2\rho - \nabla\cdot(\rho\mathbf{w})\\
&\frac{\partial\rho}{\partial n} - \rho\mathbf{w}\cdot\mathbf{n} = 0
\end{aligned}$$

We know that in two dimensions

$$\nabla\times\mathbf{w} = \frac{\partial w_2}{\partial x_1} - \frac{\partial w_1}{\partial x_2}.$$

More importantly, we know that

$$\nabla\times\mathbf{w} = \nabla\cdot\mathbf{w}_\perp, \tag{1}$$

where $\mathbf{w}_\perp = (w_2, -w_1)$, the result of a rotation of $\mathbf{w}$ by $\pi/2$. Then the Lagrangian is

$$\begin{aligned}
\mathcal{L}(\rho,\mathbf{w},q_1,q_2) =& \frac{1}{2}\int_0^T\int_\Omega (\rho-\widehat{\rho})^2\, d\mathbf{x}dt + \frac{\beta}{2}\int_0^T\int_\Omega \mathbf{w}^2 d\mathbf{x}dt + \frac{\eta}{2}\int_0^T\int_\Omega (\nabla\cdot\mathbf{w}_\perp)^2\, d\mathbf{x}dt\\
&- \int_0^T\int_\Omega q_1\left(\frac{\partial\rho}{\partial t} - \nabla^2\rho + \nabla\cdot(\rho\mathbf{w})\right) d\mathbf{x}dt\\
&- \int_0^T\int_{\partial\Omega} q_2\left(\frac{\partial\rho}{\partial n} - \rho\mathbf{w}\cdot\mathbf{n}\right) d\mathbf{x}dt.
\end{aligned}$$

Then, since we know that $q_1 = q_2$, we get

$$\begin{aligned}\mathcal{L}(\rho, \mathbf{w}, q) =& \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \, d\mathbf{x}dt + \frac{\beta}{2}\int_0^T \int_\Omega \mathbf{w}^2 d\mathbf{x}dt + \frac{\eta}{2}\int_0^T \int_\Omega (\nabla \cdot \mathbf{w}_\perp)^2 \, d\mathbf{x}dt \\ & - \int_0^T \int_\Omega -\rho\frac{\partial q}{\partial t} - \rho\nabla^2 q - \nabla q \cdot (\rho\mathbf{w}) \, d\mathbf{x}dt - \int_\Omega q(T)\rho(T) - q(0)\rho(0)d\mathbf{x} \\ & - \int_0^T \int_{\partial\Omega} -\rho\nabla q \cdot \mathbf{n}d\mathbf{x}dt.\end{aligned}$$

For the adjoint equation, we find the usual results. We take the derivative with respect to $\mathbf{w}$

$$\mathcal{L}_{\mathbf{w}}(\rho, \mathbf{w}, q)h = \int_0^T \int_\Omega \beta\mathbf{w} \cdot \mathbf{h} + \rho\nabla q \cdot \mathbf{h} + \eta\left(\nabla \cdot \mathbf{h}_\perp\right)\left(\nabla \cdot \mathbf{w}_\perp\right) d\mathbf{x}dt.$$

Then we integrate by parts (or divergence theorem) to get

$$\mathcal{L}_{\mathbf{w}}(\rho, \mathbf{w}, q)h = \int_0^T \int_\Omega \beta\mathbf{w} \cdot \mathbf{h} + \rho\nabla q \cdot \mathbf{h} - \eta\nabla\left(\nabla \cdot \mathbf{w}_\perp\right) \cdot \mathbf{h}_\perp d\mathbf{x}dt + \int_0^T \int_{\partial\Omega} \eta\left(\nabla \cdot \mathbf{w}_\perp\right)\mathbf{h}_\perp \cdot \mathbf{n}d\mathbf{x}dt.$$

Finally we need to rewrite the equations in terms of $\mathbf{h}$. We note that

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\mathbf{h} = \mathbf{h}_\perp.$$

Furthermore,

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathbf{n} \cdot \mathbf{h} = \mathbf{h}_\perp \cdot \mathbf{n}.$$

Replacing these in the Lagrangian gives

$$\begin{aligned}\mathcal{L}_{\mathbf{w}}(\rho, \mathbf{w}, q)h = & \int_0^T \int_\Omega \beta\mathbf{w} \cdot \mathbf{h} + \rho\nabla q \cdot \mathbf{h} - \eta\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\nabla\left(\nabla \cdot \mathbf{w}_\perp\right) \cdot \mathbf{h}d\mathbf{x}dt \\ & + \int_0^T \int_{\partial\Omega} \eta\left(\nabla \cdot \mathbf{w}_\perp\right)\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathbf{n} \cdot \mathbf{h}d\mathbf{x}dt.\end{aligned}$$

Finally, using (1) we get

$$\begin{aligned}\mathcal{L}_{\mathbf{w}}(\rho, \mathbf{w}, q)h = & \int_0^T \int_\Omega \beta\mathbf{w} \cdot \mathbf{h} + \rho\nabla q \cdot \mathbf{h} - \eta\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\nabla\left(\nabla \times \mathbf{w}\right) \cdot \mathbf{h}d\mathbf{x}dt \\ & + \int_0^T \int_{\partial\Omega} \eta\left(\nabla \times \mathbf{w}\right)\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathbf{n} \cdot \mathbf{h}d\mathbf{x}dt.\end{aligned}$$

Since this holds for all admissible $\mathbf{h}$ we get the gradient equation

$$
\begin{aligned}
\beta \mathbf{w} + \rho \nabla q - \eta \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla \left(\nabla \times \mathbf{w} \right) &= 0 \quad \text{in} \quad \Omega \\
\eta \left(\nabla \times \mathbf{w} \right) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{n} &= 0 \quad \text{on} \quad \partial\Omega.
\end{aligned}
$$

In component form this is

$$
\begin{aligned}
\beta w_1 + \rho \frac{\partial q}{\partial x_1} - \eta \left(\frac{\partial^2 w_1}{\partial x_2^2} - \frac{\partial^2 w_2}{\partial x_1 \partial x_2} \right) &= 0 \quad \text{in} \quad \Omega \\
-\eta \left(\frac{\partial w_2}{\partial x_1} - \frac{\partial w_1}{\partial x_2} \right) n_2 &= 0 \quad \text{on} \quad \partial\Omega.
\end{aligned}
$$

and

$$
\begin{aligned}
\beta w_2 + \rho \frac{\partial q}{\partial x_2} - \eta \left(\frac{\partial^2 w_2}{\partial x_1^2} - \frac{\partial^2 w_1}{\partial x_1 \partial x_2} \right) &= 0 \quad \text{in} \quad \Omega \\
\eta \left(\frac{\partial w_2}{\partial x_1} - \frac{\partial w_1}{\partial x_2} \right) n_1 &= 0 \quad \text{on} \quad \partial\Omega.
\end{aligned}
$$